

Вятский государственный гуманитарный университет

ИНФОРМАТИКА
МАТЕМАТИКА
ЯЗЫК

Научный журнал

№ 5

Киров
2008

Главный редактор

В. С. Данюшенков

Редакционная коллегия:

В. Т. Юнгблюд (зам. главного редактора),
М. С. Судовиков (отв. секретарь),
К. С. Лицарева, Л. А. Мосунова,
Н. И. Поспелова, В. Н. Оношко,
Т. Я. Ашихмина, Г. И. Симонова,
М. И. Ненашев

Выпуск готовили: С. М. Окулов, С. И. Калинин, А. А. Свицова

Адрес редакции: 610002, г. Киров, ул. Красноармейская, 26,
тел.: (8332) 678-860 (научный отдел), (8332) 673-674 (Издательский центр)

Редакторы: Т. Котельникова, О. Коробкова, Ю. Болдырева
Компьютерная верстка: Ю. Боброва

Свидетельство о регистрации средства массовой информации
(Министерство по делам печати, телерадиовещания
и средств массовых коммуникаций)
ПИ № 77-14376 от 17 января 2003 г.

СОДЕРЖАНИЕ

ИНФОРМАТИКА

Акимова И. В. Создание электронного учебника средствами Macromedia Dreamweaver и Macromedia Authorware	6
Ашихмина Т. В. Алгоритмизация в теории вероятностей	8
Белов Д. В. Incremental Active Set Method (INCAS) обучения SVM	12
Васенина Е. А. О проблеме интеллектуального воспитания школьников в процессе обучения информатики	15
Васенина Е. А., Хомякова Д. А. О структурировании банка задач в целях индивидуализации процесса обучения программированию	20
Вахрушев А. С., Захарова А. Н., Исупова Н. И. Из опыта подготовки к Интернет-тестированию по информатике	24
Дьячков В. П. Разработка обучающих программ по информатике с использованием различных компьютерных программ	26
Жаховская И. С. Из опыта использования метода проектов на уроках информатики	31
Исупова Н. И. Место дисциплины «Численные методы и программирование» в подготовке студентов химического факультета	37
Исупова Н. И. О применении электронных образовательных ресурсов в процессе обучения	41
Исупова Н. В., Кривокопытнова В. В. Об эффективности обучения студентов гуманитарных специальностей	46
Колеватов В. Ю. Методы искусственного интеллекта в задачах обеспечения безопасности компьютерных сетей	49
Колупаева Е. С. Изучение псевдослучайных последовательностей в школьном курсе информатики и ИКТ	53
Корякина И. А. Инновационный подход в обучении ИТ-специалистов в высшей школе	57
Корякина И. А., Тушицына О. Н. Особенности использования программного продукта «1С:Предприятие 8 Управление страховой компанией» в учебном процессе ВятГГУ	60
Котельников Е. В., Стародубова Т. А. Распараллеливание методов обучения в задачах многоклассовой классификации	63
Котельникова А. В. Кластер рабочих станций в вузе на основе Microsoft Compute Cluster Server 2003	66
Лялин А. В. Битовая арифметика и поиск подстроки в строке	73
Мельникова М. Ю. Специфика внедрения корпоративных информационных систем	81
Мельникова М. Ю. Методические аспекты изучения студентами дисциплины «Корпоративные экономические информационные системы»	84
Морданов М. А. О разложении натуральных чисел на сумму простого и двух квадратов	88
Окулов С. М. Еще раз про информатику	89
Семенов Р. А. Экспериментальное исследование проблемы Гольдбаха – Эйлера	93
Скобелев А. Е. Решета простых чисел	98
Соболева Е. В. К вопросу о разработке тренинга педагогического общения для учителей информатики	101
Соболева Е. В. Построение формальной модели описания взаимодействия учителя и ученика на уроке информатики	108
Токмакова О. В. Проблемы управления процессом обучения на уроках информатики	113
Токмакова О. В. Организация системы дистанционного образования в вузе	115
Хомякова Д. А. Об опыте проектирования информационной системы на примере разработки электронного средства для обучения программированию	117
Шарыгин Р. В. Исследование производительности параллельных программ	120
Шихов М. М., Петров Р. С. Параллельный метод дедуктивного логического вывода в исчислении предикатов	130
Юфрина Е. В. Методика изучения конфигурации 1С:Торговля и склад на основе когнитивно-прагматического подхода	136
Ямшанов М. А., Котельников Е. В. Метод определения тематической значимости лексических единиц	142

МАТЕМАТИКА

<i>Акчурина Э. Х.</i> О формировании самоконтроля у младших школьников в процессе обучения математике	149
<i>Бердников М. С.</i> Теорема о выпуклости и α -выпуклости	151
<i>Бушмелева Н. А., Разова Е. В.</i> Проблемы математической подготовки студентов и пути их решения	154
<i>Вечтомов Е. М.</i> О пучковом представлении редуцированных ограниченных полутел	158
<i>Вечтомов Е. М., Пуртова С. А.</i> Мини-топологии	161
<i>Иномистов В. Ю., Рапопорт А. Н.</i> Применение специализированных пакетов компьютерной математики в учебном процессе	168
<i>Калинин С. И.</i> О предмете математического анализа	170
<i>Клековкина М. В., Яшина А. Г.</i> Четыре метода доказательства одного неравенства	175
<i>Онегов В. А.</i> Приведение исходной линейной системы к виду, удобному для применения метода итерации с гарантированной сходимостью	179
<i>Панкратова А. В.</i> Об уточнении неравенства Коши методом Mihály Bencze	183
<i>Плетнёв К. В.</i> Обобщение одной теоремы о среднем значении	186
<i>Прилуцкий М. Х., Кумагина Е. А.</i> Применение управляемого фронтального алгоритма для решения задачи многоресурсного сетевого планирования	188
<i>Прокушкина Е. А.</i> Некоторые применения неравенства Гюйгенса и его уточнения	191
<i>Сергеева Е. В.</i> Развитие поисковой активности школьников при работе с незавершенными задачами ситуациями	194
<i>Соколова А. Н.</i> Об использовании ресурсов Интернет в учебной и научно-исследовательской работе студентов-математиков	196
<i>Таранова М. В.</i> Два подхода в организации познания	200

ЯЗЫК

<i>Агалакова Т. Б., Казаковцева Ю. В.</i> Концепт good/добро: общее и специфическое	202
<i>Агалакова Т. Б., Шляева А. М.</i> Возможности использования технологии портфолио в организации самостоятельной работы студентов	203
<i>Бардовская Л. И.</i> Некоторые закономерности номинации синестетических ощущений в произведениях русских писателей	205
<i>Кипрская Е. В.</i> Основные сферы использования эвфемизмов и их отграничение от сходных лексических явлений	208
<i>Кожина Н. С.</i> Персуазивное сообщение и его канал в современной коммуникации	212
<i>Новикова Н. В.</i> Стилистические функции архаизмов в новеллах Э. По	217
<i>Свицова А. А.</i> When in Rome, do as Romans do	220



Настоящий период характеризуется переходом к экономике, имеющей большую интеллектуальную составляющую. Чем более образованным, мобильным, энергичным будет поколение, тем больше шансов получает государство для своего развития.

Значительную роль приобретает способность человека осваивать новое информационное пространство, ориентировать специалиста на формирование системного творческого мышления и способности генерировать новые идеи, поэтому качественная подготовка специалистов по информатике, включающая, в частности, научную работу студентов и преподавателей, имеет огромное значение.

*Т. В. Машарова,
доктор педагогических наук, профессор,
профектор по учебно-воспитательной работе ВятГГУ*

ИНФОРМАТИКА

И. В. Акимова

СОЗДАНИЕ ЭЛЕКТРОННОГО УЧЕБНИКА СРЕДСТВАМИ MACROMEDIA DREAMWEAVER И MACROMEDIA AUTHORWARE

Электронный учебник в настоящее время считается одним из наиболее эффективных средств организации процесса обучения с использованием современных информационно-коммуникационных технологий. В данной статье дается определение электронного учебника как учебника, реализованного в электронном виде с использованием компьютерных средств обучения и контроля знаний обучаемых. Рассматриваются возможности разработки электронных учебников с помощью таких программных средств, как Macromedia Dreamweaver и Macromedia Authorware.

Одним из эффективных средств организации процесса обучения с использованием современных информационно-коммуникационных технологий является электронный учебник (ЭУ).

Дадим следующее определение: **электронный учебник** – учебник, реализованный в электронном виде с использованием компьютерных средств обучения и контроля знаний обучаемых.

Достоинствами электронных учебников, на наш взгляд, являются: во-первых, высокая мобильность, во-вторых, доступность широкому кругу пользователей в компьютерных сетях, в-третьих, адекватность современному уровню научного знания. Создание ЭУ эффективно решает проблему постоянного обновления информационно-дидактического материала. В нем может содержаться большое количество упражнений и примеров, подробно иллюстрируется динамика изменения объектов. Кроме того, ЭУ могут включать средства контроля знаний – компьютерное тестирование.

Незаменимы электронные учебники и для организации самостоятельной работы обучаемых, так как они облегчают восприятие и понимание изучаемого материала на основе новых способов подачи материала: индуктивного подхода, воздействия на слуховую и эмоциональную память

АКИМОВА Ирина Викторовна – кандидат педагогических наук, доцент по кафедре информатики и методики преподавания информатики Пензенского государственного педагогического университета им. В. Г. Белинского
© Акимова И. В., 2008

и т. д.; допускают адаптацию к потребностям обучаемого, уровню его подготовки, интеллектуальным возможностям и стремлениям; предоставляют широкие возможности для самоконтроля на всех этапах обучения.

Средства создания электронных учебников можно разделить на группы на основе комплексного критерия, учитывающего назначение и выполняемые функции, требования к техническому обеспечению и особенности использования. В соответствии с этим критерием можно выделить следующие классы:

- традиционные алгоритмические языки;
- инструментальные средства общего назначения;
- средства мультимедиа;
- гипертекстовые и гипермедиа средства.

При создании учебника мы рекомендуем ориентироваться на следующие основные этапы:

1. Определение целей и задач разработки.
2. Разработка структуры электронного учебника.
3. Разработка содержания по разделам и темам учебника.
4. Подготовка отдельных компонент электронного учебника.
5. Программирование.
6. Аprobация.
7. Корректировка содержания ЭУ по результатам апробации.
8. Подготовка методического пособия для пользователя.

Электронные учебники могут иметь различную структуру: линейную, древовидную, фреймовую (см. рис. 1).

При выборе средства разработки электронного учебника мы предлагаем остановить внимание на среде Macromedia Dreamweaver и Macromedia Authorware.

Macromedia Dreamweaver – современное программное средство, простое в изучении даже неопытными пользователями. Учебники будут создаваться средствами языка разметки HTML. Но полученные электронные учебные пособия ориентированы на большой объем текстового материала со статическими иллюстрациями и простейшими формулами.

Обычно ЭУ состоит из нескольких файлов-глав, поэтому начать работу необходимо с системы каталогов. Создание web-страниц за-

дается на языке **HTML** (HyperText Markup Language – язык гипертекстовой разметки). Этот язык определяет набор специальных команд, называемых тегами и используемых для задания форматирования или назначения тех или иных элементов Web-страницы. При работе в среде Macromedia Dreamweaver знание этих тегов не обязательно. Но одной из полезных особенностей среды Macromedia Dreamweaver является то, что мы можем видеть на экране как html-код нашей страницы, так и ее визуальную реализацию (рис. 2).

Для оформления (дизайна) страницы используем Properties Inspector (в нижней части экрана). Там задаются необходимые свойства, отвечающие за цвет текста на странице (Text Color), изменения цвета фона и пр. Для форматирования текста в среде Dreamweaver необходимо вос-

пользоваться пунктом меню Text. Для вставки рисунка необходимо скопировать файл с нужным рисунком в папку pic. Затем выбрать пункт меню Insert – Image.

Пакет Authorware предназначен, прежде всего, для создания мультимедийных приложений, предусматривающих совместное использование различных форм подачи материала (текста, рисунка, звукового сопровождения и т. д.). Основная же функция Authorware – создание электронных обучающих систем. Это визуальная среда для разработки интерактивных обучающих курсов. С помощью наглядного интерфейса и встроенных средств обучения – контрольных вопросов, тестов – мы имеем возможность легко и быстро создавать учебный курс. Встроенная система контроля позволяет регистрировать и хранить показатели обучаемых. Конечный про-

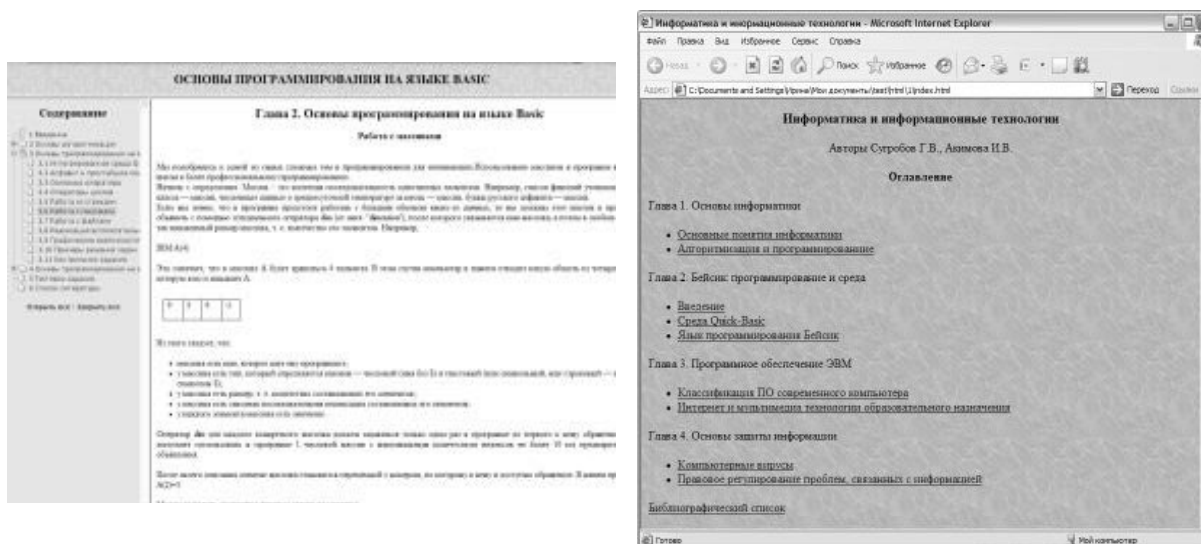


Рис. 1. Пример электронных учебников с различной структурой

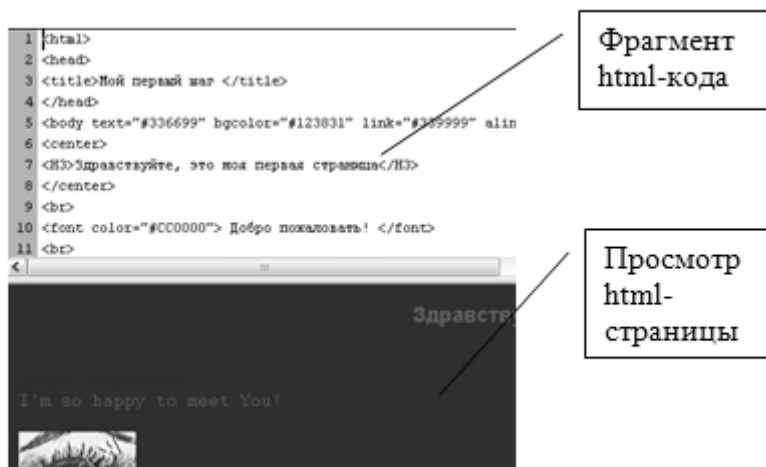


Рис. 2. Фрагмент окна Dreamweaver

дукт, созданный в системе, представляет собой независимое приложение, которое может быть записано на внешний носитель в виде exe-файла, либо опубликовано в Интернете.

В процессе создания ЭУ мы видим материал в двух формах:

- в виде схемы, описывающей последовательность кадров;
- в виде содержимого кадров, отображаемого на экране (рис. 3).

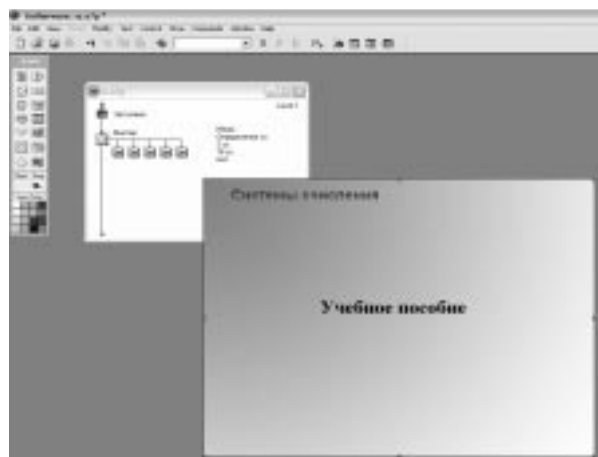


Рис. 3. Две формы представления учебного курса

При работе в системе Authorware мы имеем следующие возможности:

1. Добавлять в схему новые кадры, давать кадрам произвольные имена, изменять положение кадров на схеме.
2. Добавлять управляющие кадры, с помощью которых создается навигационная схема курса.
3. Импортировать внешние ресурсы (графику, звук, видео) как непосредственно на кадр, так и в библиотеку ресурсов.

При работе с кадром мы можем импортировать в него текст в формате rtf или txt, изображения в одном из поддерживаемых форматов, создавать новые рисунки, вводить новый текст внутренними средствами приложения, преобразовывать текст в гиперссылку, анимировать визуальный объект и пр.

Особый интерес представляет создание подсистемы контроля знаний. Для ее реализации в Authorware предусмотрена специальная библиотека Knowledge Objects. С помощью содержащихся в ней объектов мы можем реализовать систему тестирования, содержащую запрос информации об обучаемом, сохранение результатов во внешний файл, использование нескольких типов вопросов: множественного выбора, вопросов типа «да – нет», открытого вопроса, работы с изображением.

Система Authorware также имеет свой редактор макросов, который содержит встроенные процедуры и функции, возможность создания стандартных окон Windows и т. д.

В заключение отметим, что возможности системы Macromedia Authorware и Dreamweaver достаточно широки и не ограничиваются всем вышеперечисленным. Программные продукты могут реализовывать практически все требования к организации современной системы электронного обучения, хотя каждый предназначен для создания своего типа электронного учебника.

Т. В. Ашихмина

АЛГОРИТМИЗАЦИЯ В ТЕОРИИ ВЕРОЯТНОСТЕЙ

В статье раскрываются некоторые проблемы преподавания курса теории вероятностей, показывается, каким образом внедрение алгоритмизации и программирования в методику изучения этого курса помогает решать поставленные вопросы и осуществлять межпредметные связи теории вероятностей (как раздела математики) и информатики. В статье также представлены примеры вероятностных задач, при решении которых применим алгоритмический подход, и варианты аналогичных задач для самостоятельного решения.

Благодаря развитию методики применения компьютера в обучении, появлению новых информационных технологий, модернизации содержания математического образования, в частности включению в школьные программы вероятностно-статистической линии, компьютер стал использоваться при преподавании курса теории вероятностей и математической статистики.

На сегодняшний день при изучении вероятностно-статистического материала с использованием компьютера большая доля времени отводится описательной и математической статистике. В качестве средства для статистической обработки информации в некоторых существующих учебных пособиях по теории вероятностей и математической статистике предполагается использовать табличный процессор Microsoft Excel. Компьютер, как инструмент для работы с информацией, позволяет упростить сбор и обработку статистической информации, наглядно представить полученную информацию, производить моделирование случайных процессов.

Логичным продолжением внедрения инструмента информатики в методику изучения курса

АШИХМИНА Татьяна Викторовна – старший преподаватель кафедры прикладной математики ВятГУ
© Ашихмина Т. В., 2008

теории вероятностей и математической статистики было бы использование знаний, умений и навыков алгоритмизации и программирования при решении вероятностных задач.

Алгоритмизация в теории вероятностей помогает решить следующие проблемы.

- Некоторые из обучающихся теории вероятностей (по своей сути практики и исследователи) не всегда доверяют готовым формулам (даже теоретически выведенным). Их больше привлекает статистический подход, основанный на проведении опытов. Для таких учащихся и студентов на первом этапе полезно было бы сравнить результаты классического и статистического подходов к вычислению вероятностей в одной и той же задаче. Некоторые алгоритмы также помогут выполнить своеобразную проверку задач, решенных классически с использованием формул подсчета комбинаторных объектов. Для этого достаточно либо перебрать все возможные исходы и подсчитать количество благоприятных среди них (получаем точный результат), или сгенерировать достаточно большое число случайных опытов и подсчитать количество благоприятных среди них (получаем приближенный результат).

- Достаточно интересны и познавательны задачи на непосредственное вычисление вероятностей, отвечающие на вопросы: какая из двух предложенных ситуаций наиболее вероятна или при каких начальных условиях вероятность будет максимальной (минимальной). При этом не всегда такие вопросы можно свести к математической задаче о нахождении экстремума функции. Поэтому приходится решать несколько однотипных задач с различными входными данными и сравнивать полученные результаты. Программная реализация одного алгоритма и тестирование полученной программы на разных входных данных поможет избавиться от рутинных вычислений вручную.

- В практике реальной жизни решаются задачи с очень большим объемом входных данных или с перебором большого числа различных вариантов. Например, составить закон распределения (или найти среднее значение) количества студентов одной группы, сдавших экзамен. Известно количество человек в группе, число вопросов в программе экзамена и количество вопросов из программы, выученных каждым из студентов этой группы. Каждый студент может использовать еще две пересдачи в случае получения неудовлетворительной оценки, при этом вероятность сдать экзамен на каждой следующей пересдаче у студента повышается в два раза (более ответственная подготовка к экзамену). Вероятности в данной задаче вычисляются лишь как суммы произведений вероятностей независимых

событий. Но слагаемых и множителей здесь достаточно много и все они различны, следовательно, требуется полный перебор всех вариантов, в чем может помочь алгоритм.

Следовательно, использование компьютера для реализации алгоритмов при решении вероятностных задач углубляет знания курсов теории вероятностей и программирования, позволяет усилить интерес к этим предметам, показать их значимость, универсальность, способствует развитию мышления учащихся, может значительно уменьшить объем вычислительной работы, сэкономить время, усовершенствовать навыки программирования и работы с прикладными программами.

Рассмотрим некоторые вероятностные задачи, к которым применим алгоритмический подход.

Задача № 1

Сравним с помощью компьютера результаты классического и статистического подходов к вычислению вероятностей.

Для этого рассмотрим задачу: «С какой вероятностью при однократном подбрасывании игральной кости выпадет 5 очков?»

По классическому определению вероятность равна $p = 1/6 = 0,16(6)$.

Вспользуемся статистическим определением. Смоделируем подбрасывание игральной кости n раз (т. е. сгенерируем n случайных целых чисел из отрезка от 1 до 6). Подсчитаем при этом m – количество выпадений числа 5. Найдем относительную частоту появления 5 очков как $w = m/n$.

Сравним абсолютные значения отклонений теоретической вероятности p от относительной частоты w при различных способах генерации случайных чисел.

1. Вспользуемся генератором случайных чисел на языке программирования Pascal.

```
. . .
Randomize; m:=0;
For i:=1 to n do
  Begin
    x:=1+Random(6);
    If x=5 then inc(m);
  End;
w:=m/n;
. . .
```

Например, для $n = 100000$ возможное значение $w = 0,16704$, следовательно, $|p - w| = 0,00037$.

2. Генератор случайных чисел как инструмент **Анализа данных** в электронных таблицах Excel.

В меню **Сервис** выберите команду **Анализ данных**. Если эта команда недоступна, загрузите пакет анализа. Для этого:

- 1) В меню **Сервис** выберите команду **Надстройки**.
- 2) В списке надстроек выберите **Пакет анализа** и нажмите кнопку **ОК**.

Выберите функцию **Генерация случайных чисел** в диалоговом окне **Анализ данных** и нажмите кнопку **ОК**.

Установите параметры анализа в соответствующем диалоговом окне. **Число переменных = 1, число случайных чисел = 1000, распределение – равномерное, параметры – между 1 и 6.**

Для округления полученных случайных чисел используйте функцию **ОКРУГЛ**(<адрес ячейки>; 0), для вычисления числа m – функцию **СЧЕТЕСЛИ**(<диапазон>;5).

Например, для $n = 1000$ возможное значение $w = 0,186$, следовательно, $|p - w| = 0,0193$.

Варианты аналогичных задач

1. Найти вероятность, с которой, дважды подбросив игральную кость, игрок получает сумму выпавших очков, равную 6.

2. Найти вероятность, с которой корни квадратного уравнения $ax^2 + bx + c = 0$ являются рациональными числами. Коэффициенты a, b, c равны количеству выпавших очков при трехкратном подбрасывании игральной кости.

3. Три человека, сев в лифт на первом этаже семиэтажного дома, с равной вероятностью могут выйти на любом из этажей выше первого. Найти вероятность, с которой все пассажиры лифта выйдут на разных этажах.

Задача № 2

«О победе футбольной команды». Группа из n фанатов выигрывающей футбольной команды на радостях бросает свои шляпы в воздух. Шляпы возвращаются в случайном порядке – по одной к каждому из болельщиков. С какой вероятностью k болельщиков получают назад свои собственные шляпы?

Формализовав задачу, мы приходим к следующему вопросу: «Сколько существует перестановок из пяти элементов, в которых k элементов стоят на своих местах?» Ответив на этот вопрос и воспользовавшись тем, что общее число перестановок из n элементов равно $n!$, решаем задачу по классическому определению вероятности.

Используем для решения алгоритм генерации всех перестановок из n элементов с подсчетом количества k неподвижных точек в каждой перестановке.

Предлагаем фрагмент программной реализации этого алгоритма на языке программирования Pascal.

Структуры данных:

- перестановка из элементов 1, 2, ..., n – одномерный массив a ;
- количество перестановок с k неподвижными точками – целая переменная s ;
- искомая вероятность – вещественная переменная p .

Procedure P (var a:mas); {формирование следующей перестановки из n элементов}

```
var i,j:byte;
begin
  i:=n-1;
  while (i>0) and (a[i]>a[i+1]) do dec(i);
  j:=n;
  while a[j]<a[i] do dec(j);
  < a[i]+a[j] >;
  For j:=1 to (n-i) div 2 do
  <a[i+j]:a[n-j+1]> ;
end;
{основная программа}
Begin
  f:=<факториал числа n >;
  if n=k then p:=1/f
  else begin
  <генерация первой перестановки из n
  элементов, а именно, перестановки 1,
  2, 3, ...,n>;
  c:=0;
  For i:=2 to f do
  begin
    P(a);
    x:=< количество неподвижных
    точек в перестановке a >;
    if x=k then inc(c);
  end;
  p:=c/f;
end;
Writeln('Ответ: ', p:5:3);
Readln;
End.
```

Варианты аналогичных задач

1. В урне находятся 10 пронумерованных шаров. Шары перемешиваются и последовательно, один за другим, извлекаются из урны. С какой вероятностью хотя бы для двух шаров их номера не совпадут с номерами извлечения?

2. На званый обед приглашены 10 гостей. Хозяин решил, как именно нужно рассадить гостей за стол и приготовил карточки с именами. С какой вероятностью хотя бы четверо из приглашенных гостей сядут на выбранные хозяином места, если он забыл расставить карточки.

3. Найти все решения в задаче «о победе футбольной команды» для значений n и k от 0 до 10. Результат работы программы – двумерная таблица размерности 11 x 11.

Задача № 3

Имеется $2n$ шаров, из них $2k$ белых. Шары раскладывают в две коробки поровну. С какой вероятностью в коробках окажется одинаковое число белых шаров?

Решение задачи с помощью классического

определения:
$$p = \frac{C_{2k}^k \cdot C_{2n-2k}^{n-k}}{C_{2n}^n}.$$

Для $n = 4$, $k = 3$ получаем

$$p = \frac{C_6^3 \cdot C_2^1}{C_8^4} = \frac{4}{7} \approx 0,57142857.$$

Число сочетаний C_n^k для достаточно больших различных значений n и k может быть вычислено либо в Excel с помощью функции ЧИСЛКОМБ(), либо с помощью функции, написанной на языке программирования Pascal, в которой вычисляется значение Ф4.

Алгоритмический подход к решению задачи

Заполним одномерный массив a размерности $2n$ целыми числами (все имеющиеся шары), среди которых $2k$ чисел отрицательные (белые шары), а остальные положительные. Для моделирования разбиения шаров на две равные группы сгенерируем все последовательности из нулей и единиц длиной $2n$ (одномерный массив b). При этом элементы, помеченные 0, относятся к первой группе, а помеченные 1 – ко второй. Если количество единиц и нулей в массиве b совпадают и равны n , то произошло разбиение шаров на равные части (количество таких разбиений подсчитываем в переменной v). Если количество отрицательных чисел (белых шаров) в одной из двух равных частей совпадает с k , следовательно, в обеих коробках по k белых шаров (количество таких разбиений подсчитываем в переменной m). Искомую вероятность найдем по классическому определению $p = m/v$.

```
...
const n=4;k=3;
type mas=array[1..2*n] of integer;
var a,b:mas; m,v:byte;p:real;
procedure Input (var a:mas);
var i:byte;
begin
  for i:=1 to 2*k do
    a[i]:=(-1)*i;
  for i:=2*k+1 to 2*n do
    a[i]:=i-2*k;
end;
procedure Rec(i,c,x:byte);
begin
  If i<=2*n then
```

```
begin
  b[i]:=0; Rec(i+1,c,x);
  b[i]:=1; If a[i]<0 then
  Rec(i+1,c+1,x+1) else Rec(i+1,c+1,x);
end
else
  if c=n then
  begin
    inc(v);
    if x=k then inc(m);
  end
end;
End.
Begin
  Input(a);
  m:=0; v:=0;
  Rec(1,0,0);
  p:=m/v;
  writeln("p=",p:5:12);
  readln;
End.
```

Варианты аналогичных задач

1. Группу, в которой $3n$ студентов и из них $3k$ девушек ($k \leq n$), произвольным образом разбивают на три равные подгруппы. С какой вероятностью в каждой из подгрупп окажется одинаковое число девушек?

2. Колода из 52 карт делится на две равные части. С какой вероятностью в одной из частей будет ровно 3 туза?

3. Было куплено 30 лотерейных билетов, среди которых оказалось 9 выигрышных (3 билета с выигрышем по 100 рублей, 3 билета – по 200 рублей и 3 билета – по 300 рублей). Три друга поровну поделили все билеты между собой. С какой вероятностью все получают равные по сумме выигрыши?

Таким образом, алгоритмизацию и программирование можно отнести к области взаимного проникновения и обогащения методики изучения теории вероятностей (как раздела математики) и информатики. Решение представленных в статье задач способствует развитию мышления студентов, помогает углубить знания курсов математики и информатики, усилить интерес к этим предметам, показать их универсальность.

INCREMENTAL ACTIVE SET METHOD (INCAS) ОБУЧЕНИЯ SVM

В статье рассматривается метод инкрементного активного множества (Incremental Active Set method, INCAS) для обучения машины опорных векторов. Метод INCAS возможно применять в задачах классификации и распознавания образов. Предлагается использовать метод Лагранжа для решения подзадачи оптимизации. Приведены результаты тестирования метода INCAS

Введение

Классификация – это в общем случае нахождение правила, которое, основываясь на некоторых свойствах, относит объект к одному из существующих классов.

Современная теория классификации включает в себя подходы и методы из различных областей науки: статистики, теории измерений, систем искусственного интеллекта. Во второй половине XX в. было разработано множество методов решения задач классификации, каждый из которых имеет свои достоинства, недостатки и традиционные области применения.

В начале 90-х формулируется и начинает развиваться новый метод классификации под названием Support Vector Machine (SVM, машина опорных векторов), основанный на работах В. Н. Вапника и А. Я. Червоненкиса [1].

В настоящее время известны несколько алгоритмов, реализующих обучение машины опорных векторов, причем среди них сложно выбрать наилучший, так как каждый приспособлен для решения задач с определенным набором данных в определенных условиях [2].

Ниже приводится описание одного из алгоритмов обучения машины опорных векторов.

1. Алгоритм обучения INCAS

Используем классическую формулировку задачи обучения SVM.

1.1. Постановка задачи

Будем рассматривать задачу классификации с двумя непересекающимися классами. Пусть заданы: множество $X = \{X_1, X_2, \dots, X_n\}$ обучающих объектов, описываемых n -мерными вещественными векторами, и множество Y ответов для обучающих объектов $Y = \{y_1, y_2, \dots, y_n\}$, где $y_i \in \{1, -1\}$. Необходимо построить некоторую функцию (алгоритм) $\varphi : X \rightarrow Y$, который каждому объекту обучающего множества X сопоставляет верный ответ множества Y . В дальнейшем полученную функцию предполагается ис-

пользовать для классификации объектов, о принадлежности к классам которых неизвестно.

Идея машины опорных векторов состоит в том, чтобы построить в n -мерном евклидовом пространстве некоторую гиперплоскость, разделяющую множество X . При этом требуется, чтобы разделяющая гиперплоскость максимально далеко отстояла от ближайших к ней точек обеих классов.

Гиперплоскость в пространстве R^n описывается уравнением $(w, x) = w_0$, т. е. нужно найти такой вектор w и скалярный порог w_0 , чтобы любой обучающий вектор X функция $y(X) = \text{sign}((w, x) - w_0)$ отображала в верный ответ y .

Вопрос нахождения оптимальной гиперплоскости приводит к следующей двойственной задаче выпуклого квадратичного программирования,

$$\left\{ \begin{array}{l} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \lambda_i \lambda_j y_i y_j (y_i, y_j) - \sum_{i=1}^l \lambda_i \rightarrow \min; \\ \sum_{i=1}^l \lambda_i y_i = 0; \\ 0 \leq \lambda_i \leq C, i = 1, \dots, l. \end{array} \right. \quad (1.1)$$

где $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ – вектор двойственных переменных.

Решение данной задачи относительно λ дает нам вектор w , вычисляемый по формуле

$$w = \sum_{i=1}^l \lambda_i y_i x_i, \quad (1.2)$$

а также порог

$$w_0 = \text{med}\{w \cdot x_i - y_i : \lambda_i > 0, i = 1, \dots, l\}. \quad (1.3)$$

Общие методы таких задач известны, однако при достаточно большой размерности задачи (больше 10^5) малоэффективны, так как решение требует хранения в памяти матрицы коэффициентов, размер которой пропорционален квадрату числа обучающих примеров. Поэтому необходимо оптимизировать процесс решения задачи, используя особенности SVM.

1.2. Построение алгоритма

Рассмотрим алгоритм, впервые описанный в 2002 г. [3] под названием «метод инкрементного активного множества» (Incremental Active Set method, INCAS).

Введем векторно-матричные обозначения задачи (1.1).

Введем матрицу $Q = (y_i y_j K(x_i, x_j))_{i=1..l}^{j=1..l}$ размер $l \times l$ и три вектор-столбца длины l : вектор ответов $y = (y_i)_{i=1..l}$, вектор двойственных переменных $\lambda = (\lambda_i)_{i=1..l}$ и вектор единиц $e = (1)_{i=1..l}$.

В данных обозначениях двойственная задача будет иметь следующий вид:

$$\left\{ \begin{array}{l} \frac{1}{2} \lambda^T Q \lambda - e^T \lambda \rightarrow \min; \\ y^T \lambda = 0; \\ 0 \leq \lambda \leq C e. \end{array} \right. \quad (1.4)$$

Присвоим каждому вектору из обучающего множества индекс i ($i \in \{1, \dots, l\}$).

Предположим, что решение двойственной задачи λ еще не найдено, но известно разбиение множества индексов на три непересекающихся множества (построение начального разбиения показано в пункте 2.1):

$I_O = \{i : \lambda_i = 0\}$ – периферийные объекты;

$I_S = \{i : 0 < \lambda_i < C\}$ – опорные объекты;

$I_C = \{i : \lambda_i = C\}$ – объекты-нарушители (т. е. объекты, которые при текущей построенной гиперплоскости, попадают в полосу).

Основная идея метода INCAS состоит в том, чтобы выделить объекты (активный набор), на которых ограничения-неравенства обращаются в равенства (периферийные объекты и объекты – нарушители), и затем решить уменьшенную двойственную задачу. После получения решения определить, является ли оно оптимальным для полной двойственной задачи. Если нет, то изменить активный набор, выявив объекты, нарушающие условия принадлежности к классам.

Для постановки уменьшенной задачи введем дополнительные обозначения. Опираясь на разбиение индексов (I_S, I_C, I_O), обозначим Q_{SS} (Q_{CS} , Q_{SO} и т. д.) подматрицу матрицы Q , в которой столбцы индексируются из множества I_S (I_C, I_S и т. д.), строки – из множества I_S (I_C, I_O и т. д.).

Аналогично обозначим столбец y_S (y_C, y_O) и λ_S (λ_C, λ_O), являющийся подвектором y и λ , с индексами из I_S (I_C, I_S) и единичный столбец e .

В этих обозначениях уменьшенная задача принимает вид:

$$\left\{ \begin{array}{l} \frac{1}{2} \lambda_S^T Q_{SS} \lambda_S + C e_C^T Q_{CS} \lambda_S - e_S^T \lambda_S \rightarrow \min; \\ y_S^T \lambda_S + C e_C^T y_C = 0. \end{array} \right. \quad (1.5)$$

Это задача минимизации квадратичного функционала от $t = |I_S|$ переменных при одном условии-равенстве. Решение данной задачи будет продемонстрировано в пункте 2.2.

Алгоритм реализации обучения выглядит так:

Шаг 0. Построение начального разбиения индексного множества на I_S, I_C, I_O .

Шаг 1. Решение уменьшенной задачи относительно λ_S :

$$\left\{ \begin{array}{l} \frac{1}{2} \lambda_S^T Q_{SS} \lambda_S + C e_C^T Q_{CS} \lambda_S - e_S^T \lambda_S \rightarrow \min; \\ y_S^T \lambda_S + C e_C^T y_C = 0. \end{array} \right.$$

Решением будет вектор λ_S .

Пока $|I_S| > 2$, то просмотреть все множество I_S ; если $\lambda_i \leq 0$, то i перевести в I_O ; если $\lambda_i \geq C$, то i перевести в I_C .

Повторять шаг 1 до тех пор, пока в I_S есть объекты не своего класса.

Шаг 3. Вычислить вектор w и порог w_0 по формулам (1.2) и (1.3).

Вычислить отступы s_i объектов из $I_O \cup I_C$ от текущей границы классов, $s_i = y_i(w \cdot x_i - w_0)$.

Просмотреть уже множество $I_O \cup I_C$;

если $i \in I_O$ и $m_i \leq 1$, то i перевести в I_S ;

если $i \in I_C$ и $m_i \geq 1$, то перевести в I_S .

Повторять шаги 1, 2, 3 до тех пор, пока в $I_O \cup I_C$, нуждающиеся в переводе в I_S .

Конец алгоритма.

2. Решение подзадач алгоритма

В построенном алгоритме присутствуют некоторые шаги, требующие более детального рассмотрения.

2.1. Построение начального разбиения

Из-за особенностей алгоритма количество итераций существенно зависит от того, насколько близко выбрано начальное множество I_S от реального множества опорных векторов. Наилучшим решением было бы выбрать в качестве начального приближения две ближайших точки (понятно, что они будут входить в реальное множество опорных векторов) из различных множеств. Общий алгоритм для одного множества известен [4]. Однако этот алгоритм не позволяет находить две ближайшие точки из разных множеств. Поэтому будем использовать прием, позволяющий приблизиться к данному варианту.

Выбирается произвольная точка первого класса, для нее находится ближайшая точка из второго класса, путем простого перебора. Для найденной точки находится ближайшая точка из первого класса и так далее. Этот итерационный процесс довольно быстро приводит к паре некоторых близких точек, которые и можно взять в качестве начального приближения.

Этот процесс можно повторить для некоторой другой или даже нескольких точек и выбрать среди них ближайшие.

2.2. Решение мини-задачи оптимизации

При выполнении алгоритма на каждой итерации возникает следующая мини-задача оптимизации:

$$\left\{ \begin{array}{l} \frac{1}{2} \lambda_S^T Q_{SS} \lambda_S + C e_C^T Q_{CS} \lambda_S - e_S^T \lambda_S \rightarrow \min; \\ y_S^T \lambda_S + C e_C^T y_C = 0. \end{array} \right. \quad (2.1)$$

Авторы алгоритма аналитически сводят решение этой задачи к обращению матрицы коэффициентов [3], попробуем решить её, применяя ме-

тод Лагранжа [5, 6], используя особенности матриц Q_{SS} и Q_{CS} .

Далее будем обозначать Q с индексами, соответствующий член матрицы Q_{SS} и \bar{Q} с индексами, соответствующий член матрицы Q_{CS} . Обозначим $m = |I_S|$, количество опорных векторов, $n = |I_C|$, количество векторов-нарушителей.

Таким образом, требуется найти минимум функции:

$$F(\lambda_1, \lambda_2, \dots, \lambda_m) = \sum_{i=1}^m \left(\frac{1}{2} (\lambda_1 Q_{1i} + \lambda_2 Q_{2i} + \dots + \lambda_m Q_{mi}) \lambda_i + C (\bar{Q}_{1i} + \bar{Q}_{2i} + \dots + \bar{Q}_{mi}) \lambda_i - \lambda_i \right) \quad (2.2)$$

при ограничении $\sum_{i=1}^m y_i \lambda_i + C \sum_{i=1}^n y_i = 0$.

Введем дополнительную переменную λ и, используя функцию Лагранжа, найдем её условный экстремум.

$$\begin{cases} \frac{dF}{d\lambda_i} = \lambda_i Q_{ii} + \frac{1}{2} \sum_{j=1, j \neq i}^m \lambda_j (Q_{ji} + Q_{ij}) + C \sum_{j=1}^n \bar{Q}_{ji} - 1 + \lambda y_i \\ \frac{dF}{d\lambda} = \sum_{i=1}^m y_i \lambda_i + C \sum_{i=1}^n y_i \end{cases} \quad (2.3)$$

$$\frac{d^2 F}{d\lambda_i^2} = Q_{ii} \quad (2.4)$$

Получаем $d^2 F = \sum_{i=1}^m Q_{ii} d\lambda_i^2$, тогда $d^2 F > 0$. Таким образом, в точке, которая удовлетворяет системе (2.3), находится условный локальный минимум.

Найдем её:

$$\begin{cases} \lambda_i Q_{ii} + \sum_{i=1, i \neq j}^m \lambda_j Q_{ij} + C \sum_{j=1}^n \bar{Q}_{ji} - 1 + \lambda y_i = 0 \\ \sum_{i=1}^m y_i \lambda_i + C \sum_{i=1}^n y_i = 0 \end{cases} \quad (2.5)$$

$$\begin{cases} \sum_{j=1}^m \lambda_j Q_{ij} + C \sum_{j=1}^n \bar{Q}_{ji} - 1 + \lambda y_i = 0, \forall i \in [1, \dots, m] \\ \sum_{i=1}^m y_i \lambda_i + C \sum_{i=1}^n y_i = 0 \end{cases} \quad (2.6)$$

$$\begin{cases} \sum_{j=1}^m \lambda_j Q_{ij} + \lambda y_i = 1 - C \sum_{j=1}^n \bar{Q}_{ji} = 0, \forall i \in [1, \dots, m] \\ \sum_{i=1}^m y_i \lambda_i = -C \sum_{i=1}^n y_i \end{cases} \quad (2.7)$$

Решение этой системы дает нам вектор $(\lambda, \lambda_1, \dots, \lambda_m)$, из которого получаем нужный нам вектор $\lambda_S = (\lambda_1, \dots, \lambda_m)$.

Система (2.7) – это система $m + 1$ переменных от $m + 1$ неизвестных. Поэтому можно предположить, что метод будет эффективен при небольшом количестве опорных векторов. В программной реализации систему линейных уравнений решаем методом Гаусса с выбором главного элемента [7].

3. Экспериментальная оценка алгоритма

Для проведения экспериментальной оценки алгоритма была разработана программа в среде программирования Borland Delphi 7 и подготовлено несколько тестов различной размерности с разным количеством опорных векторов. На каждом тесте измерялось время работы алгоритма с соблюдением следующих условий:

- тесты предварительно загружались в оперативную память (в виде массива), время считывания в массив не учитывалось;
- на каждом тесте алгоритм запускался 5 раз, время выбиралось наименьшее.

Для сравнения с другими алгоритмами была использована реализация алгоритма SMO под названием SMOlight.

В ходе тестирования были получены следующие результаты (см. таблицу).

Как и предполагалось, алгоритм показывает хорошие результаты на тестах с небольшим ко-

Размерность пространства	Количество векторов	Количество опорных векторов	Время выполнения INCAS (мс)	Время выполнения SMO ^{light}
90	500	3	незначит.	2
90	500	150	14	3
400	500	2	незначит.	3
400	500	150	98	3
200	8000	10	132	128
200	8000	1500	224	142
700	8000	10	407	341
700	8000	1500	1048	402

личеством опорных векторов. Это обуславливается тем, что на каждом шаге необходимо решать систему линейных уравнений, размерность примерно равную количеству опорных векторов. В случае большого количества опорных векторов время работы алгоритма существенно увеличивается за счет решения системы линейных уравнений.

Заключение

Построенный алгоритм показывает высокую эффективность в тех случаях, когда множество опорных векторов и размерность пространства невелики. Метод прост в реализации и может использоваться для решения небольшого размера задач. При этом мини-задача была решена отличным от автора метода способом. Полученная система позволяет использовать для решения задачи оптимизации известные методы решения систем линейных уравнений, причем её коэффициенты достаточно просты для численной реализации, а также зависят от скалярных произведений, а не от самих векторов. Поэтому не теряется уникальная простота для перехода в другие пространства.

Примечания

1. *Vapnik, V. N.* The Nature of Statistical Learning Theory [Text] / V. N. Vapnik. N.Y.: Springer-Verlag, 1995.
2. *Osuna, E.* An improved training algorithm for support vector machines [Text] / E. Osuna, R. Freund, F. Girosi // Neural Networks for Signal Processing. 1997. VII.
3. *Scheinberg, K.* An efficient implementation of an active set method for SVMs [Text] / K. Scheinberg // Journal of Machine Learning Research. 2006. № 7. P. 2237–2257.
4. *Препарата, Ф.* Вычислительная геометрия [Текст] / Ф. Препарата, М. Шеймос. М.: Мир, 1989.
5. *Онегов, В. А.* Исследование операций. Задачи, методы, алгоритмы [Текст] / В. А. Онегов. Киров: Изд-во ВГПУ, 2001.
6. Исследование операций в экономике [Текст] / под ред. Н. Ш. Кремера. М.: ЮНИТИ, 2006.
7. *Фадеев, Д. К.* Вычислительные методы линейной алгебры / Д. К. Фадеев, Д. Н. Фадеева. М.: Физматгиз, 1963.

Е. А. Васенина

О ПРОБЛЕМЕ ИНТЕЛЛЕКТУАЛЬНОГО ВОСПИТАНИЯ ШКОЛЬНИКОВ В ПРОЦЕССЕ ОБУЧЕНИЯ ИНФОРМАТИКИ

Данная статья посвящена характеристике концепции обучения информатике, ориентированной на интеллектуальное воспитание школьников.

Вступление общества в информационную фазу развития определяет качественно новый уровень требований, которые общество предъявляет к своим членам, а следовательно, к качеству и уровню их подготовки в процессе воспитания и образования.

Главной чертой, которая отличает информационное общество, является значительное «повышение роли и степени воздействия интеллектуальных видов деятельности на все стороны его жизни» [1]. Во всех общественно значимых видах человеческой деятельности требуется «все более полное использование достоверного, исчерпывающего и своевременного знания» [2]. А значит, к человеку, живущему в таком обществе, предъявляются соответствующие требования.

С одной стороны, это *владение инструментарием интеллектуального труда*, инфраструктурой средств накопления, хранения, обработки и передачи информации. Умение свободно применять в своей профессиональной работе средства ИКТ для многих сфер деятельности стало непременным условием профессиональной компетентности. Возможность свободного доступа отдельного человека к информационному ресурсу, накопленному человечеством, простота обращения к информационным источникам, интенсивность информационного обмена значительно повышает интеллектуальный потенциал общества в целом.

С другой стороны, это определенный *уровень способностей к интеллектуальной работе*, который позволяет эффективно распорядиться полученным информационным ресурсом. Важную роль здесь играет способность к анализу и оценке информации, к установлению связей и причинно-следственных отношений, способность к формированию суждений и принятию обоснованных решений, умение планировать деятельность по их реализации, прогнозировать ее результаты, просчитывать развитие ситуации с учетом различных факторов, способность к порождению нового знания.

А. А. Кузнецов в статье [3] со ссылкой на А. П. Ершова отмечает «принципиальный мо-

ВАСЕНИНА Елена Александровна – кандидат педагогических наук, доцент по кафедре информатики и методики обучения информатике ВятГГУ
© Васенина Е. А., 2008

мент», который «заключается в том, что социотехническая революция, в частности, широкое распространение информационных технологий, является внешней стороной информационной цивилизации. Главное же ее содержание состоит в новом этапе *интеллектуального развития* (курсив наш. – Е. В.), когда происходит “философское и конкретное научное осмысление роли информации в естественных и социальных процессах”» [4].

Таким образом, подготовку человека к жизни, труду и продолжению образования в условиях информационного общества следует вести в двух направлениях.

1. Формирование системы знаний, составляющих основу современной информационной картины мира; формирование общеучебных умений, универсальных способов деятельности и ключевых компетенций в области информатики; приобретение опыта применения компьютера и других средств информационно-коммуникационных технологий для решения разнообразных задач по работе с информацией.

2. Развитие интеллектуальных способностей личности, раскрытие ее интеллектуального потенциала, формирование стиля и качеств мышления, востребованных в обществе, в котором информационная деятельность является ведущим видом деятельности.

Если важность первого направления осознается буквально всеми – от ученых и администраторов до школьников и их родителей – и для его реализации прилагаются серьезнейшие усилия на всех уровнях, то значимость второго направления признается и декларируется, но реализуется далеко не в полной мере. Это определяет актуальность проблематики, связанной с ростом и развитием интеллектуальных сил подрастающего поколения и с активным педагогическим воздействием на этот процесс.

Одной из основных причин, обуславливающих недостаток внимания к проблемам интеллектуального развития школьников, является *отсутствие глубинной внутренней мотивации* как со стороны педагогов, так и со стороны обучаемых, которое является отражением неоднозначности признания интеллекта как ценности на всех уровнях общества.

Действительно, с одной стороны, общество, в особенности на информационной стадии своего развития, нуждается в значительном повышении интеллектуального уровня *большинства* своих членов, поскольку только самостоятельно мыслящие, высокоинтеллектуальные люди способны генерировать идеи, создавать новое знание, могут правильно и эффективно организовать деятельность самого разного характера и, в конечном итоге, создать интеллектуальный продукт,

ту самую интеллектуальную собственность, которая становится во всем мире основным видом собственности.

Также повышение общего интеллектуального уровня членов общества дает снижение уровня агрессивности, особенно ее немотивированных и неосмысленных проявлений. Развитый умственный опыт ведет к осознанию этических норм как ценности жизни в социуме, позволяет уйти от низменных проявлений, поддерживает человеческое в человеке.

Не менее ценен интеллектуальный ресурс для отдельного человека, поскольку он определяет его успешность в профессии, позволяет противостоять внешним влияниям, обеспечивает самодостаточность и личную свободу, помогает делать верный выбор, строить отношения с внешним миром и в итоге определяет судьбу.

С другой стороны, практически все перечисленные достоинства, казалось бы, столь очевидные, вполне могут быть оспорены. В общественном сознании причудливым образом соединяется уважение и настороженность по отношению к интеллекту и его носителям, в особенности к проявлениям интеллектуальной одаренности. Мы осознаем, что для успеха общего дела необходимо участие в нем высокоинтеллектуального человека, но в то же время ощущаем зависимость от него в интеллектуальном плане, что, в свою очередь, порождает страхи, обиды и комплексы. К тому же для управления самостоятельно мыслящими людьми требуется на порядок более высокий уровень принимаемых управленческих решений и качество совершаемых управляющих воздействий.

В школе все эти противоречия в отношении к интеллекту проявляются весьма ярко. Взрослый человек достаточно точно оценивает свой интеллектуальный уровень и свой интеллектуальный потолок и, как правило, старается его не афишировать – или из боязни потери авторитета, статуса, уважения, или, наоборот, стараясь избежать обид и неприязни со стороны окружающих. Но если взрослые уже умеют скрывать свои эмоции, будь то ревнивое чувство к чужому интеллектуальному превосходству или раздражение проявлениями чужой интеллектуальной недостаточности, то дети проявляют свои чувства открыто и весьма категоричны в оценках и отношениях.

В этих обстоятельствах особую важность приобретает сочетание педагогической заботы об интеллектуальном развитии учеников с заботой о воспитании способности к нравственному контролю интеллектуальных проявлений, ответственности человека как за состояние своего интеллекта, так и за общественно значимое использование своих интеллектуальных способностей и

достижений, формировании мотивации к своему интеллектуальному совершенствованию.

К сожалению, несмотря на оптимистические декларации, школа страдает явным недостатком внимания к вопросам интеллектуального развития учеников. Школа контролирует и оценивает учебные достижения своих учеников – это есть видимый результат ее работы. Гораздо в меньшей степени ее интересует, в какой степени достижения ученика являются результатом формального заучивания и натаскивания, а в какой – результатом интеллектуального развития его личности. Кроме того, школа не без основания считает, что интеллектуальные способности, в основном, либо определены на генетическом уровне, либо развиваются в раннем возрасте. И за первое, и за второе ответственны родители, а школа работает с тем, что досталось, и ее ответственность за интеллектуальное развитие не так уж велика.

Кроме того, организация учебного процесса в целом и разрабатываемые методики обучения далеко не всегда предполагают интеллектуальное развитие школьников в качестве образовательного приоритета, даже если речь идет о предметах, роль которых в становлении и развитии мышления ребенка достаточно велика – к их числу, в частности, относится информатика. В то время как наличие методик, ориентированных на интеллектуальное развитие учащихся в условиях массовой школы, могло бы, с одной стороны, определить ценностный ориентир для учителя и поставить заботу о развитии мышления учеников в ряд основных приоритетов образования, а с другой – вооружить учителя методическим инструментарием для воздействия на интеллектуальную сферу учащихся.

Обычно, обсуждая проблемы роста интеллектуальных сил и возможностей подрастающего поколения, педагоги и психологи используют термин «развитие» – развитие мышления, развитие умственных способностей, интеллектуальное развитие.

Само понятие «развитие», согласно Словарю Ожегова, определяется как «процесс перехода из одного состояния в другое, более совершенное, переход от старого качественного состояния к новому качественному состоянию, от простого к сложному, от низшего к высшему» [5]. Следовательно, интеллектуальное развитие подразумевает не только количественное приращение отдельных когнитивных свойств, но и качественные изменения, происходящие с интеллектом в целом как со сложной системой взаимосвязанных, влияющих друг на друга характеристик.

Проблемами развития способностей личности в процессе обучения и воспитания, причем, по

преимуществу, интеллектуальных, умственных способностей, занимались многие психологи и педагоги, среди которых как создатели теоретических основ, так и творцы педагогических систем, на практике воплощающих идеи развивающего обучения. Это ставшие уже классическими труды Л. С. Выготского, Ж. Пиаже, М. Вергеймера, С. Л. Рубинштейна, А. Н. Леонтьева, П. Я. Гальперина, Н. Ф. Талызиной, Л. В. Занкова, В. В. Давыдова, Д. Б. Эльконина и многих других.

Отметим, что такой переход в иное, «более совершенное состояние» осуществляет сам ученик, процесс развития предполагает его личностную активность. Однако если учителя также рассматривать как активную фигуру, мотивированно и целенаправленно влияющую на данный процесс, то разумно будет использовать другой термин – *интеллектуальное воспитание*. Данное понятие Педагогической энциклопедией трактуется как «важнейшая сторона подготовки к жизни и труду подрастающих поколений, заключающаяся в руководстве развитием интеллекта и познавательных способностей путем возбуждения интереса к интеллектуальной деятельности, вооружения знаниями, методами их добывания и применения на практике, привития культуры интеллектуального труда» [6].

В свое время этот термин не получил широко распространения, но сейчас становится все более востребованным и приобретает новое звучание, ибо предполагает *активное целенаправленное воздействие* на интеллектуальную сферу ребенка.

М. А. Холодная широко использует понятие интеллектуального воспитания в исследовании психологии интеллекта [7], особенно в той его части, которая посвящена преломлению психологической теории в школьной практике. Она рассматривает ряд методических моделей, построенных с учетом психологических механизмов умственного развития учащихся, среди которых предлагает разработанную ею самой в сотрудничестве с Э. Г. Гельфман, Л. Н. Демидовой и другими *обогащающую методическую модель*, «основным назначением которой является интеллектуальное воспитание учащихся за счет актуализации и усложнения ментального (умственного) опыта ребенка» [8].

М. А. Холодная определяет *интеллектуальное воспитание* как *создание условий для совершенствования интеллектуальных возможностей каждого ребенка за счет обогащения его ментального опыта на основе индивидуализации учебного процесса и внешкольной деятельности*.

Также она отмечает, что *интеллектуальное воспитание* есть *форма организации учебно-воспитательного процесса, которая обеспечивает*

оказание каждому ученику индивидуализированной педагогической помощи с целью развития его интеллектуальных возможностей.

Обозначим еще один аспект проблемы интеллектуального воспитания.

Трудность оценки эффективности работы учителя по интеллектуальному воспитанию связана еще и с тем, что ученики изначально имеют разный интеллектуальный потенциал. Но каким бы ни был диапазон интеллектуального ресурса каждого отдельного ребенка, интересы и его самого, и общества в целом требуют, чтобы были предприняты усилия для его интеллектуального роста по максимуму имеющихся возможностей.

Таким образом, можно предложить еще одно определение понятия интеллектуального воспитания.

Интеллектуальное воспитание – глубоко мотивированная, направленная деятельность педагога, имеющая целью интеллектуальное развитие каждого ученика в соответствии с его интеллектуальным потенциалом и осуществляемая во взаимодействии и сотрудничестве с учеником.

Данное определение не претендует на полноту и всеобъемлющий характер, но отражает соотношение понятий *интеллектуальное воспитание* и *интеллектуальное развитие*, а также раскрывает роль учителя в интеллектуальном воспитании.

Говоря об интеллектуальном воспитании, необходимо связывать его с другими сторонами воспитательного процесса, в особенности с нравственным воспитанием. Начало отношения к интеллекту как к ценности в обществе на всех его уровнях, бесспорно, закладывается в школе, где и требуется формировать уважительное, доброжелательное отношение к любым интеллектуальным достижениям, будь то проявление интеллектуальной одаренности или стоившее больших усилий небольшое интеллектуальное приращение. Формирование положительной мотивации, утверждение в сознании учеников значимости интеллектуального роста также является важной задачей интеллектуального воспитания. Ее решение напрямую связано с нравственным воспитанием, поскольку касается взаимодействия человека с окружающими людьми, с обществом в целом. Назовем эту часть интеллектуального воспитания его мотивационно-нравственной составляющей.

Среди общеобразовательных дисциплин курса средней школы информатику можно выделить как предмет, предоставляющий большие возможности для решения задач интеллектуального воспитания. Данная учебная дисциплина имеет мощ-

ный образовательный потенциал, который определяется общеобразовательной значимостью изучения области деятельности, связанной с информационными процессами в живой природе, обществе, технике [9]. В качестве одной из основных целей ее изучения выдвигается развитие мышления, формирование способов и приемов умственной деятельности, общеинтеллектуальных умений, овладение современными методами научного познания, поскольку освоение учениками этого предмета «оказывает влияние на развитие теоретического, творческого мышления, на формирование нового типа мышления... направленного на выбор оптимальных решений» [10]. Изучая информатику, школьник изучает область человеческой деятельности, связанную реализацией информационных процессов, и на практике убеждается в ценности работы с информацией, т. е. ценности умственного труда, учится уважать продукт этого труда и его создателя – человека, наделенного интеллектом.

Проблемам становления информатики как общеобразовательной дисциплины, определению приоритетных целей ее изучения и формированию содержания школьного курса информатики посвящены работы С. А. Бешенкова, А. А. Кузнецова, К. К. Колина, В. С. Леднева, С. М. Окулова, Е. А. Ракитиной и других.

Кроме того, в качестве инструментария в обучении информатике выступают средства информационно-коммуникационных технологий, и компьютер как средство обучения, значительно увеличивающее количество и интенсивность информационных потоков и активизирующее обратные связи в учебном процессе, также имеет достаточно высокий образовательный потенциал, что при соответствующей организации обучения также может положительно влиять на достижение целей интеллектуального воспитания.

Все вышесказанное определяет цель проведенного автором исследования по очерченной проблематике как *определение возможностей информатики в качестве общеобразовательной дисциплины и компьютера как качественно нового средства обучения для решения задач интеллектуального воспитания и разработку концепции обучения информатике, ориентированную на интеллектуальное воспитание учащихся.*

При этом предполагается, что предложенная концепция будет эффективной, если:

– в качестве психологически мотивированных приоритетов в организации учебного процесса, ориентированного на интеллектуальное воспитание школьников, определить индивидуализацию обучения; проблемно-деятельностный подход к обучению; самостоятельность в обучении; ориентацию учебного процесса на формирование критичности мышления;

– в качестве базовых идей в основу предлагаемой концепции обучения информатике положить:

1) интеллектуальное развитие личности как приоритетное направление в обучении информатике;

2) проблемно-деятельностный подход как методологическую основу обучения информатике;

3) индивидуализацию обучения информатике; – определить следующие направления реализации базовых идей:

1) задачный подход в обучении информатике;

2) эксперимент, исследование, творчество;

3) самостоятельность в обучении информатике;

4) взаимодействие участников учебного процесса на уроке информатики;

5) программирование как деятельность, ориентированную на развитие интеллекта;

– выделить и предложить пути реализации в учебном процессе особенностей структурной организации обучения информатике на следующих уровнях:

1) особенности структурной модели содержания обучения информатике;

2) структурные особенности организации освоения материала на уроке информатики;

3) структурные особенности организации деятельности учеников на уроке информатики.

Для достижения поставленной цели, проверки и обоснования правильности высказанных предположений требовалось решить следующие задачи:

1. Определить понятие интеллектуального воспитания, рассмотреть, как оно соотносится с понятием интеллектуального развития.

2. На основе анализа психолого-педагогической литературы определить психологически мотивированные приоритеты в организации учебного процесса, ориентированного на интеллектуальное воспитание школьников.

3. Исследовать образовательный потенциал информатики как общеобразовательной дисциплины и компьютера как средства обучения, значительно увеличивающего количество и интенсивность информационных потоков и активизирующего обратные связи в учебном процессе.

4. Характеризовать содержательные блоки курса информатики с точки зрения их ценности для решения задач интеллектуального воспитания.

5. Выделить особенности компьютерного урока информатики.

6. Разработать базовые идеи, направления их реализации и особенности структурной организации процесса обучения информатике в рамках концепции обучения информатике, ориентированной на интеллектуальное воспитание.

7. Характеризовать основные методы применения компьютера в обучении информатике.

В ходе проведенного исследования были получены следующие результаты:

1. Определено понятие интеллектуального воспитания, рассмотрено его соотношение с понятием интеллектуального развития.

2. Выделена мотивационно-нравственная составляющая интеллектуального воспитания, определены личностные качества, формирование которых входит в круг ее задач, а также направления и приемы деятельности учителя по ее реализации.

3. Определены психологически мотивированные приоритеты в организации учебного процесса, ориентированного на интеллектуальное воспитание школьников: индивидуализация обучения; проблемно-деятельностный подход к обучению; самостоятельность в обучении; ориентация учебного процесса на формирование критичности мышления.

4. Раскрыты возможности информатики как общеобразовательной дисциплины, в процессе изучения которой формируются качества, отвечающие критериям интеллектуальной воспитанности.

5. Раскрыты возможности информатики как дисциплины, изучение которой позволяет соединить усвоение фундаментальных теоретических знаний с деятельностным подходом к обучению, с практико-ориентированным обучением. Обоснована ценность компьютерной программы как информационного объекта, который позволяет в процессе практико-преобразовательской деятельности исследовать фундаментальные понятия, принципы и закономерности.

6. Рассмотрено влияние компьютера как качественно нового средства обучения на организацию учебного процесса через увеличение количества и многообразия направлений информационных потоков в обучении и через активизацию обратных связей в процессе обучения, рассматриваемого как с позиций управления (повышение эффективности управляющих воздействий учителя в ходе познавательной деятельности учеников), так и с позиций познания (приближение обучения к реальному процессу познания).

7. Выполнена характеристика содержательных блоков курса информатики с точки зрения их ценности для решения задач интеллектуального воспитания, а также предложены подходы к их изучению, повышающие эту ценность.

8. Выделены мыслительные умения и интеллектуальные качества, формируемые на разных этапах деятельности по разработке и составлению программ.

9. Выделены особенности компьютерного урока информатики в направлении повышения уровня

самостоятельности и творческой активности в работе учащихся, в организации общения, коллективной деятельности и учета эмоционального состояния учеников на компьютерном уроке информатики, в направлении индивидуализации обучения.

10. Разработана концепция обучения информатике, ориентированной на интеллектуальное воспитание, на уровне базовых идей, направленных их реализации, а также особенностей структурной организации процесса обучения информатике.

11. Выполнена характеристика основных методов применения компьютера в обучении.

12. Предложено описание фрагментов уроков, являющихся примерами практического применения описанных методов.

Примечания

1. Концепция информатизации образования [Текст] // Информатика и образование. 1990. № 1. С. 3.

2. Там же.

3. Кузнецов, А. А. Современный курс информатики: от элементов к системе [Текст] / А. А. Кузнецов, С. А. Бешенков, Е. А. Ракитина // Информатика и образование. 2004. № 1. С. 5.

4. Ершов, А. П. Информатизация: от компьютерной грамотности школьников к информационной культуре общества [Текст] / А. П. Ершов // Коммунист. 1988. № 2. С. 82–92.

5. Ожегов, С. И. Словарь русского языка [Текст] / С. И. Ожегов; под ред. Н. Ю. Шведовой. М.: Рус. яз., 1988.

6. Педагогическая энциклопедия [Текст] / под ред. И. Я. Каирова, Ф. Н. Петрова. М.: Сов. энцикл., 1966.

7. Холодная, М. А. Психология интеллекта. Парадоксы исследования [Текст] / М. А. Холодная. СПб.: Питер, 2002.

8. Там же. С. 212.

9. Кузнецов, А. А. О Концепции содержания образовательной области «Информатика» в 12-летней школе [Текст] / А. А. Кузнецов // Информатика и образование. 2000. № 7.

10. Концепция содержания образовательной области «Информатика» в 12-летней школе [Текст] // Информатика и образование. 2000. № 2.

Е. А. Васенина, Д. А. Хомякова

О СТРУКТУРИРОВАНИИ БАНКА ЗАДАЧ В ЦЕЛЯХ ИНДИВИДУАЛИЗАЦИИ ПРОЦЕССА ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Данная статья посвящена описанию процесса создания структурированной системы задач для реализации индивидуального подхода к обучению программированию.

Образовательный потенциал информатики, с одной стороны, определяется общеобразовательной значимостью изучения области деятельности, связанной с информационными процессами в живой природе, обществе, технике, а с другой стороны, связан с возможностью освоения фундаментальных теоретических понятий в процессе практико-преобразовательской деятельности. Работа с информационным объектом, в качестве которого выступают разрабатываемая программа, исследуемая информационная среда или создаваемый в этой среде информационный продукт: текстовый документ, электронная таблица, электронная база данных (информационная система), позволяет соединить теоретическую деятельность (умственные действия) с деятельностью практической, в ходе которой ученик может преобразовывать данный объект: манипулировать им, помещать в некоторые условия, наблюдать реакцию, отслеживать происходящие изменения. Ученик получает возможность изучать фундаментальные основы не абстрактно, не умозрительно, но в процессе практической деятельности, видеть, как они работают, и осознавать их ценность. Все это способствует его интенсивному интеллектуальному развитию.

Сочетание практических действий с информационным объектом и формирование фундаментального знания осуществляется постепенно и возможно лишь вследствие целенаправленной работы преподавателя. Для достижения целей интеллектуального воспитания такую работу следует максимально индивидуализировать.

В обучении программированию основой индивидуализации является система задач, предлагаемая учащемуся для решения. Здесь использование средств ИКТ значительно облегчает работу преподавателя по подбору задач и составлению индивидуальных заданий для каждого студента.

ВАСЕНИНА Елена Александровна – кандидат педагогических наук, доцент по кафедре информатики и методики обучения информатике ВятГГУ
ХОМЯКОВА Дарья Александровна – ассистент по кафедре информатики и методики обучения информатике ВятГГУ

© Васенина Е. А., Хомякова Д. А., 2008

В качестве вспомогательной информационной среды, поддерживающей процесс обучения программированию, целесообразно разработать электронное учебное средство для организации самостоятельной работы по решению задач на ЭВМ. Основой его информационно-методического наполнения является банк задач.

Описанию процесса формирования банка задач уделим особое внимание. Как таковую базу задач создать легко – существует множество электронных и печатных источников задач. Трудность составляет структурирование собранных задач, их преобразование в целостную систему, для чего задачи требуется классифицировать.

Проблеме классификации задач в методической, психологической, кибернетической литературе посвящено немало работ. Анализ различных исследований данной темы приведен в работе Е. В. Разовой, посвященной методике изучения теории чисел и криптографии на основе задачного подхода (Разова Е. В. Построение методики обучения элективному курсу информатики «Теория чисел и криптография» на основе задачного подхода: Дис. ... канд. пед. наук. М., 2004. 99 с.). В частности, приводятся следующие классификации.

- По величине проблемности (У. Рейтман, Ю. М. Колягин): стандартные, обучающие, поисковые, проблемные.

- По дидактическим функциям (К. И. Нешков, А. Д. Семушин, Ю. М. Колягин, Е. И. Лященко и др.): дидактические, познавательные, развивающие.

- По месту в процессе обучения (Ю. К. Бабанский): задачи, стимулирующие учебно-познавательную деятельность; задачи, организующие и осуществляющие учебно-познавательную деятельность; задачи, в процессе выполнения которых осуществляется контроль и самоконтроль эффективности учебно-познавательной деятельности.

- По характеру деятельности (Ю. К. Бабанский): репродуктивные, поисковые, творческие задачи.

- По возрастанию когнитивной сложности и операциональной ценности (Д. Толлингерова). Задачи разделяются по операционной структуре, то есть по операциям, необходимым для их выполнения.

- По характеру требования задачи и совокупности действий, необходимых для ее решения.

На основе перечисленных выше классификаций выработана классификация, которая была применена авторами для структурирования созданного банка задач. Данная классификация включает группирование и маркировку задач по трем критериям:

- по умениям, которые формируются в процессе решения задачи;

- по видам действий, которые требуется выполнить для получения решения;

- по функциям задач в процессе обучения.

Рассмотрим более подробно каждый из критериев классификации задач.

1. Классификация задач на основе умений, которые формируются в процессе их решения

Чтобы обеспечить усвоение учащимся знаний и формирование у него навыков работы с изучаемыми структурами, требуется для каждой задачи определить перечень базовых знаний и основных умений, формированию которых будет способствовать ее решение.

В качестве примера такой классификации задач приведем перечень понятий раздела «Структура данных массив», которые усваивает учащийся, и перечень умений, которыми он должен овладеть в рамках изучения данной темы:

- умение формировать массив;
- умение выводить элементы массива;
- умение преобразовывать элементы массива;
- умение находить сумму и произведение элементов массива;

- умение выбирать элементы по заданному условию и производить действия над ними: вывод, перестановка, нахождение количества, суммы, произведения;

- умение поиска экстремума и его индекса в массиве;

- умение поиска заданного элемента в массиве;

- умение «удаления»/«вставки» элементов массива (организация сдвига);

- умение сортировать элементы массива простыми методами;

- умение сортировать элементы массива методами быстрой сортировки;

- умение применять массив для работы с длинными числами.

Таким образом, для каждого теоретического подраздела можно создать набор *стержневых* задач. Под стержневыми задачами будем понимать такие задачи, набор которых обеспечивает полноту раскрытия и освоения изучаемого материала, освоение методов и способов деятельности, характерных для данной темы.

Эти базовые знания и умения должны стать личным приобретением каждого учащегося в результате решения задач. Но индивидуальные способности студентов различаются, поэтому преподавателю, чтобы достичь цели обучения и следовать при этом выбранному индивидуальному подходу, следует предъявлять каждому учащемуся свою задачу с посильным для него уровнем сложности. В то же время эта задача обязательно должна быть стержневой, то есть ее решение

должно обеспечивать формирование определенного умения.

С целью реализации индивидуального подхода к обучению введем термин «Клон задачи». Он появился в результате разработки метода «клонирования задач», который состоит в следующем. Задача базового уровня, обязательная для понимания каждым студентом и взятая в качестве стержневой, модифицируется различными способами. В результате «клонирования» получаем целый массив задач, так называемых «клонов», формирующих одно и то же знание и служащих отработке одного и того же умения, но предназначенных для решения различными учащимися. В зависимости от способа клонирования задачи в массиве разделяются на два уровня.

Первое подмножество задач образовано при помощи изменения формулировки. Такие задачи отличаются незначительно внутренне, то есть по сути, но совершенно непохожи внешне. Например, для задач на нахождение экстремума в массиве клонами одинакового уровня являются следующие: «В числовом массиве записаны значения роста учеников 10А класса. Найти самого высокого ученика в классе» и «В массиве T [1.. 31] хранятся значения дневных температур июля. Определить, какой день в июле был самым жарким».

Второй способ «клонирования» состоит в изменении самой задачи, ее усложнении путем добавления некоторых условий или ограничений. Эти задачи по уровню сложности будут на порядок выше, хотя по-прежнему остаются стержневыми в рамках рассматриваемого теоретического подраздела. Например, задача на поиск экстремума в массиве может быть дополнена ограничением на поиск первого экстремума из нескольких, на поиск первого максимума и последнего минимума одновременно.

Итак, наличие множества задач, «клонированных» разными способами, позволяет преподавателю без лишних усилий и наиболее точно подобрать задачу для каждого студента. При этом задача будет соответствовать его возможностям, активизирует зону ближайшего развития и послужит наиболее подходящим материалом для формирования умений и развития требуемых навыков.

2. Классификация задач по видам действий, которые требуется выполнить для получения решения

Вспомним, что реализация задачного подхода возможна только при вовлечении учащегося в деятельность. От того, какие действия выполняются и каких мыслительных операций они требуют, зависит, насколько интенсивно происходит интеллектуальное развитие личности учащегося. Учитывать этот аспект помогает данная классификация. Принадлежность к одной из групп классификации отражается в формулировке задачи.

Задачи на трассировку алгоритма
Данные задачи следует предлагать к решению после разбора задач по новой тематике. Основная цель – закрепление изученного материала.

Формулировка задач данного типа в общем случае звучит следующим образом: «Что является результатом работы приведенного фрагмента программы?».

Возможны вариации по степени сложности:

- Трассировать алгоритм и определить, что делает данная программа (с использованием и без использования входных данных).

- Трассировать алгоритм на нескольких вариантах предложенных входных данных (возможность рассмотреть все «крайние» случаи и в полной мере осознать логику решения задачи).

Задачи на поиск ошибок в алгоритме

В зависимости от цели, которая должна быть достигнута в результате решения учеником задачи, в алгоритме могут быть предъявлены ошибки разных типов:

- Ошибки «не по существу». К этому типу относятся синтаксические ошибки, незначительные нарушения в логике алгоритма, которые не влекут за собой серьезных изменений. Такие задачи могут быть использованы при первоначальном знакомстве с новой структурой данных с целью закрепления навыка работы с ней. Например, ошибки допускаются в описании типа или в применении стандартных функций работы с переменными данного типа.

- Ошибки в логике. Это гораздо более серьезные ошибки, которые меняют логику работы всего алгоритма. Задачи с ошибками такого типа могут быть предъявлены на завершающем этапе изучения темы с дидактической целью проверки усвоенного материала. Грамотно подобранная задача с ошибками в логике также может послужить источником новых знаний, если при их исправлении ученик освоит новый способ действия с уже изученными структурами или получит новый вариант известной ему логики решения задачи. Цель использования таких задач – обучающе-развивающая.

Задачи на составление алгоритма

Наиболее распространенный тип задач, на практике используется гораздо чаще, чем задачи других типов. Задачи могут варьироваться по степени сложности:

- Если учащегося просят составить алгоритм задачи, логика решения которой ему известна (разобрана с учителем на подобном примере), при написании программного кода задействуются механизмы воспроизведения уже известных фактов. Это так называемая работа по образцу. Основная цель – отработка навыка.

- Второй подвид задач на составление алгоритма – задачи, решение которых требует не просто применить известный алгоритм, а сделать в нем некоторые изменения, доработать программу, адаптировать ее к конкретным условиям задачи.

- Если же задача формулируется так, что способ ее решения учащемуся не известен, ему требуется провести анализ имеющихся в инструментарии способов действий и на его основе разработать свой. Здесь речь идет о непосредственном развитии интеллекта в ходе обобщения имеющегося опыта, составления нового, ранее не известного учащемуся способа решения на основе анализа.

Задачи на модернизацию алгоритма

Как правило, решение задач данной группы требует от ученика высокого уровня владения материалом. Предполагается, что он в полной мере усвоил логику решения, представленную в данном алгоритме, и обладает достаточным набором навыков работы, чтобы вносить в алгоритм конструктивные изменения. Такие задачи можно применять для диагностики понимания учеником практически любой темы курса, от числового типа данных до методов быстрой сортировки массива.

Задачи-вопросы

Решение таких задач, как правило, не требует составления алгоритмов. В общем случае это вопросы на знание фактического материала. Задачи более высокого уровня – так называемые «аналитические вопросы» – подразумевают интеллектуальную работу учащегося по отбору и анализу имеющихся знаний с целью синтеза новых понятий и установления связей между ними в рамках изучаемой темы. Задачи-вопросы, в зависимости от степени сложности, могут быть применены на любом этапе обучения.

3. Классификация задач по функциям в процессе обучения.

Данная классификация наиболее точно отражает суть процесса обучения программированию. Она в полной мере обеспечивает реализацию задачного подхода. Напомним, что он заключается в приобретении учащимся нового знания в процессе решения некоторой проблемы. Проблема оказывается неразрешимой средствами уже имеющегося у студента инструментария знаний и способов действий. Это заставляет его разрешать противоречия между желаемым и возможным путем приобретения нового теоретического знания и его применения к решению проблемы. Очередная барьерная макропроблема ставится перед учащимся при изучении каждого нового подраздела теории.

В качестве такого препятствия выступает стержневая задача. Она предъявляется преподавателем и разбирается в ходе общего обсуждения. Преподаватель вводит новое понятие, дает некоторый объем теоретического материала.

Для того чтобы материал перешел в собственные знания учащегося, общее представление трансформировалось в полную и многогранную картину в его сознании, предлагается набор задач для обязательного самостоятельного решения. Это один или несколько «клонов» стержневой задачи, естественно, соответствующих уровню интеллектуальных возможностей обучающегося. И на этом этапе учащийся сталкивается с целым рядом новых проблем. Он затрудняется в решении задач, так как ограничен в инструментарии и знаниях. Назовем эти трудности микропроблемами. Начальные теоретические знания уже получены, можно также обратиться за подсказкой к справочному материалу обучающей системы. Но навык работы с новой структурой еще не сформирован, и задача в предъявленном виде не может быть решена. Однако заметим, что структурный стиль программирования, который закладывается в курсе обучения «Основам программирования», предполагает решение любой задачи по частям. Таким образом, каждая из микропроблем, с которой сталкивается учащийся, может быть разбита на атомарные подзадачи, которые либо уже знакомы учащемуся, либо гораздо проще в освоении, чем стержневые задачи целиком. Эти подзадачи, назовем их тренировочными, рекомендуются учащемуся как помощь, подсказка. Они не обязательны для решения, к ним учащийся обращается только при возникновении микропроблем.

Таким образом, выстраивается четкая структура, которую преподаватель использует при формировании заданий для каждого студента к определенному занятию. Она выглядит следующим образом:

- Задачи «Решаем вместе» (базовая стержневая задача (1-2), которая выступает мотивационной к изучению нового материала).

- Задачи для обязательного самостоятельного решения (разноуровневые «клоны» стержневой задачи).

- Задачи «Тренировочные».

- Задачи «Для особо хитрых» повышенного уровня сложности (для дополнительного решения).

В силу прямого назначения данной структуры (реализация индивидуального подхода) невозможно классифицировать с ее помощью все задачи отвлеченно от конкретной ситуации обучения.

Таким образом, разработаны три группы критериев классификации:

(1) Классификация задач по тематическим подразделам и формируемым умениям.

(2) Классификация задач по видам действий, которые требуется выполнить для их решения.

(3) Классификация задач по дидактическим функциям. Отметим, что одна и та же задача может выполнять разную функцию в конкретной ситуации обучения. Таким образом, классификация задач по функциям обладает внутренней динамикой. Она будет заново реализовываться в процессе непосредственной работы конкретного преподавателя по формированию заданий к каждому занятию.

Приведенная классификация задач представляется авторам достаточно простой и пригодной в качестве содержательной основы электронного учебного средства. Она осуществлена в терминологии, приближенной к конкретной деятельности преподавателя, предполагает удобство работы с задачами и минимизацию усилий преподавателя, которому требуется:

- регулярно подбирать комплекс заданий для каждого студента (группы студентов) в рамках изучаемой темы;
- при необходимости добавлять новые задачи, маркировать их и определять групповую принадлежность.

Работа с задачами на базе приведенной классификации и использование электронного учебного средства позволит индивидуализировать процесс обучения программированию.

А. С. Вахрушев, А. Н. Захарова, Н. И. Исупова

ИЗ ОПЫТА ПОДГОТОВКИ К ИНТЕРНЕТ-ТЕСТИРОВАНИЮ ПО ИНФОРМАТИКЕ

В статье приводится обобщение опыта работы со студентами в рамках подготовки к срезовому контролю знаний с помощью Интернет-тестирования.

Тестирование выпускников общеобразовательных учреждений как сравнительно новый, прогрессивный способ контроля знаний учащихся получает в последнее время все большее распространение. Так, на протяжении нескольких последних лет в практику работы нашего университета было внедрено так называемое Интернет-тестирование по различным учебным дисциплинам с целью проверки уровня подготовки студентов.

ВАХРУШЕВ Алексей Серафимович – кандидат биологических наук, доцент по кафедре прикладной информатики ВятГГУ

ЗАХАРОВА Любовь Николаевна – ассистент кафедры прикладной информатики ВятГГУ

ИСУПОВА Наталья Ивановна – кандидат педагогических наук, старший преподаватель кафедры прикладной информатики ВятГГУ

© Вахрушев А. С., Захарова А. Н., Исупова Н. И., 2008

Результаты первого тестирования студентов по дисциплине «Информатика» показали очень низкую степень владения учащимися необходимыми знаниями (в среднем от 10 до 30%). Естественно, такой показатель не устраивает ни преподавателей, ни студентов.

Стало ясно, что применение традиционных методов обучения, наряду с увеличением теоретического блока дисциплины, усилением контролирующей функции и т. д., не дает желаемого результата для решения задачи повышения уровня освоения студентами программы данной дисциплины. Нужно готовить студентов к Интернет-тестированию целенаправленно, доводя до их сведения спектр используемых в тестах вопросов и прощивая варианты возможных заданий.

С этой целью были разработаны специальные, пробные тесты по дисциплине «Информатика». Отбор материала осуществлялся на основе анализа государственных образовательных стандартов и учебно-методических комплексов данной дисциплины для различных специальностей. В результате были выделены основные типы тестовых заданий, которые естественным образом были разбиты на девять дидактических единиц (ДЕ), соответствующих основным разделам курса информатики (см. табл. 1). По каждой дидактической единице было отобрано 50–70 вопросов и заданий, которые заложены в тестовую оболочку Test Builder.

Вот лишь некоторые примеры заданий из разработанных тестов (по одному на каждую дидактическую единицу).

Задание 1. Наиболее точный смысл понятия «семантическая емкость информации» – это...

- полнота информации;
- точность информации;
- важность информации;
- смысл информации.

Задание 2. Укажите три типа шин ввода/вывода с соответствующими слотами расширения в персональном компьютере...

- шина PCI;
- шина EISA;
- шина SVGA;
- шина ISA.

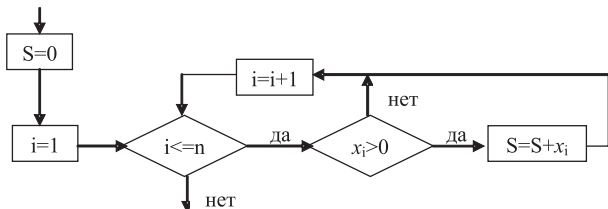
Задание 3. Из предложенных программ укажите служебные

- Format;
- Defrag;
- Paint;
- Winword.

Задание 4. Одна из математических моделей, с помощью которой может быть описана (задаана) работа уличного светофора, – это модель...

- вероятностного автомата;
- описываемая системой дифференциальных уравнений;
- описываемая системой алгебраических уравнений;
- детерминированного конечного автомата.

Задание 5. Задан одномерный массив x_1, x_2, \dots, x_n . Фрагмент алгоритма...



определяет:

- максимальный элемент массива;
- сумму отрицательных элементов;
- сумму положительных элементов;

- количество положительных элементов;
- индекс последнего положительного элемента.

Задание 6. Что такое формальные параметры подпрограммы?

- конкретное значение входных данных;
- входные данные, определяющие тип;
- идентификаторы входных данных;
- неопределенные входные данные.

Задание 7. Таблица базы данных имеет вид:

№	Пол	Возраст	Рост
1	Ж	25	1,40
2	М	20	1,65
3	М	27	1,80
4	Ж	18	1,75
5	М	35	2,00
6	Ж	20	1,64
7	Ж	18	1,70

№ п/п	Название ДЕ	Содержание раздела
1	Понятие информации. Общая характеристика процессов сбора, передачи, обработки и накопления информации	Информатика. Предмет информатики. Основные задачи информатики. Сигналы. Данные. Информация. Информационный процесс. Основные операции с данными. Кодирование. Системы счисления. Единицы представления, измерения и хранения данных. Предпосылки возникновения и история становления информатики
2	Аппаратные реализации информационных процессов	Состав и назначение основных элементов персонального компьютера. Запоминающие устройства. Устройства ввода/вывода данных. Понятие вычислительной системы
3	Программные средства реализации информационных процессов	Классификация программного обеспечения. Обзор системного и прикладного ПО. Пользовательские интерфейсы. ОС Windows. Управление файловой структурой. Службное программное обеспечение. Программное обеспечение обработки текстовых данных. Электронные таблицы. Электронные презентации
4	Модели решения функциональных и вычислительных задач	Моделирование как метод познания. Классификация структуры моделей. Классификация задач, решаемых с помощью моделей. Методы и технологии моделирования. Интеллектуальные системы
5	Алгоритмизация и программирование. Языки программирования высокого уровня	Алгоритмизация. Чтение блок-схемы алгоритма. Программирование. Эволюция языков программирования
6	Программное обеспечение и технологии программирования	Системы программирования. Структурное программирование. Чтение структурированных программ. Стратегии решения задач
7	Базы данных	Структуры данных. Базы данных; системы управления базами данных. Объекты базы данных. Чтение фрагментов базы данных
8	Локальные и глобальные сети ЭВМ	Программные и аппаратные компоненты вычислительных сетей. Принципы построения сети Интернет. Сервисы Интернет. Средства использования сетевых сервисов: Браузеры. Интернет. Почтовые программы
9	Основы защиты информации и сведений, составляющих государственную тайну; методы защиты информации	Информационная безопасность (ИБ) и ее составляющие. Компьютерные вирусы и средства антивирусной защиты. Защита от несанкционированного вмешательства. Специфика обработки конфиденциальной информации

Пользователь установил фильтр по полю «Пол»=Ж и задал условие сортировки по возрастанию поля Рост. Укажите получившийся порядок записей по их номеру (поле №)

- 1, 6, 7, 4, 2, 3, 5;
- 1, 6, 7, 4;
- 1, 6, 2, 7, 4, 3, 5;
- 4, 7, 6, 1;
- 2, 3, 5, 1, 6, 7, 4.

Задание 8. Укажите два возможных адреса электронной почты

- Mail.ru@egorov;
- http//gov.nicola;
- avgust@basa.mmm.ru/ivanov/mail;
- Abbi-qwe@mail.ru;
- avgust@basa.mmmn.ru.

Задание 9. Укажите три способа защиты информации с помощью программных и аппаратных средств:

- криптография;
- разграничение доступа к ресурсам;
- организационный контроль за работой;
- проверка на наличие вредоносных вирусов;
- введение охраны и ограничение доступа к системе.

Тестирование студентов проводится несколько раз по мере изучения дисциплины «Информатика». Во-первых, обеспечивается так называемая входная проверка с целью определения базового уровня подготовки студентов. В ходе дальнейшего обучения тесты предлагаются после изучения каждого раздела, соответствующего некоторой дидактической единице. Завершается курс итоговым тестом, охватывающим весь пройденный материал.

Результатом такой работы явилось повышение уровня овладения студентами необходимыми знаниями в области данной дисциплины (процент студентов, освоивших все дидактические единицы, по результатам последних тестирований составил в среднем 80–95%).

Кроме того, хотелось бы отметить еще некоторые положительные аспекты применения данной методики подготовки студентов.

Во-первых, регулярная проверка и самопроверка студентов с помощью разработанных тестов является дополнительным стимулом к более систематической работе по усвоению изучаемого материала. Во-вторых, анализ результатов промежуточных тестирований позволяет выявить основные ошибки, недочеты, пробелы в знаниях учащихся и вовремя попытаться их устранить. В-третьих, такая работа помогает преподавателю в решении вопросов промежуточной и итоговой аттестации студентов, которые могут выс-

тавляться по результатам тестовых работ. Наконец, в ходе применения подобного опыта было замечено повышение интереса студентов к изучению информатики вообще и таких ее разделов, как алгоритмизация и программирование, в частности.

В. П. Дьячков

РАЗРАБОТКА ОБУЧАЮЩИХ ПРОГРАММ ПО ИНФОРМАТИКЕ С ИСПОЛЬЗОВАНИЕМ РАЗЛИЧНЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ

В статье приведены примеры разработки обучающих программ, используемых в практике преподавания.

Обучающая программа – это программное средство, предназначенное для передачи знаний студентам и формирования умений и навыков у них по определённой дисциплине. Обучающие программы можно условно разделить на обучающие игры, обучающие среды и обучающие программы.

Обучающие программы отличаются от обучающих игр и обучающих сред тем, что они сначала фиксируют предполагаемый начальный уровень знаний обучаемого, затем записывают цели обучения, а потом в соответствии с этими целями создается сценарий обучающей программы. В некоторых программах обучаемый может согласовать режим овладения знаниями, умениями и навыками с индивидуальными учебными задачами, выбрав из меню курса необходимые разделы и пропустив уже изученные.

В качестве средств разработки обучающих программ используются различные программные средства:

1) Asymetrix ToolBook ИТМ Instructor – программный продукт для профессиональных разработчиков, предназначенный для создания интерактивных обучающих курсов и других мультимедийных приложений для Интернета;

2) дизайнер курсов «УНИАР_Продюсер 2002» – уникальный инструментальный пакет для разработки высококачественных компьютерных обучающих курсов, контролирующих и демонстрационных учебных программ, материалов для самостоятельной работы учащихся, позволяющий создавать мультимедийные обучающие курсы пользователям, которые не являются профессиональными программистами;

ДЬЯЧКОВ Валерий Павлович – кандидат педагогических наук, доцент по кафедре информатики и вычислительной техники Вятского социально-экономического института

© Дьячков В. П., 2008

3) инструментальный пакет eLearning Office 3000 – предназначен для преподавателей высших и средних учебных заведений; его отличают дружелюбный интерфейс и максимальная автоматизация работ;

4) технология «TeachPro™», позволяющая создавать обучающие программы по различным дисциплинам с возможностью хранения большого объема учебного материала, использованием концепции деятельного участия обучающегося, организации контроля над процессом обучения и возможности организации дистанционного обучения;

5) программная оболочка TestBOX представляет собой среду для автоматизации учебного процесса, включающую пять основных программ, с помощью которых можно набрать тексты обучающих курсов любой степени сложности, представляемых в виде Web-страниц (CourseBuilder), создать тесты (TestBuilder), проанализировать ответы на тест как отдельного пользователя, так и группы в целом, вывести результаты тестирования в виде таблицы или отчета (Administrator), компоновать электронные курсы и тесты в зависимости от желания его создателя (Collection Builder).

Каждое из перечисленных средств разработки имеет свои достоинства и недостатки. Однако все они представляют достаточно сложные программные комплексы, которые требуют от разработчиков обучающихся программ специальных знаний, умений и навыков, в том числе и умения программировать. Но не каждый преподаватель даже высшей школы способен создать обучающую программу с элементами программирования на языках высокого уровня. Поэтому ниже приведены несколько примеров разработки программ с помощью простейших программных сред: Power Point, HTML, Visual Basic 6.0 и др.

Прикладная программа для создания презентаций Power Point, входящая в состав Microsoft Office XP, обладает достаточно широким набором средств для работы с текстом, графикой (GIF, JPG, BMP) и позволяет вставлять в обучающие программы звук, анимацию объектов, гипертекстовые ссылки и другие эффекты. Сохранить электронное учебное пособие, созданное в Power Point, можно в виде исполняемого exe-файла, запускаемого в дальнейшем на любом компьютере под управлением практически любой операционной системы Windows. В качестве примера приведем разработку обучающей программы по изучению основных компонентов персонального компьютера «Аппаратные средства персонального компьютера». Она состоит из информационного блока, в котором дается теоретический материал, переработанный таким образом, чтобы на

слайдах было минимальное количество слов и максимальный объем информации. На двадцати шести слайдах программы Power Point размещена информация об основных компонентах компьютера, включающая как краткую формулировку определений о каждом из них, так и основные технические характеристики. Например, для мониторов дается определение понятия «монитор», приводится их классификация по основным отличительным признакам, а также их технические данные: размер по диагонали, разрешающая способность, частота регенерации кадров и др.

Разработку обучающих программ на языке HTML покажем на примере создания электронных курсов по изучению прикладных программ по MS Word.

Разработка обучающего курса по MS Word включает создание трех основных блоков: 1) информационного; 2) блока упражнений и 3) контролирующего блока.

Информационный блок содержит в себе необходимую для изучения информацию. Это текстовая информация, которая дублирует некоторую часть живых лекций. Она хорошо структурирована и представляет собой законченные фрагменты курса.

Блок упражнений представляет совокупность тренировочных упражнений, выполняемых в интерактивном режиме. То есть пользователь не просто выполняет представленные ему упражнения, а втягивается в систему обучения благодаря различным анимационным рисункам и специально подобранным подсказкам.

Контролирующий блок состоит из тестовых заданий по данному курсу и пройденным в нём темам и упражнениям.

Информационный блок представлен в виде HTML-страниц, связанных друг с другом. В них представлена информация, которая дублирует часть живых лекций, обычно преподаваемых при основе работы в MS Word.

Принципы изложения учебного материала в условиях компьютерного обучения приобретают все большее значение по мере того как возрастают возможности компьютера в предъявлении и интерпретации разных типов разнообразной информации и углубляется понимание наиболее рационального использования мультимедийного предъявления информации. Современный компьютер обладает большими возможностями в применении разнообразных типов информации. Это и текст, и чертежи, и графика, и анимация, и видеоизображения, и звук, и музыкальное сопровождение. Эффективное использование различных типов предъявления информации с учетом психологических особенностей ее переработки позволяет значительно повысить эффективность учебного процесса.

Нередки примеры, когда разработчики обучающих программ механически переносят способ расположения текста на экран монитора, пренебрегают закономерностями психологии восприятия текста и рисунка, задавая темп изменения изображения, не учитывая, что разные учащиеся имеют неодинаковую смысловую скорость и требуют для переработки информации различные временные интервалы.

В связи с этим следует предоставить учащимся возможность самим выбирать темп смены изображения, при этом учащийся должен иметь возможность в любое время повторно вывести на экран любую необходимую ему информацию.

При построении интерфейса обучающей системы необходимо учитывать достижения теории дизайна. Это прежде всего касается таких основных принципов теории живописи, как пропорция, порядок, акцент, единство и равновесие.

Принцип пропорции касается соотношения между размерами объектов и их размещением в пространстве. Организуя данные на экране дисплея, необходимо стремиться к тому, чтобы логически связанные данные были явно сгруппированы и отделены от других категорий данных. Функциональные зоны на дисплее должны разделяться с помощью пробелов и других средств: разные типы строк, ширина, уровень яркости, геометрическая форма, цвет. Для сокращения времени поиска табличные данные должны разделяться на блоки. Необходимо учитывать, что плоскость теплых цветов обычно кажется больше, чем холодных. Разбиение на блоки, использование пробелов, табуляции, ограничителей, а также варьирование яркости цвета групп данных – важнейшие средства упорядочения графической информации.

При размещении данных необходимо помнить о правиле «золотого сечения», в соответствии с которым объекты, которые привлекают внимание, лучше размещать в разных частях изображения, а не группировать в центре.

Порядок означает такую организацию объектов на экране дисплея, которая учитывает движение глаза. Установлено, что глаз, привычный к чтению, начинает движение обычно от левого верхнего угла и движется взад-вперед по экрану к правому нижнему. Поэтому начальная точка восприятия должна находиться в левом верхнем углу экрана, а списки для быстрого просмотра должны быть подогнаны к левому полю и выровнены вертикально.

Для облегчения восприятия разные классы информации должны специально кодироваться. Так, связанные, но разнесенные по экрану данные должны кодироваться одним цветом. Цвет можно использовать и для выделения заголов-

ков, новых данных или данных, на которые следует немедленно обратить внимание. В целом организация данных на экране должна облегчать нахождение подобий, различий, тенденций и соотношений.

Необходимо выделять критическую информацию, необычные данные, элементы, требующие изменения, сообщения высокого приоритета, ошибки ввода, предупреждения о последствиях команды и т. п. Для того чтобы привлечь внимание учащихся к основному объекту, целесообразно использовать цветовое пятно: самым ярким цветом изображается основной объект, остальные его части – дополнительным. Если цветовая гамма строится без учета психологии восприятия рисунка, это затрудняет выделение главного, приводит к утомлению зрения.

Нужно учитывать, что светлые цвета на темном фоне кажутся приближенными к зрителю, а темные на светлом – удаленными. В тех случаях, когда речь идет об эвристических рекомендациях, цвет можно согласовывать с обычным изображением: красный – запрет, зеленый – рекомендация, желтый – предосторожность.

Принцип единства требует, чтобы элементы изображения выглядели взаимосвязанными, правильно соотносились по размеру, форме, цвету. С этой целью необходимо позаботиться об упорядочении организации данных. Они могут быть организованы последовательно, функционально, по значимости. При этом учащегося следует ознакомить с принципом расположения данных.

Следует позаботиться о том, чтобы идентичные данные были представлены унифицированно, а разноплановые – по-разному.

Для передачи разграничения нужно использовать контрастные цвета, а для передачи подобия – похожие, но различные. Представление информации должно быть унифицированным и логичным.

Для достижения единства изображения в целом используются рамки, оси, поля. Впечатление единства группы создает свободное пространство вокруг них. Считается, что уравновешенное изображение создает у пользователя ощущение стабильности и надежности, а неуравновешенное вызывает стресс.

Для правильного распределения визуальной тяжести на экране дисплея необходимо помнить, что любой хроматический цвет зрительно тяжелее, чем ахроматические – белый и черный; большие предметы зрительно тяжелее маленьких; черное тяжелее белого, неправильные формы тяжелее правильных.

Принцип равновесия (баланса) требует равномерного распределения оптической тяжести изображений. Поскольку одни объекты зритель-

но воспринимаются как более тяжелые, а другие – как более легкие, необходимо распределять эту оптическую тяжесть равномерно по обеим сторонам изображения.

Информация не должна скапливаться на одной стороне экрана, логические группы информации должны продуманно размещаться в пространстве, заголовки – хорошо центрироваться.

Блок упражнений предусматривает реализацию следующих целей:

- выработку у студентов устойчивых навыков решения подобных заданий;
- закрепление на практике полученных теоретических знаний;
- оценку качества усвоения студентами нового материала;
- повторение и восстановление в памяти ранее изученного материала;
- выработку у студентов навыков компьютерного общения и самостоятельного решения задач в условиях ограниченного времени.

При подборе практических заданий учитывались следующие требования:

- всестороннее отражение в заданиях нового теоретического материала;
- сходность предлагаемых заданий с теми, что рассматривались ранее в виде решенных примеров;
- отсутствие примеров повышенной трудности или требующих нестандартного подхода;
- простота получаемых ответов и удобство их ввода и редактирования.

Преподавателю нужно всего лишь открыть папку с упражнениями, сделанными учеником, проверить их и оперативно оценить успешность изложения темы и, если необходимо, принять меры по корректировке учебного процесса.

Контролирующий блок предназначен для закрепления нового материала и контроля его усвоения. Форма ввода ответа на вопросы предполагает использование как классической системы (один правильный из четырех ответов), так и возможность ввода конструктивного ответа, когда студент конструирует свой ответ из предложенных фрагментов. Система вопросов подбиралась с учетом следующих требований:

- широкий охват нового теоретического материала;
- разноплановость в смысле возможных вариантов ответов;
- отсутствие вопросов, предполагающих ответы типа «да» – «нет», и ответов, требующих пояснения.

Блок ответов на контрольные вопросы устроен таким образом, что, дав ответ на первый вопрос, студенты могут перейти к последнему, затем вернуться назад и исправить первый ответ. Ответ, данный на вопрос, не исчезает, он остается

доступным для редактирования. По окончании сеанса работы с тестовыми заданиями студент получает определенный балл, который соответствует стандартной пятибалльной системе. Преподавателю остаётся лишь проверить количество набранных баллов для получения точных данных о знаниях ученика. Так как предложенные тесты охватывают весь учебный материал, можно с уверенностью говорить об уровне подготовки студента по MS Word.

Обучающий курс по MS Excel представляет автоматизированный обучающий курс, состоящий из трёх блоков: информационного, блока упражнений и контролирующего. Все блоки реализованы с помощью HTML. Оболочка для загрузки HTML-страниц написана на Visual Basic 6.0.

Для реализации оболочки электронного учебника был выбран язык программирования Visual Basic 6.0, так как он, в отличие от многих других языков, позволяет быстро и легко создавать интерфейс приложения. Также он дает возможность подгружать HTML-страницы и создавать кнопки управления, которые имеются в любом браузере.

Оболочка электронного учебника выполняет следующие функции: 1) регистрацию учётных записей пользователей; 2) загрузку учётной записи конкретного пользователя; 3) сохранение последней просмотренной страницы; 4) вызов помощи; 5) загрузка HTML-страниц; 6) управление HTML-страницами.

Также имеется кнопка вызова справки о программе и кнопка выхода из неё. Пользовательское меню программы позволяет реализовать три режима работы: 1) в режиме регистрации создается новая учётная запись, при этом в каталоге «users» создаются файл с именем пользователя и расширением «.user» и каталог в папке «xls», названный также именем пользователя; 2) в режиме загрузки открывается учётная запись пользователя на той странице учебника, на которой пользователь остановился на последнем сеансе работы с программой; 3) в режиме сохранения записывается путь к последней просмотренной странице учебника.

Разработка информационного блока реализована в виде HTML-страниц, так как система, построенная на гипертексте, более удобна и эффективна для работы пользователя в электронном учебнике.

В информационном блоке изучаются основы приемы работы с MS Excel. Главной задачей является предоставление обучаемому необходимых знаний и подготовка его к выполнению практических упражнений.

Учебный материал разбит на 4 урока, каждый из которых состоит из нескольких глав.

Урок 1. Электронная таблица. Включает восемь глав: 1) Основные понятия; 2) Ввод данных; 3) Оформление ячеек; 4) Сไตล์; 5) Формат чисел; 6) Добавление и удаление строк и столбцов; 7) Размеры ячеек и их замораживание; 8) Контрольное упражнение. На этом занятии обучаемый знакомится с основными понятиями и структурой электронных таблиц Excel, узнает, как добавлять, копировать и переименовывать листы Excel, строить таблицы, оформлять и форматировать их. Знакомится с разными форматами чисел и учится настраивать собственные стили.

Урок 2. Формулы и функции. Состоит из семи глав: 1) Ввод формул и ссылки на ячейки; 2) Функции; 3) Диапазон ячеек; 4) Копирование формул; 5) Поиск ошибок; 6) Применение функций; 7) Контрольное упражнение. Здесь студент учится вводить формулы, пользоваться функциями, копировать формулы, находить и исправлять ошибки. Узнаёт о механизме относительной адресации ячеек и изучает некоторые функции, работающие с текстовыми строками и величинами типа даты и времени.

Урок 3. Графики и диаграммы. Позволяет сформировать у студента умения и навыки построения диаграмм и графиков по данным, размещенным в таблицах: 1) Добавление диаграммы; 2) Выбор типа диаграммы; 3) Подписи рядов данных; 4) Форматирование текста; 5) Объёмная диаграмма; 6) Ряды данных; 7) Контрольное упражнение. В данном уроке студенты изучают ввод данных в таблицы Excel и выполняют необходимые расчёты, отображают результаты вычислений в виде диаграмм. На занятии рассмотрены вопросы применения мастера диаграмм, способов задания источников данных, приемы форматирования текста и рядов данных. Таким образом, обучаемый знакомится с методикой смены типа диаграммы и приёмами настройки объёмных диаграмм.

Урок 4. Подготовка листов Excel к печати. Знакомит студентов с вопросами: 1) ориентации и изменения масштаба страницы; 2) настройки полей; 3) добавления колонтитулов; 4) сортировки данных; 5) фильтрации данных; 6) разбиения текста на страницы; 7) обучения способам скрытия строк и столбцов; 8) задания диапазонов ячеек; 9) настройки диаграмм; 10) выполнения контрольного упражнения. В этом уроке рассмотрены вопросы выбора диапазона печатаемых ячеек, настройки параметров страницы и формирования схемы разбиения листа Excel на страницы. Студент учится фильтровать и сортировать строки, скрывать ячейки, дублировать заголовки таблиц, подгонять размеры выводимой области под

формат страницы, печатать одновременно несколько листов в нескольких экземплярах.

Разработка блока упражнений неразрывно связана с первым блоком информации, поэтому он встроен в информационный блок, что позволяет студенту сразу перейти от теории к выполнению тренировочных упражнений, закрепив свои знания практикой. Выполнение упражнений происходит в интерактивном режиме. Упражнения находятся в каждой главе учебного материала. По прохождению всего урока даётся контрольное упражнение, которое обобщает и закрепляет полученные навыки и умения. Переход в интерактивный режим осуществляется по нажатию ссылки «Открыть Excel», находящейся в начале каждого упражнения. В случае затруднения студент может открыть программу MS Excel как обычное офисное приложение и выполнить упражнения, сохранив их на свой носитель (гибкий диск или флэш-память).

Разработка контролирующего блока позволяет проверить уровень усвоения полученных знаний, умений и навыков. Контролирующий блок реализован в виде тестов, написанных с помощью HTML и заканчивающих изучение урока. Каждый тест состоит из десяти вопросов, связанных одной темой. После того как студент ответит на все вопросы, система сама просчитывает количество ошибок и выдает оценку: если менее 5 правильных ответов – выставляется оценка «неудовлетворительно», при наличии 4–5 ошибок – оценка «удовлетворительно», две или три ошибки позволяют студенту получить оценку «хорошо», и только при наличии одной ошибки выставляется оценка «отлично». Данная система тестов удобна тем, что преподаватель не тратит время на проверку знаний учащихся, программа сама проверяет и оценивает знания студентов.

Руководство пользователя позволяет ознакомиться с порядком работы в обучающей программе. Оно представляет перечень инструкций для начинающего пользователя и прорабатывается им при первом обращении к обучающей программе.

Ограниченность размера статьи не позволяет привести примеры разработки листингов программ, реализованных в данных обучающих программах, однако можно сделать вывод, что приведенные программы достаточно просты и не требуют каких-то специальных знаний в области программирования. Подходы и программы могут быть использованы и в других обучающих программах, где требуется разработка трёх основных компонентов: информационного, тренировочного и контролирующего блоков.

И. С. Жаховская

ИЗ ОПЫТА ИСПОЛЬЗОВАНИЯ МЕТОДА ПРОЕКТОВ НА УРОКАХ ИНФОРМАТИКИ

В статье рассмотрена авторская методика использования метода проектов в преподавании информатики.

Проектирование – любимый вид учебной деятельности наших учащихся. При предметном проектировании они выбирают проблему, предмет, явление или событие, а затем изучают их, решают технические задачи, связанные с выбранной темой, анализируют полученные результаты и делают выводы. Эти проекты выполняются по индивидуальным заданиям и носят характер учебной задачи. Здесь легче всего проявить все свои положительные черты, интересы и возможности. А на защите проектов проявляются все таланты учащихся, их авторитет среди одноклассников растёт.

Поэтому для преподавания блока, выделенного нами в курсе информатики «Программное обеспечение ПК», выбран *метод проектов* как способ организации процесса познания учащимися учебного материала, суть которого в современном понимании характеризуется тезисом: «Все, что я познаю, я знаю, для чего это мне надо и где и как я могу эти знания применить!».

Все начинается с постановки проблемы (практически или теоретически значимой), а в итоге получаем результат, который можно применить в реальной практической деятельности. Метод проектов как нельзя лучше формирует компетентности в практической деятельности у учащихся.

В старшей школе работа по проектированию значительно усложняется. Если в начальной и основной школе дети, в основном, обучаются проектированию как методу работы, то в старшей школе они обязаны провести исследование, проанализировать результаты, выполнить их оформление и представить конечный продукт – готовый проект. Старшеклассники самостоятельно выбирают тему проекта, над которой будут работать (выбор по принципу: пригодится в будущем; интересный вопрос одного из школьных предметов; непонятное явление; хочу рассказать о том, что знаю; приведу в порядок знания по данному вопросу; тема для экзаменационного реферата по любому предмету; это полезно знать, и др.). Диапазон выбираемых тем очень широк. Усложняется значимость тем, их практическая направленность, социальный характер, воспита-

тельная роль. В связи с этим учителю приходится разбираться в любой области науки или искусства (невольно повышается разностороннее образование, пополняются знания, растёт учительский авторитет). Акцент делается на индивидуальную работу с учениками. Чаще она проходит не только на уроке, но и в послеурочное время, когда ребята приходят за консультацией и помощью. Усложняется процесс обработки полученных данных в ходе работы над проектом. Для разнообразия конечных продуктов, которые будут созданы по результатам работы, увеличивается перечень программ, обязательных для обработки результатов (выбор программ – в зависимости от изучаемого ПО). Усложняется аналитическая деятельность учащихся. Усложняются критерии оценки каждого этапа и требования к защите проекта в целом.

Работа выполняется индивидуально или в парах (по выбору учеников). С порядком работы ученики знакомятся в самом начале во время урока-консультации для того, чтобы знать, каких результатов от них ожидают, и планировать свои действия.

Подготовительный этап

Для вводной консультации отбираются качественно выполненные проекты учеников прошлых лет. Демонстрируется социальная, практическая и воспитательная направленность тем, разнообразие конечных продуктов обработки результатов. Учащиеся настраиваются на активную творческую работу. Они знакомятся с задачами, содержанием, организацией работы над проектом, с критериями оценки результатов и процесса проектной деятельности.

Планирование

Учащиеся получают набор бланков информационных листов проекта (бланки – в приложениях) (за основу взяты разработки программы Intel – обучение для будущего). Каждый ученик продумывает тему проекта, индивидуально обсуждает тему с учителем. В этой беседе он формулирует цели работы над проектом, выдвигает гипотезу, которую хочет доказать в результате, называет возрастную категорию людей, для которых разрабатывается проект, определяет основополагающий и проблемные вопросы, предполагает, какими могут быть источники необходимой информации, из которых они смогут брать дополнительную информацию. Бланки информационных листов ученики заполняют самостоятельно в ходе планирования промежуточных результатов работы над проектом.

Исследование темы проекта и анализ полученной информации

Схема работы каждого ученика следующая:

1. Выбрать тему, над которой будешь работать.

2. Заполнить лист планирования проекта (лист № 1).

3. Продумать цель, составить и провести социологический опрос или анкетирование (по выбору). Данные занести в лист планирования дополнительных результатов (лист № 2).

4. Выполнить анализ результатов соцопроса (анкетирования). Ознакомиться с порядком оформления результатов соцопроса с помощью MS Excel (лист № 3). Выполнить работу в MS Excel.

5. Напечатать реферат в текстовом редакторе MS Word (не более 10–15 страниц). Учитывать требования к содержанию и оформлению реферата (лист № 4).

6. В графическом редакторе создать иллюстрации к проекту (не менее двух). Использовать Paint и Adobe Photoshop.

7. Подобрать или самостоятельно выполнить дополнительное оформление: тематические картинки, фоновое оформление, портреты ученых, свое фото, музыкальное оформление, видеосюжеты. Использовать программы для обработки фото, музыки, видео.

8. Спланировать (лист № 5) и создать познавательную презентацию для представления своей темы с помощью MS Power Point.

9. Создать познавательный буклет по выбранной теме с помощью MS Publisher. Разработать и

создать свою визитку и календарь с тематикой проекта.

10. Разработать (лист № 2) викторину или тест по теме проекта. Оформить в MS Excel, используя раскрывающиеся списки и организовав подсчет правильных ответов.

11. Создать сайт по теме проекта с помощью MS Front Page.

12. Спланировать (по примеру листа № 5) и создать отчетную презентацию для представления своей работы над темой проекта MS Power Point.

13. Во время работы над проектом фиксировать источники текстовой, графической, аудио- и видеоинформации (лист № 7).

Защита проекта. Оценка результатов и процесса деятельности.

Подготовить выступление, представить проект перед одноклассниками.

После защиты каждого проекта одноклассники проводят анализ выполненной работы, задают вопросы, отмечают положительные и отрицательные стороны данной работы, высказывают свои предложения по улучшению итоговой работы. Используем принцип: «черный» и «белый» рецензенты.

Итоговая работа оформляется на бумажных и электронных носителях.

ПРИЛОЖЕНИЯ

Лист № 1. Планирование проекта

Фамилия, Имя, класс автора (-ов) проекта _____

Предполагаемая тема проекта _____

Для какой возрастной категории людей разрабатывается проект _____

Основной вопрос проекта _____

Гипотеза решения выдвинутой проблемы проекта (*какую идею хотим проектом доказать*) _____

Подвопросы темы проекта _____

Предметные области, в рамках которых разрабатывается проект _____

Познавательные цели проекта (*что можно узнать и чему можно научиться в процессе создания проекта*) _____

Творческое название проекта _____

Ключевые слова и понятия _____

Лист № 2. Планирование дополнительных материалов

Название работы	Цель работы	Содержание работы
Анкета или социологический опрос		
Викторина или тест		

Лист № 3. Порядок оформления результатов социологического опроса с помощью MS Excel

1. Откройте MS Excel.
2. Переименуйте листы книги (двойной щелчок мышью по названию листа):
1-й – на ТИТУЛ; 2-й – на ТАБЛИЦА;
3-й – на ДИАГРАММЫ; 4-й – на АНАЛИЗ РЕЗУЛЬТАТОВ.
3. На листе ТИТУЛ запишите: название проекта, авторов проекта, тему соцопроса, цель соцопроса, содержание соцопроса; среди какой части населения проводился соцопрос (или укажите возрастную категорию); даты проведения соц.опроса; выводы, сделанные после анализа результатов опроса.
4. На листе ТАБЛИЦА расположите таблицу, содержащую информацию по проведенному соцопросу. Продумайте вид таблицы, чтобы она как можно нагляднее отображала результаты опроса. Выполните форматирование таблицы.
5. На листе ДИАГРАММЫ постройте две диаграммы:
 - a. Сравнительная гистограмма данных ответов на вопросы.
 - b. Выберите вопрос, который наиболее явно отвечает цели соцопроса, и составьте долевую диаграмму ответов на этот вопрос в %.
6. На листе АНАЛИЗ РЕЗУЛЬТАТОВ оформите ваши выводы по результатам проведенного опроса. Укажите наиболее удачные моменты работы, сделайте предположения по улучшению процедуры проведения соцопроса на будущее.
7. Сохраните документ под именем РЕЗУЛЬТАТЫ СОЦОПРОСА в вашей папке.

Лист № 4. Как работать над рефератом

РЕФЕРАТ – творческая работа (статья), в которой более или менее подробно излагаются сведения по выбранной теме.

1. В общих чертах наметьте содержание работы, набросайте предварительный план.
2. Составьте список литературы, которую следует прочитать. Вам помогут в этом систематические каталоги в библиотеках, библиографические указатели, обзоры литературы в книгах или журналах.
3. Изучая литературу, делайте выписки, составляйте тезисы, конспектируйте необходимые источники. Все записи ведите на одной стороне листа или на небольших карточках. Не забывайте указывать номера страниц и библиографические данные книги: автор, название, издательство, место и год издания.
4. В результате работы с литературой у вас должны появиться собственные мысли и выводы, которые также необходимо записать.
5. Проанализируйте и систематизируйте наработанный материал. Выделите наиболее важные мысли, свяжите их с примерами и собственными выводами. Возможно, некоторые рабочие листы вам придется разрезать и по-новому объединить с другой информацией. При этом у вас выделится несколько смысловых частей, их необходимо расположить в логической последовательности.
6. Составьте оглавление реферата. В общем виде структура любого реферата следующая:
введение – обоснование актуальности (важности) выбранной темы;
основная часть – общая характеристика, описание и анализ предмета (проблемы, явления), предварительные выводы. В основной части обычно выделяют разделы или главы (не менее 2–3), в каждой главе – параграфы. Главам (разделам) и параграфам даются названия;
заключение – окончательные выводы;
библиография (список использованной литературы);
приложения (графики, схемы, таблицы).
7. Распределите собранный материал по главам и параграфам.
8. Приступайте к написанию реферата.

Реферат начинается с *введения*. Само название этого раздела говорит о необходимости ввести в курс того, о чём пойдёт речь в реферате. Постарайтесь обосновать выбор данной темы, т. е. объясните, чем она интересна, раскройте ее значимость, актуальность (важность), перечислите задачи, которые вы ставите перед собой, выделите основные вопросы, которые рассматриваются в реферате. Введение может быть небольшим и состоять, например, всего из одного абзаца, а может включать несколько абзацев или даже целую страницу, а то и больше. Но в любом случае во введении должны быть изложены самые общие сведения о предмете или проблеме реферата и показана значимость этого предмета (проблемы).

В *основной части* реферата последовательно раскрывайте все предусмотренные планом вопросы, обосновывайте, разъясняйте основные положения, подкрепляйте их конкретными примерами и фактами. Избегайте «уходов в сторону» от темы. Недопустимо искажение мыслей авторов научных

трудов. Не забудьте о собственной точке зрения на вопросы реферата. При этом постарайтесь избежать путаницы, неразберихи: нельзя «перепрыгивать» с одного на другое, писать о третьем и вновь возвращаться к первому. Ваше изложение должно быть логичным, последовательным. При этом очень важно сохранить стилистическое единство, то есть сведения из разных источников и ваши собственные мысли должны составлять единое целое.

В конце каждой главы желательно сформулировать краткие выводы, основные идеи.

При изложении материала правильно делайте ссылки на литературные источники:

- после изложения точки зрения какого-либо автора в конце предложения указывается в скобках фамилия автора и через запятую год публикации (Иванов, 1975);
- после цитаты или какого-либо факта в круглых или квадратных скобках указывается номер источника по списку литературы и номер страницы (4, с. 208).

Если вы не просто высказываете какую-то мысль автора, а точно воспроизводите её формулировку, не забудьте цитату заключить в кавычки.

Заключительная часть реферата может называться «Заключение» или «Выводы», при этом обычно заключение пишется сплошным текстом, а выводы формулируются в виде отдельных тезисов, обозначенных цифрами. В любом случае вы должны подвести общие итоги, сформулировать основные выводы, проанализировать, насколько успешно удалось решить поставленные во введении задачи, определить перспективы разработки данной темы.

Вся использованная для написания реферата *литература* должна быть перечислена в разделе, который так и называется «Литература», или «Библиография», или «Использованная литература».

– Обычно печатные работы располагаются либо в алфавитном порядке (по фамилиям авторов), либо в соответствии со сроком выхода из печати.

– В списке указываются фамилия и инициалы автора, полное название книги, место издания (если это Москва, то пишется М., если Санкт-Петербург, то СПб.; названия других городов пишутся полностью), далее после двоеточия указывается издательство, через запятую год издания цифрами без указания слова «год», например: *Гумилев А. Н. От Руси до России. – М.: Дрофа, 1996.*

– Если при написании реферата вы использовали всю книгу целиком, указывается общее количество страниц в книге (375 с.), а если была использована отдельная глава или часть книги, то указываются страницы, на которых изложена использованная информация (С. 75–81).

– Если при работе над рефератом были использованы газетные или журнальные статьи, то ссылка на них даётся следующим образом: после фамилии, инициалов автора и названия статьи ставится точка и двойной слэш //. Затем пишется название журнала, в котором эта статья напечатана, год издания, указывается его номер, номера страниц, на которых напечатана статья, например: *Корецкая Т. И. Могучие сладёны // Свирель. 2000. № 8. С. 10–11.*

В тексте даются ссылки на приложения, которые номеруются и помещаются после списка литературы. Это графики, схемы, таблицы, рисунки, фотографии.

Оформление реферата

Объём работы обычно не менее 15 печатных страниц. Размер шрифта 14. Межстрочный интервал 1,5. Отступ первой строки абзаца 12,5 мм. Текст выравнивается «по ширине». В верхнем колонтитуле, по центру, проставляются номера страниц. Текст реферата пишется или печатается на одной стороне листа формата А4. Размеры полей: левое – не менее 30 мм, правое – не менее 10 мм, верхнее и нижнее – по 25 мм.

Титульный лист оформляется следующим образом:

- в верхней части листа, отступив 25 мм от верхнего края, пишется полное название школы;
- в центральной части страницы прописными буквами пишем РЕФЕРАТ ПО ... (НАЗВАНИЕ ПРЕДМЕТА);
- строчкой ниже – слова «на тему», после которых ставим двоеточие;
- ещё ниже приводим полное название темы реферата ПРОПИСНЫМИ БУКВАМИ без кавычек и без точки в конце;
- в правой половине нижней трети листа пишем: Работу выполнил ученик... класса и указываем фамилию и имя автора работы;
- ещё ниже пишем: Учитель-консультант (или Руководитель) и указываем фамилию и инициалы учителя;
- в нижней части страницы указываются место и год написания реферата: г. Киров, 2004 год.

Титульный лист включается в общую нумерацию страниц, но номер страницы на нём не проставляют.

За титульным листом следует план реферата (или оглавление).

Заголовки разделов и глав пишут прописными буквами без точки в конце, не подчеркивая.

Заголовки параграфов начинают с абзацного отступа и с прописной буквы, а затем пишут строчными, не подчёркивая, без точки в конце. Переносы слов в заголовках не допускаются. Если вы пишете оглавление, а не план, справа не забудьте указать номера страниц.

В тексте работы заголовки разделов, глав, параграфов пишутся прописными буквами. Основной текст отделяется от заголовков дополнительным интервалом. Каждая глава начинается с новой страницы.

Лист № 5. Сценарий познавательной презентации

Тема проекта _____ Автор проекта _____

В каждой ячейке данной таблицы нужно вписать планируемое название слайда, его содержание и оформление. Действия гиперссылок обозначайте стрелками.

Слайд № 1. Титульный. Название: Содержание:	Слайд № 2 Название: Содержание:	Слайд № 3 Название: Содержание:
Слайд № 4 Название: Содержание:	Слайд № 5 Название: Содержание:	Слайд № 6 Название: Содержание:

Лист № 6. Список использованных материалов при создании проекта

Текстовые материалы (книги, энциклопедии, статьи из журналов, газет)

	Название документа:	Автор	Год издания

Графические изображения

	Название изображения:	Источник

Видеофрагменты

	Название фрагмента:	Источник

Звуковые ресурсы

	Название фонограммы:	Источник

Электронные энциклопедии

Название энциклопедии:	
Название фирмы-производителя:	
Год выпуска:	
URL страницы сайта энциклопедии:	

Консультации преподавателей

	Дата	Тема консультации	Ф.и.о. преподавателя

Лист № 7. Что должно быть сделано к защите проекта

Отметка о выполнении

I. Оформлена папка с бумажными документами

- 1. Титульный лист проекта: творческое название проекта, автор проекта, год.
- 2. Заполненный лист планирования проекта (лист № 1).
- 3. Заполненный лист планирования дополнительных материалов: анкета или социологический опрос, викторина или тест (лист № 2).
- 4. Черновик проведенного соцопроса или анкеты: содержание опроса, таблица результатов, выводы.
- 5. Черновик реферата.
- 6. Зарисовки иллюстраций проекта.
- 7. Заполненный лист сценария познавательной презентации (лист № 5).
- 8. Заполненный лист сценария презентации-отчета (лист № 5а).
- 9. Заполненный лист списка использованных материалов (лист № 6).
- 10. Эскиз буклета.
- 11. Буклет.
- 12. Эскиз сайта.
- 13. Вырезки из газет, журналов по теме проекта (или ксерокопии статей).
- 14. План выступления на защите проекта, текст выступления.
- 15. Словарик основных понятий, специфических слов по теме.

II. Электронные документы на CD-диске

- 16. Реферат (MS Word).
- 17. Соцопрос или анкета: тема, цель, вопросы, ответы (MS Word).
- 18. Отчет по проведенному соцопросу (MS Excel) (лист № 3).
- 19. Викторина или тест: тема, цель, вопросы, ответы (MS Word, MS Excel).
- 20. Папка с иллюстрациями (авторские – 2 шт., подобранные из библиотек).
- 21. Познавательная презентация (MS PowerPoint).
- 22. Презентация-отчет о проделанной работе (MS PowerPoint).
- 23. Буклет (MS Publisher).
- 24. Календарь (MS Publisher).
- 25. Визитка (MS Publisher).
- 26. Сайт (MS FrontPage).

III. Все документы должны быть проверены и оценены учителем

Н. И. Исупова

МЕСТО ДИСЦИПЛИНЫ «ЧИСЛЕННЫЕ МЕТОДЫ И ПРОГРАММИРОВАНИЕ» В ПОДГОТОВКЕ СТУДЕНТОВ ХИМИЧЕСКОГО ФАКУЛЬТЕТА

Статья посвящена особенностям преподавания учебной дисциплины «Численные методы и программирование» для специальности 011000 Химия. Рассматриваются вопросы мотивации, отбора содержания курса, выбора средств реализации численных методов и некоторые другие методические аспекты.

Развитие новых информационных технологий и широкое внедрение математических методов в инженерные исследования привели к обширному использованию компьютера практически во всех областях современной науки и техники. В настоящее время в химии, как и во многих других науках, проводимые с помощью компьютерной техники вычисления составляют значительный процент от общего объема работ. Компьютеры стали одним из наиболее важных и универсальных из доступных химику средств исследования.

В настоящее время компьютеры в химии используются, например, для определения структуры кристаллов и квантово-химических расчетов, для сбора и обработки экспериментальных данных, управления экспериментом в режиме реального времени, при синтезе органических соединений [1]. Стремительный прогресс в области вычислительной техники обусловил успешное применение методов компьютерного моделирования в биологической и супрамолекулярной химии, не говоря уже о том, что современные исследования в области атомной промышленности вряд ли были бы возможны без применения компьютеров и численных методов.

Таким образом, уровень современного специалиста в области физической, органической, координационной химии предполагает, с одной стороны, знание основных расчетных схем и методов соответствующих разделов химии, с другой – знание основных численных методов решения задач и умение применять их при компьютерном моделировании химических процессов.

Все вышеизложенное позволяет сделать вывод о необходимости изучения предмета «Численные методы и программирование» студентами химического факультета (особенно в рамках специальности, связанной с так называемой классической химией).

ИСУПОВА Наталья Ивановна – кандидат педагогических наук, старший преподаватель кафедры прикладной информатики ВятГГУ
© Исупова Н. И., 2008

Как разработка любой новой дисциплины, планирование данного курса привело к необходимости решить ряд важных вопросов.

1. Мотивация изучения курса

При изучении информатики студенты, как правило, не задают вопросов, зачем им это нужно, так как всем ясно, что сегодня информационная культура и компьютерная грамотность являются залогом будущей успешной профессиональной деятельности практически в любой сфере.

А первое знакомство с численными методами часто приводит к негативному отношению к предмету, поскольку их фундаментом является математический аппарат. И у студентов может сложиться ошибочное впечатление, что изучаемые вопросы являются прерогативой математики, которую, как известно, многие учащиеся нематематических профилей недолюбливают.

Поэтому возникает необходимость в дополнительной мотивации, цель которой – убедить студентов в том, что знание численных методов решения задач действительно может быть полезно в их будущей профессиональной деятельности.

Такая мотивация проводится за счет включения в теоретический и практический разделы курса задач химического исследования, приводящих к необходимости применения тех или иных численных методов для их решения. Так, уже на первой лекции рассматривается несколько примеров задач из химии, которые наиболее ярко иллюстрируют применение математического аппарата и элементов компьютерного моделирования.

Пример 1. Этапы моделирования при решении химической задачи

1) Постановка задачи. Проводится химическая реакция омыления жиров щелочью. Определить степень полноты реакции в некоторый момент времени t [2].

2) Математическая модель. Известно, что скорость реакции при постоянной температуре пропорциональна произведению концентрации веществ, участвующих в реакции, и определяется следующим обыкновенным дифференциальным

уравнением (ОДУ): $\frac{dy}{dt} = K(P - y)(Q - y)$, где

y – степень полноты реакции;

P, Q – концентрации веществ, измеряемые в моль/л;

K – коэффициент равновесия.

3) Математический анализ модели. Проводится анализ данного ОДУ на предмет решения. Согласно теореме Коши данное уравнение имеет единственное решение.

4) Подготовка исходной информации. Состоит в определении конкретных значений P, Q и K для веществ, участвующих в реакции и подстановке их в уравнение.

5) Численное решение задачи. Предполагает составление алгоритма для ее решения на каком-либо языке программирования либо использование какой-либо информационных технологий соответствующего назначения.

6) Анализ результатов моделирования, т. е. проверка полученного решения на предмет соответствия целям поставленной задачи.

На практических занятиях мы возвращаемся к приведенным в лекциях задачам и решаем их с помощью одного или нескольких численных методов.

2. Отбор содержания курса

Классическое содержание дисциплины «Численные методы» для студентов-математиков или студентов-информатиков включает достаточно много разделов, изучение которых предполагается в течение двух-трех семестров. Среди основных объектов изучения – вопросы математического анализа (вычисление погрешностей, дифференциальное и интегральное исчисления, задачи оптимизации), задачи алгебры (решение алгебраических, трансцендентных, дифференциальных и интегральных уравнений и их систем), элементы линейного программирования, статистические методы обработки данных и т. д. Их охват слишком широк для не столь большого по объему курса, предназначенного для студентов химического факультета.

В связи с этим встает проблема отбора содержания курса. Госстандарт по данной дисциплине, как обычно, содержит общие, отвлеченные формулировки предполагаемых разделов изучения, требующие конкретизации.

При отборе содержания мы также руководились мыслью о том, чтобы осветить вопросы, максимально приближенные к специальности студентов. Поэтому приоритет при составлении тематического плана отдавался тем разделам, методы которых в первую очередь можно применить при моделировании химических процессов.

В результате содержание курса составили следующие основные темы:

- основы теории погрешностей;
- численные методы решения нелинейных уравнений;
- численные методы решения систем линейных алгебраических уравнений;
- интерполирование функций;
- численное дифференцирование;
- численное интегрирование;
- численные методы решения дифференциальных уравнений;
- численные методы обработки результатов экспериментальных данных.

Первые три из них и начало четвертой изучаются в рамках одного семестра, который заканчивается зачетом, остальные – во втором семестре со сдачей экзамена по всей дисциплине.

И опять же, для поддержания интереса к предмету изучение каждой темы начинается с разбора задач из химии, решение которых предполагает использование того или иного численного метода.

Пример 2. Решение систем алгебраических линейных уравнений

Проводится реакция получения окиси этилена. Необходимо рассчитать оптимальные стехиометрические коэффициенты [3].

Математическая модель: Обозначим через x_i стехиометрические коэффициенты, тогда, учитывая уравнение получения окиси этилена, получим уравнение:



Для расчета стехиометрических коэффициентов находят однородную, т. е. без свободных членов, систему линейных уравнений (СЛУ), учитывая число атомов в молекулах:

	C_2H_4	O_2	$\text{C}_2\text{H}_4\text{O}$	CO_2	H_2O
С	2	0	2	1	0
Н	4	0	4	0	2
О	0	2	1	2	1

$$\Rightarrow \begin{cases} 2x_1 + 0 \cdot x_2 = 2x_3 + 1 \cdot x_4 + 0 \cdot x_5 \\ 4 \cdot x_1 + 0 \cdot x_2 = 4 \cdot x_3 + 0 \cdot x_4 + 2 \cdot x_5 \\ 0 \cdot x_1 + 2 \cdot x_2 = 1 \cdot x_3 + 2 \cdot x_4 + 1 \cdot x_5 \end{cases}$$

Математический анализ модели: данную систему нельзя решить точными методами, так как система содержит 5 неизвестных и 3 уравнения, поэтому для ее решения применяется один из численных методов.

В результате решения получаются числа: $x_1 = 8$, $x_2 = 9$, $x_3 = 6$, $x_4 = 4$, $x_5 = 4$. То есть исходное уравнение реакции можно записать в следующем виде:
 $8 \text{C}_2\text{H}_4 + 9 \text{O}_2 \rightarrow 6 \text{C}_2\text{H}_4\text{O} + 4 \text{CO}_2 + 4 \text{H}_2\text{O}.$

3. Выбор средств реализации численных методов

При изучении численных методов решения задач математического моделирования возможно использование различных средств.

Во-первых, все вычисления можно проводить по детально разработанным алгоритмам с использованием калькулятора. Преимуществом такого подхода является детальная проработка алгоритма метода. Однако большая трудоемкость «ручного» счета, а также трудоемкость исправления ошибок снижают эффективность этого подхода. Погрузившись в последовательность арифметических действий, обучаемый часто упускает из вида суть – алгоритм метода.

С другой стороны, можно тщательно разработать алгоритм и запрограммировать его на одном из языков программирования. преимуще-

ством этого является, конечно, непосредственная работа с алгоритмом метода, который постоянно находится в центре внимания. Однако недостаточная культура программирования на алгоритмических языках может привести (и часто приводит!) к тому, что обучаемый большую часть времени тратит на отладку программы, что по большей части развивает его навыки программирования на алгоритмических языках, чем его умение решать задачи математического моделирования с использованием численных методов.

Кроме того, для успешного решения задач численных методов средствами программирования студенты должны обладать хорошими знаниями какого-либо языка программирования и навыками написания программ. Однако для студентов химического факультета не предусмотрено предварительное знакомство с соответствующими вопросами, так как при изучении дисциплины «Информатика» рассматриваются лишь современные информационные технологии, не затрагивающие программирование. Изучать же основы программирования в рамках данного предмета с их последующим применением к численным методам нет возможности в силу ограниченности курса по времени.

Таким образом, сформулируем общие характеристики программного средства, наиболее удобного для преподавания численных методов решения задач математического моделирования. Хорошо было бы иметь средство, которое:

во-первых, имело удобный графический интерфейс, не требовало дополнительных знаний какого-либо языка программирования;

во-вторых, обладало наглядными и интуитивно понятными средствами для представления алгоритма метода решения задачи;

в-третьих, давало возможность отображать все промежуточные вычисления в виде таблицы, так чтобы наглядно видеть идеи метода;

в-четвертых, давало возможность автоматически пересчитывать все вычисления, например, при других исходных данных или при обнаружении и исправлении ошибки в какой-либо формуле.

Очевидно, что всеми этими возможностями обладают табличные процессоры (например, Microsoft Excel), а также специализированные математические программные продукты (например, MathCAD).

При этом использование каждого из этих средств, в свою очередь, обладает своими досто-

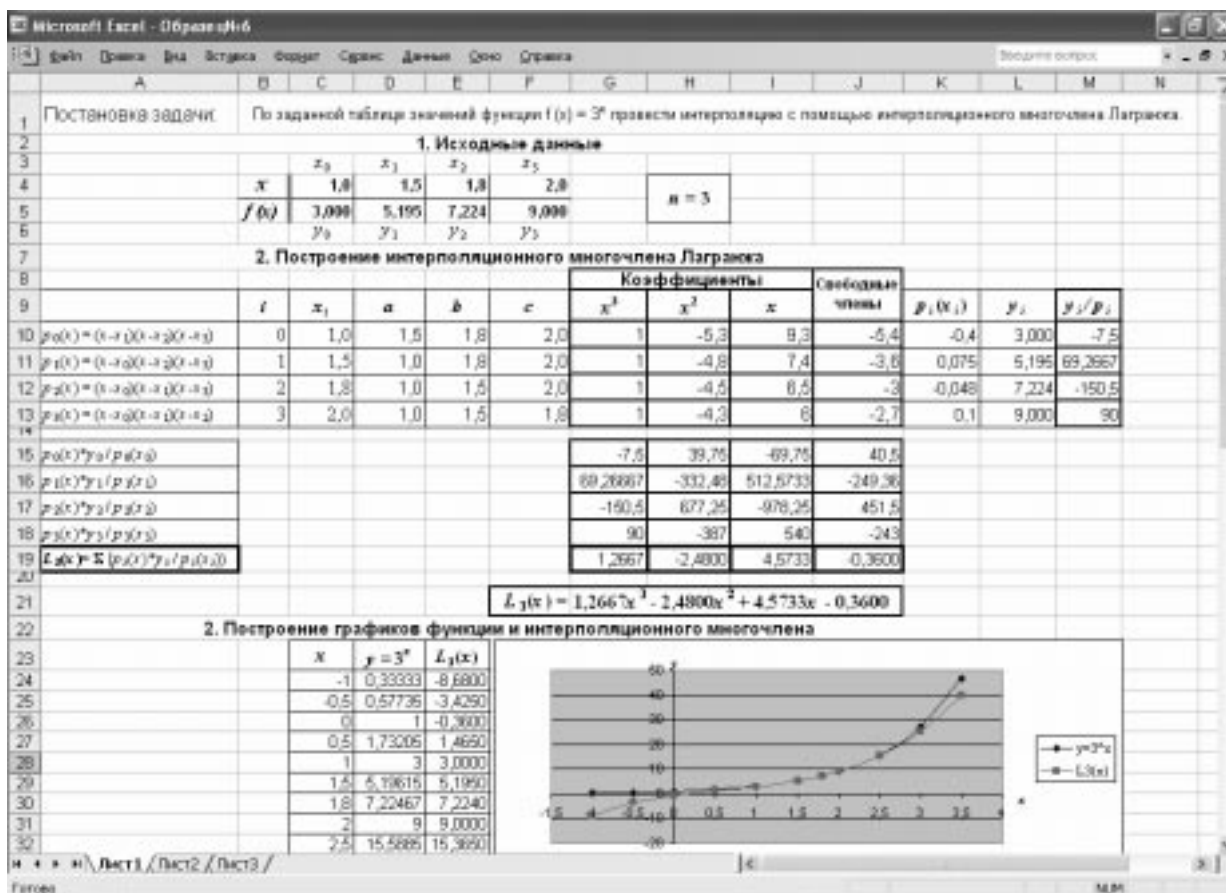


Рис. 1. Решение задачи интерполяции в Microsoft Excel

инствами и недостатками. Так, при работе в Microsoft Excel можно достаточно хорошо видеть общую схему метода, использование ссылочной адресации заставляет продуманно применять те или иные формулы. Однако некоторые численные методы решения задач невозможно реализовать средствами табличного процессора либо их реализация достаточно громоздка.

В то же время применение математических пакетов, несмотря на свои широкие возможности для решения подобных задач, неудобно для учебного процесса, а более целесообразно для научных расчетов специалистов. Действительно, они являются отличным инструментом для научно-исследовательской работы, но слишком быстро приводят к результату, к ответу, зачастую скрывая алгоритм его получения от пользователя, что не позволяет достигнуть хорошего усвоения алгоритмов численных методов решения задач математического моделирования. При таком подходе этап моделирования явно будет преобладать над этапом алгоритмизации, а этап программирования вообще будет отсутствовать, что вряд ли

можно назвать удачным в свете целей изучения данной дисциплины.

Разумным компромиссом в сложившейся ситуации, на наш взгляд, является рациональное сочетание работы в табличном процессоре Microsoft Excel и математическом пакете MathCAD.

Так, простые задачи, решение которых в MathCAD предполагает лишь применение соответствующей встроенной функции, учащимся предлагается реализовать в Excel, где подобные алгоритмы не столь очевидны и требуют некоторого осмысления. Например, средствами Microsoft Excel мы рассматриваем решение уравнений, систем уравнений, простейшие задачи интерполяции.

Задачи же более сложного характера, методы решения которых, как правило, состоят из нескольких этапов, решаются с помощью MathCAD: численное дифференцирование и интегрирование, решение дифференциальных уравнений и др.

В идеале полезно рассмотреть решение задач в обеих программах и сравнить полученные результаты. Однако это не всегда получается в силу ограниченности объема учебной дисциплины.

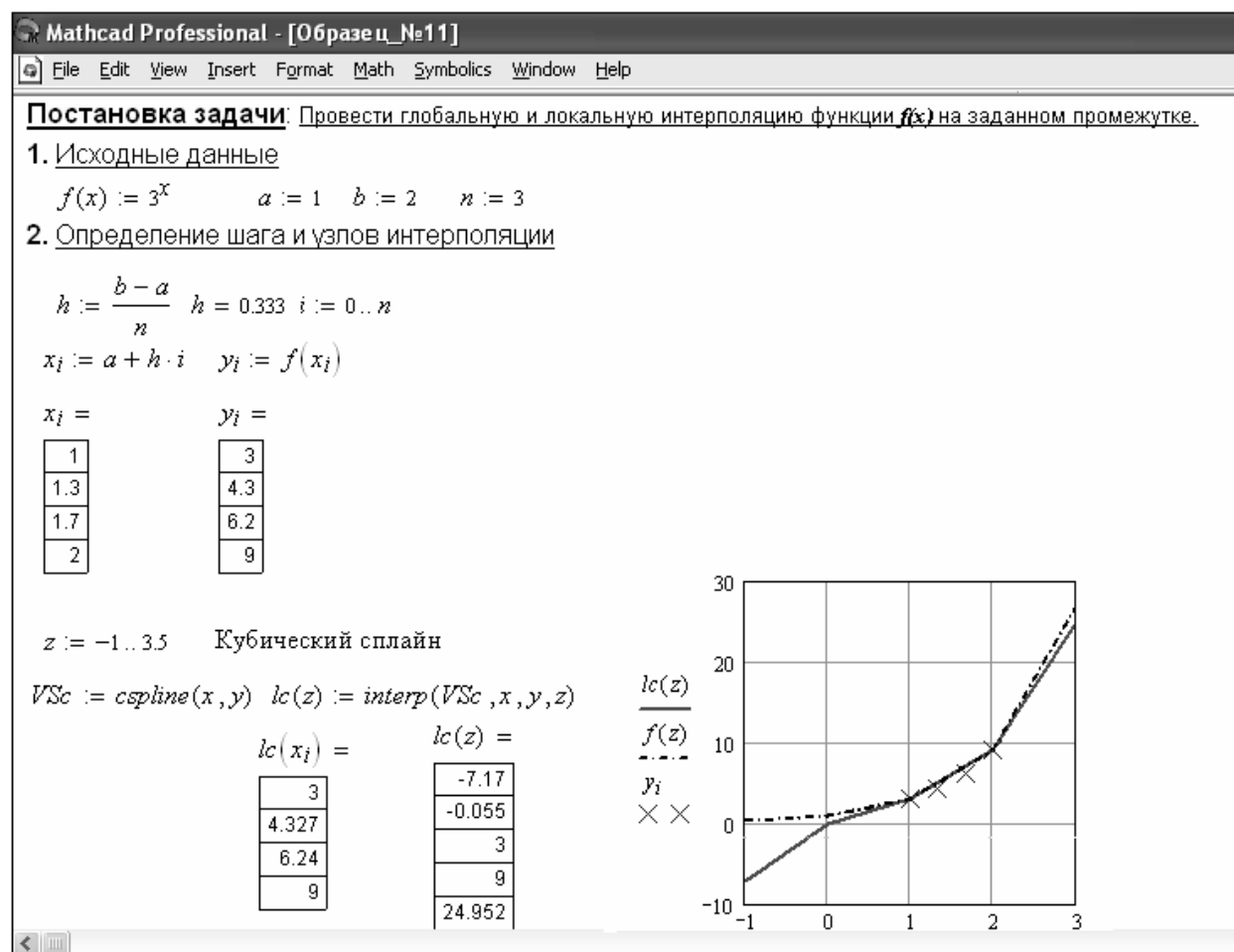


Рис. 2. Решение задачи интерполяции в MathCAD

Пример 3. Интерполирование функций

Постановка задачи. По заданной таблице значений функции $f(x) = 3^x$

x	1,0	1,5	1,8	2,0
$f(x)$	3,000	1,195	7,224	9,000

1) составить формулу интерполяционного многочлена Лагранжа;

2) доказать, что построенный многочлен действительно является интерполяционным;

3) вычислить значение интерполяционного многочлена в точках 1,3; 1,65; 1,9 и сравнить полученные данные со значениями исходной функции в этих точках;

4) построить графики функции и интерполяционного многочлена;

5) оценить погрешность произведенной интерполяции.

Реализации решения данной задачи средствами табличного процессора Microsoft Excel и математического пакета MathCAD приведены на рис. 1 и рис. 2 соответственно.

4. Методы контроля

Достаточно сложный для усвоения материал численных методов, опирающийся на математический инструментарий, требует дополнительных мер контроля знаний учащихся, в качестве которых используются:

1) Регулярная проверка домашнего задания

В конце каждого занятия студенты получают домашнее задание, которое, как правило, состоит в закреплении изученных методов путем решения своего варианта задания. Каждому студенту выдается индивидуальная карточка с задачей, подобной решенной на занятии (количество вариантов равно количеству студентов во избежание коллективного решения). В начале следующего занятия домашнее задание проверяется и выставляется оценка. За не вовремя сданное домашнее задание оценка снижается на балл за каждую отсрочку.

2) Домашние контрольные работы

По некоторым темам курса, где не предусмотрен компьютерный практикум (например, «Основы теории погрешностей»), студентам выдается домашняя контрольная работа (тоже индивидуальная для каждого студента), которая выполняется на оценку.

3) Аудиторные контрольные работы

В конце каждого изученного раздела студентам предлагается аудиторная контрольная работа по темам:

- решение уравнений и систем уравнений;
- интерполяция функций;
- численное дифференцирование и интегрирование.

Каждая контрольная работа разработана в трех вариантах и состоит из теста, включающего

20 вопросов по теоретической части раздела и 2–3 задач по данным темам.

4) Стандартные формы итогового контроля – зачет и экзамен

Организованный таким образом промежуточный контроль, во-первых, дисциплинирует студентов и стимулирует выполнение ими всех заданий (поскольку своевременная сдача всех лабораторных работ и успешное написание контрольных освобождает учащихся от сдачи зачета и выполнения практического задания на экзамене), во-вторых, помогает преподавателю выставлять промежуточную и итоговую аттестацию студентов.

Таким образом, методика преподавания такой, казалось бы, уже классической дисциплины, как «Численные методы и программирование», в настоящее время требует совершенствования. Подобный пересмотр методических аспектов преподавания курса обусловлен, в первую очередь, включением данной дисциплины в учебный план нематематических специальностей, что приводит к необходимости адаптации традиционных форм и методов изучения к особенностям соответствующих профилей.

Примечания

1. Кларк, Т. Компьютерная химия [Текст] / Т. Кларк. М.: Мир, 1990. 383 с.

2. Джонсон, Дж. К. Численные методы в химии [Текст] / Джеффри К. Джонсон. М.: Мир, 1993. 504 с.

3. Кафаров, В. В. Математическое моделирование основных процессов химических производств [Текст] / В. В. Кафаров, М. Б. Глебов. М.: Высш. шк., 1991. 399 с.

Н. И. Исупова

О ПРИМЕНЕНИИ ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ В ПРОЦЕССЕ ОБУЧЕНИЯ

В статье обсуждаются проблемы и условия эффективности внедрения в образовательный процесс новых средств обучения – электронных образовательных ресурсов.

Российское образование сегодня находится в процессе модернизации, которая в значительной мере обусловлена тем, что традиционное понимание цели образования и сущности образовательных результатов стали все в меньшей степени адекватны интересам личности, новым социальным ожиданиям и требованиям государства [1].

К системе современного образования предъявляются новые требования, направленные в первую очередь на достижение качественно новых образовательных результатов. При этом под об-

ИСУПОВА Наталья Ивановна – кандидат педагогических наук, старший преподаватель кафедры прикладной информатики ВятГГУ
© Исупова Н. И., 2008

Классификация ЭОР по видам учебной деятельности и ожидаемым образовательным результатам

Тип ЭОР	Иницилируемые виды учебной деятельности	Изменения образовательного процесса	Новые планируемые образовательные результаты
Демонстрационные ЭОР	Познавательная деятельность (получение мировоззренческих, ориентировочных, оценочных знаний)	<i>ЭОР, не приводящие к существенному изменению образовательного процесса</i> Повышение уровня наглядности (визуальное представление знаний на уровне репродукции) в рамках традиционного образовательного процесса. Воспроизводство демонстрационной части учебного процесса в динамике	Формирование познавательной потребности; повышение мотивации у обучаемых
Информационно-справочные, информационно-поисковые системы, базы данных и знаний, электронные библиотеки и др.	Познавательная деятельность (получение мировоззренческих, ориентировочных, оценочных знаний)	Реализуется существенное расширение функционала традиционной образовательной среды за счет расширения возможностей предоставления учебной информации с привлечением средств технологий мультимедиа, гипертекста, гипермедиа. Возрастают скорости получения необходимой информации	Формирование аналитических способностей. Перенос на эти средства функции передачи и воспроизводства социального опыта
Контролирующие программы	Регулятивная деятельность	Обеспечение значительной оперативности и повышение объективности контроля учебных достижений и диагностирования интеллектуальных возможностей обучаемых	Развитие умений самопроверки и самооценки. Рефлексия собственной деятельности обучаемых
Компьютерные тренажеры	Предметная деятельность (развитие способностей, умений и навыков в оперировании реальными предметами)	Расширение возможностей формирования типовых умений и навыков. В совокупности с использованием приборного оборудования, датчиков и т. д. добавляется некоторый творческий элемент, обеспечивающий саморегуляцию в деятельности обучаемого. Внедрение осуществляется, как правило, в рамках традиционного процесса обучения	Формирование организационно-практических способностей обучаемых, рефлексивных умений и навыков
Инструментальные программные средства (текстовые и графические редакторы, СУБД, электронные таблицы и т. д.)	Предметная деятельность (развитие способностей, умений и навыков в оперировании реальными предметами)	Обучение наполняется компьютерным инструментарием (инструментальная форма представления знаний), позволяющим расширить некоторые возможности по представлению данных и работы с ними, выполнить разнообразные виды учебной практической деятельности: регистрацию, сбор, хранение, обработку информации в рамках традиционной системы обучения. У обучаемых происходит процесс овладения информационными орудиями труда. Каждый инструментарий обладает узкой функцией, которая направлена на формирование репродуктивных практических умений	Универсализация специалиста (овладение новыми инструментами деятельности, междисциплинарными знаниями, умениями быстро перепрофилироваться)

<i>ЭОР, приводящие к существенному изменению образовательного процесса</i>	
Имитационные и моделирующие ЭОР	<p>Ориентировочно-исследовательская деятельность</p> <p>Коммуникационные (совместные) способы деятельности (пространственное и временное сопричастие участников, обмен действиями, информацией и др.)</p>
Средства компьютерных телекоммуникаций	<p>Появление принципиально новых видов учебной деятельности, связанной с созданием информационных моделей, исследованием их поведения, проведением «компьютерных» экспериментов, умением интерпретировать данные эксперимента</p> <p>Появляются возможности более полной реализации информационных взаимодействий в образовательном процессе, позволяющие строить новые модели и технологии организации учебной деятельности, вести интерактивный диалог, создавать условия для повышения эффективности как коллективно распределенной, так и самостоятельной учебной деятельности студентов</p>
Автоматизированные обучающие системы	<p>Переход к нетрадиционным системам обучения, реализация зачетно-модульной и модульно-рейтинговой технологий обучения. Автоматизация процессов контроля результатов учебной деятельности, тренажа, самостоятельной творческой деятельности обучаемых</p> <p>Познавательная, регулятивная, предметная, коммуникативная, ориентировочно-исследовательская деятельность</p>
Интегрирующие среды обучения	<p>Реализация проектно-творческой деятельности, основанной на выдвижении предположений, гипотез, идей, включающая планирование, сбор дополнительной информации, выполнение и защиту проекта. Приводит к самоопределению, самоактуализации, пику мотивации</p> <p>Создание новой образовательной среды. Возможности инновационного обучения (процесс трансформации – существенное преобразование сложившейся системы обучения и превращение обучаемого в субъект учебной деятельности)</p>
	<p>Создаются условия для реализации новых видов деятельности, стимулирующие познавательную активность обучаемого, исследовательские навыки. Развитие общеинтеллектуальных умений, навыков проектной и исследовательской деятельности</p> <p>Формирование умений работать в коллективе, исследовательских аналитических навыков. Развитие универсальных способностей деятельности (коммуникационных или совместных)</p> <p>Развитие самостоятельности учебной деятельности, умений выбора индивидуального образовательного маршрута, проектирование образовательного процесса</p> <p>Реализация проектной деятельности, в основе которой лежит развитие познавательных навыков обучаемых, любознательности, компетентности, профессионализма, умений самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развитие их системного, критического, прогностического и творческого мышления, умения увидеть, сформулировать и решить проблему.</p> <p>В результате самостоятельной работы обучающего может быть представлен «продукт» инновационной деятельности, который оценивается преподавателем, аудиторией и, возможно, работодателем</p>

разовательными результатами понимаются изменения в личностных ресурсах, которые могут быть использованы при решении значимых для личности проблем [2]. Среди личностных ресурсов можно выделить мотивационные (ценностные ориентации, потребности, запросы и т. д., которые конкретизируются в мотивах деятельности), операциональные (освоенные универсальные и специальные способы деятельности), когнитивные (знания, обеспечивающие возможность ориентации в явлениях действительности, предметные умения и навыки).

Развитию этих личностных ресурсов соответствуют личностные, метапредметные и предметные результаты образования, показывающие уровень сформированности соответствующих компонентов опыта учащихся. Следовательно, сегодня ставится задача выхода системы образования на новое качество, на новые образовательные результаты, связанные с пониманием развития личности как цели и смысла образования.

Чтобы добиться образовательных результатов, отвечающих новым запросам общества, нужны новые средства и построенные на их основе новые технологии обучения. Как показывают проведенные психолого-педагогические и дидактические исследования ([3]), необходимым потенциалом в значительной мере обладают сравнительно новые средства обучения – электронные образовательные ресурсы (ЭОР). Фактически речь идет о создании новой среды обучения на основе средств информационные технологий, направленной на самостоятельную учебную деятельность, развитие творческих способностей и личности обучаемых.

Различные аспекты применения информационно-коммуникационных технологий (ИКТ) в процессе обучения рассматривались в ряде педагогических, психологических, методических исследований [4]. Эти работы в целом создали научные и методические основы применения информационно-коммуникационных технологий в образовательном процессе. Однако их анализ позволяет выявить некоторые противоречия в развитии этой проблемы.

Первое из них связано с тем, что в большинстве работ в качестве основной предпосылки исследований выступают не столько потребности развития процесса образования, сколько возможный дидактический потенциал электронных образовательных ресурсов. В результате этого среди их возможностей используются в основном те, которые «лежат на поверхности» и наиболее просто реализуемы (повышение наглядности, оперативный контроль, тренинг типовых умений, повышение интерактивности). При этом их реальная педагогическая эффективность, как правило, не оценивается и предполагается, что она

очевидна в силу самого использования средств ИКТ как основы данного ресурса.

Второе противоречие касается возможностей использования ЭОР в образовательном процессе. Любая образовательная технология, так или иначе, инициирует определенный вид (или несколько видов) учебной деятельности. Электронные образовательные ресурсы также направлены на активизацию различных видов деятельности. Однако в настоящее время не существует критериев, позволяющих соотнести педагогические технологии и электронные образовательные ресурсы, отобрать средства обучения на базе ИКТ, обеспечивающие реализацию тех видов учебной деятельности, которые являются приоритетными для данной образовательной технологии.

Разрешением указанных противоречий может стать внедрение в учебный процесс новых видов ЭОР, которые, во-первых, будут созданы исходя из потребностей образовательного процесса, во-вторых, будут ориентированы на новые образовательные результаты.

Анализ литературы, посвященной описанию планируемых образовательных результатов [5], показывает, что современное образование испытывает существенную потребность в качественных электронных образовательных ресурсах, которые на практике позволили бы:

- 1) усилить мотивацию, повысить интерес и расширить познавательные потребности обучаемых;
- 2) обеспечить индивидуализацию обучения, создать предпосылки для перехода к личностно-ориентированному обучению;
- 3) повысить интерактивность обучения, развивать диалогический характер учебного процесса;
- 4) расширить круг учебных задач, используемых в обучении, включить в арсенал методов познавательной деятельности ряд новых методов, основанных на использовании средств ИКТ (информационное моделирование, вычислительный эксперимент и т. д.);
- 5) создать возможности для использования новых источников учебной информации (информационно-справочные системы, электронные энциклопедии, файловые архивы, ресурсы Интернета и др.);
- 6) повысить оперативность контроля результатов обучения, создавать базы данных учебных достижений обучаемых;
- 7) обеспечить возможность «погружения» обучаемых в виртуальную среду с возможностью имитации учебных и профессиональных ситуаций, инициирующих проявление готовности к решению возникающих проблем.

В связи с этим весь спектр электронных образовательных ресурсов можно разделить на две группы: ресурсы, нацеленные на поддержку тра-

диционных видов учебной деятельности и не приводящие к существенному изменению учебного процесса, и ресурсы, ориентированные на новые виды учебной деятельности, способствующие достижению качественно новых результатов образовательного процесса. Общая характеристика конкретных видов ЭОР, относящихся к каждой из указанных групп, приведена в таблице.

Сопоставляя данные таблицы, можно сделать вывод, что наиболее приоритетными видами электронных ресурсов являются такие, которые способствуют формированию следующих компонентов опыта учащихся:

– среди личностных результатов образования – мотивация, познавательные потребности, творческие способности обучаемых (развиваются с помощью демонстрационных, информационно-справочных, информационно-поисковых систем, контролирующих программ, тренажеров, имитационных и моделирующих ЭОР);

– среди операциональных (надпредметных) результатов – развитие универсальных способов деятельности, общеинтеллектуальных умений и навыков, включая, прежде всего, навыки проектной и исследовательской деятельности, эффективного использования ЭОР (осуществляется с помощью моделирующих программ; средств компьютерных телекоммуникаций, интегрирующих сред обучения);

– среди предметных (когнитивных) результатов – переход от репродуктивных способов усвоения учебного материала к формированию навыков самостоятельной познавательной деятельности – поиска информации, ее анализа, проведению экспериментов, обработки и интерпретации их результатов и т. д. (осуществляется с помощью информационно-справочных, информационно-поисковых систем, баз данных, средств компьютерных телекоммуникаций, интегрирующих сред обучения).

Данный подход позволяет выделить типы ЭОР, использование которых в образовательном процессе приоритетно для развития ресурсов личности, наиболее важных для будущих специалистов определенного профиля. Например, для специалистов в области естественных наук наиболее востребованы предметно-ориентированные программные среды, предназначенные для моделирования изучаемых объектов и явлений, позволяющие обрабатывать результаты учебных экспериментов и др. Для гуманитарных направлений наилучшая поддержка будет оказана средствами компьютерных телекоммуникаций, а также информационно-справочными и информационно-поисковыми системами.

Кроме того, наличие в данной классификации такого фактора, как иницилируемые виды деятельности, позволяет решить второе из обозначен-

ных выше противоречий, а именно, соотносить педагогические технологии и электронные образовательные ресурсы. Строя образовательный процесс, преподаватель для решения различных задач выбирает разные педагогические технологии, создавая тем самым собственную, авторскую методику, и под нее может выбирать электронные образовательные ресурсы, соответствующие выбранной технологии (т. е. иницилирующие те же виды деятельности). Правильный выбор ЭОР и грамотная организация образовательного процесса на основе их использования являются условиями, обеспечивающими эффективность процесса обучения в плане достижения новых образовательных результатов.

Таким образом, электронные образовательные ресурсы нового типа целесообразно разрабатывать с ориентацией на приоритетное формирование у обучаемых умений и способностей, соответствующих планируемым результатам современного образования. При этом акцент должен быть перенесен с создания средств, поддерживающих традиционный образовательный процесс и репродуктивные методы обучения (повышение уровня наглядности, оперативный контроль усвоенных знаний, тренинг типовых умений и т. д.), на обеспечение внедрения методов и средств, ориентированных на развитие универсальных умений, творческих, исследовательских способностей учащихся (умения находить, обрабатывать и использовать информацию, создание информационных моделей изучаемых процессов и объектов, эксперименты с ними и т. д.).

Анализ дидактических возможностей ЭОР позволяет сформировать системный и функционально полный перечень их типов, обеспечивающих реализацию определенных видов учебной деятельности. Такая классификация позволит создать гибкую структуру банка электронных образовательных ресурсов, позволяющую осуществлять грамотный отбор средств, адаптированных к особенностям конкретной образовательной технологии.

Примечания

1. Концепция модернизации российского образования до 2010 года: Распоряжение правительства РФ от 29.12.01 № 1756-Р [Текст] // Бюл. Минобразования России. 2002. № 2.

2. Стандарт общего образования: требования к результатам освоения основных общеобразовательных программ [Текст] / под общ. ред. А. М. Кондакова, А. А. Кузнецова. М.: «Просвещение», 2006.

3. Константиновский, Д. А. Мониторинг потребностей личности, общества и государства в общем образовании [Текст] / Д. А. Константиновский, В. С. Вахштайн, Д. Ю. Куракин. М.: «Просвещение», 2006; Лапчик, М. П. Методика преподавания информатики [Текст]: учеб. пособие для студ. пед. вузов / М. П. Лапчик, И. Г. Семакин, Е. К. Хеннер; под общ. ред. М. П. Лапчика. 2-е изд., стер. М.: «Академия»,

2005. 624 с.; *Ракитина, Е. А.* Теоретические основы построения концепции непрерывного курса информатики [Текст] / Е. А. Ракитина. М., 2002. 88 с.; *Роберт, И. В.* Информационные технологии в науке и образовании [Текст]: учебно-методическое пособие / И. В. Роберт, Я. И. Самойленко. М.: Моск. гос. заочный ин-т пищевой промышленности Мин-ва общ. и проф. образования РФ, 1999.

4. *Окулов, С. М.* Информатика: развитие интеллекта школьника [Текст] / С. М. Окулов. М.: Бином, Лаборатория знаний, 2005; *Панюкова, С. В.* Информационные и коммуникационные технологии в личностно ориентированном обучении [Текст] / С. В. Панюкова. М.: «Прогресс», 1998; *Полат, Е. С.* Метод проектов в современной школе. Методология учебного проекта [Текст] / Е. С. Полат. М.: МИПКРО, 2000; *Филатова, Л. О.* Развитие преемственности школьного и вузовского образования в условиях введения профильного обучения в старшем звене средней школы [Текст] / Л. О. Филатова. М.: Лаборатория Базовых Знаний, 2005. 192 с.

5. *Константиновский, Д. А.* Указ. соч.; Стандарт общего образования: требования к результатам освоения основных общеобразовательных программ...

Н. В. Исупова, В. В. Кривокорытова

ОБ ЭФФЕКТИВНОСТИ ОБУЧЕНИЯ СТУДЕНТОВ ГУМАНИТАРНЫХ СПЕЦИАЛЬНОСТЕЙ

В статье рассмотрены способы повышения эффективности обучения за счет внедрения новых форм контроля и проверки знаний.

Понятие эффективности обучения давно используется в педагогической практике. Теоретическое же обоснование данного термина до сих пор нуждается в уточнении.

Понимание сущности эффективности обучения связано с тенденциями развития информационного общества. Современное информационное общество – это общество, основанное на знаниях, а высшее образование – это механизм создания нового информационного общества.

Основные тенденции раскрытия сущности эффективности обучения представлены информационными, операциональными, личностно-ориентированными моделями.

1. Информационная модель

Эффективностью обучения считается мера результативности обучения, которая определяется сравнением начального и конечного результатов

ИСУПОВА Наталья Валентиновна – старший преподаватель кафедры прикладной информатики ВятГГУ

КРИВОКОРЫТОВА Валентина Викторовна – старший преподаватель кафедры прикладной информатики ВятГГУ

© Исупова Н. В., Кривокорытова В. В., 2008

(Р. А. Кричевский, В. Я. Ляудис, Я. А. Пономарёв и др.).

2. Операциональная модель

Эффективность – это мера успешности овладения компонентами учебной деятельности (успешности, действенности, оперативности, продуктивности) (С. Ф. Жуков, А. А. Люблинская, В. М. Манахов и др.).

3. Личностно-ориентированная модель

Эффективность – это мера изменений в позициях, направленности личности участников процесса обучения. Предполагается возможность развивающего влияния обучаемого на личность преподавателя, на саморазвитие последнего (Г. А. Балл, И. С. Якиманская, М. С. Бургин и др.).

Таким образом, можно привести следующую обобщенную формулировку сущности эффективности обучения.

Эффективность обучения – это мера результативности, успешности, действенности, оперативности, продуктивности учебной и педагогической деятельности и мера изменений в позициях, в направленности личности участников процесса обучения.

Рассмотрим пути повышения эффективности обучения студентов в период становления информационного общества.

Фундаментальные изменения, происходящие в российской экономике, социальной сфере, политике, требуют нетрадиционных подходов к организации учебного процесса. Такой подход должен заключаться в информатизации образования и основываться на новейших достижениях науки и техники, которые дают возможность сочетать традиционные и новые современные технологии в обучении специалиста.

Информатизация образования – это комплекс мероприятий, направленных на обеспечение сбора, обработки, систематизации, накопления, хранения, передачи информации и обмена между участниками системы образования с помощью ПК и средств телекоммуникации в целях интенсификации функционирования системы образования.

При переходе к информационному обществу в области образования необходимо решать следующие задачи:

- Совершенствовать компьютерную технику.
- Осуществлять выбор информационно-коммуникационных технологий, адекватных образовательным технологиям.
- Создавать информационные ресурсы: базы данных, электронные библиотеки, электронные гиперссылочные и мультимедийные учебные материалы, архивы, информационно-обучающие сайты.
- Организовать центры подготовки и переподготовки преподавателей, обладающих высокой

информационной культурой, готовых и умеющих применять новые информационные технологии в процессе обучения и управления образованием.

Для решения этой проблемы необходимо создание единой информационной образовательной среды, которая комплексно объединит материально-техническую базу, основанную на локальных компьютерных сетях, включающих автоматизированные рабочие места учащихся, преподавателя и линии связи между ними. Рабочее место учащегося может содержать принтер, сканер, элементы мультимедиа, такие, как динамики, микрофоны. Предполагается возможность выхода в Интернет.

Высшая школа в процессе своего развития выработала основные формы и методы обучения студентов в процессе преподавания учебных дисциплин: лекции, семинары, практические и лабораторные занятия. Это относится и к предмету информатика. Преподаватели обычно уделяют максимальное внимание содержательной стороне дисциплины, и заметно в меньшей степени это внимание обращено к проблемам использования компьютера как средства обучения информатике, а также к выбору методов и форм обучения информатике с помощью ПК. Эффективное использование компьютера в этом направлении возможно лишь при определенной организации учебного занятия. Преподаватель должен так организовать самостоятельную деятельность студентов, чтобы студент из субъекта, усваивающего готовые суммы знаний, становился субъектом познающим и приобретающим опыт самостоятельного познания в следующих направлениях:

- Самостоятельно формулировал цели своей деятельности.
- Ориентировался в содержании и предмете деятельности.
- Умел выбрать средства деятельности, активно использовать справочный материал. Преподаватель при этом не должен спешить с помощью или с подсказкой.
- Видел итоговый результат своей деятельности.

Для организации таких занятий преподавателю необходимо разработать учебно-методические пособия, в которых предоставить инструкции по выполнению заданий. Сначала это очень подробные, пошаговые указания деятельности, адаптированные на первичное восприятие и выполнение. Затем эти инструкции становятся более краткими, с использованием терминологии и языка, принятого в информатике. Также желательно в ходе работы привлекать студентов к использованию справки и учить их ею пользоваться.

Самостоятельность в работе студентов высвобождает время преподавателя, которое можно использовать для индивидуальной помощи тем

студентам, которые имеют низкий уровень начальной подготовки по информатике, и тем, которые не имеют возможности упражняться дома.

Для эффективности таких занятий можно использовать модульную технологию. Структура модуля позволяет представить содержание курса в различных вариантах: полном и углубленном. Первый вариант рекомендуется для «средних» студентов, второй – для «сильных». Оба варианта дают положительный результат – усвоение студентами знаний. При выборе варианта модуля нужно учитывать особенности профессиональной подготовки студентов. Учебный материал разбивается на блоки и наполняется заданиями с учетом будущей профессии студента. Это – контекстное обучение.

Контекстное обучение предполагает ориентацию студента на профессиональную деятельность, то есть студент должен получить информацию по предмету в контексте своей будущей профессии. При таком подходе студенты овладевают знаниями, умениями и навыками применения информационных технологий в профессиональной деятельности за счет системы профессионально-направленных задач.

С целью реализации этого подхода для студентов заочной формы обучения экономических специальностей нами были разработаны методические рекомендации по изучению программного обеспечения ПК с подбором заданий, учитывающая специфику их будущей специальности. При этом:

- студент повышает уровень информационной культуры, так как сам участвует в поиске и обработке информации;
- осознает практическую целесообразность предмета информатики, так как видит способы применения полученных знаний в своей профессиональной деятельности;
- реализуется принцип индивидуализации обучения, когда студент выбирает свой темп изучения материала и выполнения упражнений.

Таким образом, построение процесса контекстного обучения на модульной основе позволяет:

- Интегрировать и дифференцировать содержание обучения путем группировки модулей учебного материала.
- Акцентировать работу преподавателя на консультативно-координирующей функции управления познавательной деятельностью студентов.
- Сократить курс обучения без особого ущерба для полноты изложения и глубины усвоения учебного материала.

В последнее время в стандартах высшего образования усиливается тенденция к увеличению объема знаний курса, с одной стороны, и уменьшение количества часов, отводимых на его освоение – с другой.

Для повышения эффективности обучения (усвоения большего объема знаний за меньшее время) можно предложить использование параллельного способа обучения, который мы применяем на учебной практике по информатике. Создаются рабочие мини-группы студентов в рамках организованной проектной деятельности. Каждая группа параллельно отрабатывает свой раздел (например, решение индивидуальной практической задачи, в процессе которого происходит исследование дополнительных возможностей изученных программных средств: Microsoft Windows, стандартных Windows-приложений, Microsoft Word, Microsoft Excel, Microsoft Access).

Таким образом, использование параллельного способа обучения позволяет интенсифицировать учебный процесс. Кроме этого подобная методика существенно повышает мотивацию студентов к обучению, учит их работать в коллективе.

Большую значимость в повышении эффективности обучения мотивации студентов к обучению приобретает балльно-рейтинговая система контроля успеваемости. Нами разработана по курсу информатики для бакалавров по специальности «Коммерция» система рейтинговых оценок всех контролируемых мероприятий и достижений студентов в обучении. На основании итоговой рейтинговой оценки студенты имеют возможность получить соответствующую набранным баллам оценку на экзамене без сдачи экзамена.

Наиболее перспективным вариантом организации обучения является дистанционное обучение с использованием технологий глобальной сети Интернет. В основе дистанционной формы обучения лежит самостоятельная работа студентов с образовательным контентом. Контент – информационно значимое наполнение Интернета – тексты, графика, мультимедиа; существенными параметрами контента являются его объем, актуальность и релевантность (смысловое соответствие между информационным запросом и полученным сообщением).

Многочисленные эксперименты, проводимые в вузах РФ, показывают высокую эффективность дистанционного обучения (ДО) при заочной форме обучения. Но для успешной реализации учебного процесса самостоятельная работа студентов должна быть обеспечена методической помощью и образовательным контентом (учебно-методические комплексы, электронные учебники, видеолекции, мультимедиа лекции, Интернет-библиотека с наглядно структурированным представлением информации, программные средства учебного назначения, обучающие и контролирурующие программы) и должным контролем со стороны преподавателя и методиста. В системе ДО используются практически все возможные организационные

формы контроля, дополненные специально разработанными компьютерными программами, позволяющими усилить эффективность и своевременность контроля. Контроль должен обеспечивать и контролируемую и управляющую функции. Применение информационно-коммуникационных технологий (ИКТ) расширяет возможности контроля учебного процесса.

Для повышения эффективности обучения актуален опыт использования такой формы контроля, как тесты. Можно выделить два основных варианта проведения тестов: тесты для самоконтроля и тесты для контроля преподавателем, в том числе и с возможностью вывода на экран соответствующей справки при неверном ответе, а также проведение письменного экзамена с отправкой ответов по e-mail для проверки знаний при ДО.

Использование тестов невозможно без внедрения автоматизированных систем обучения. Одним из примеров такой реализации является применение программного комплекса TestBOX (программный продукт «Парус»), который рекомендует к применению Кировский институт повышения квалификации и переподготовки работников образования.

TestBOX – это комплексная обучающая система – идеальное решение для автоматизации учебного процесса в рамках образовательных и других учреждений (вузы, колледжи, школы, отделы кадров предприятий и т. п.). Позволяет организовывать дистанционное, заочное обучение, создавать тесты, электронные учебники, курсы.

Программный пакет состоит из 8 модулей.

- TestBuilder – редактор тестов, позволяет создавать тесты и курсы любых типов и сложности.

- Administrator – позволит провести анализ результатов как по одному пользователю, так и по целой группе, представить результаты в виде таблиц и создать отчет.

- CourseBuilder – системы создания электронных учебников.

- TestReader – может запустить и пройти любой созданный тест или курс. Обработать результаты тестирования и сохранить их.

- Standalone builder – создание автономных тестов, курсов.

- TestBox direct – система электронного обслуживания и сопровождения.

- Collection Builder – система компоновки электронных курсов и тестов.

- Scripted Question – инструмент позволяет разрабатывать задания любой сложности, не ограничиваясь стандартными типами заданий.

Приведем в качестве примера некоторые возможности использования TestBOX в образовательных учреждениях.

Данная система позволит создавать обучающие курсы по разным методикам преподавания и, как следствие, более эффективно обучать. Вариантов создания курсов может быть несколько. Это зависит от методики, какую применяет преподаватель.

Первый вариант – поисковая система обучения. При такой системе обучения студент при запуске курса через TestReader или EXE-файл работает в двух окнах: в окне с теоретическим материалом и в окне с контрольными тестами, на которые ищет ответы в теории. В программе реализована возможность «нечеткой логики». Таким образом, даже неполные и неточные ответы студентов на вопросы теста могут быть оценены преподавателем. Данная система считается наиболее эффективной для обучения. Теоретический материал может быть представлен в любом формате, это могут быть файлы doc, rtf, xls, pdf, mp3 или любой мультимедийный формат.

Второй вариант – тренажер. При такой системе обучения студент может пользоваться любым теоретическим материалом (книгой). При прохождении курса студент отвечает на контрольные вопросы. При неправильном ответе «выскакивает» файл с теоретическим материалом, содержащим развернутый ответ.

Третий вариант – это совмещенная система первого и второго вариантов.

Как организовать заочное обучение на TestBox? Допустим, готовый учебный курс для самостоятельного изучения можно записать на CD и предоставить его студентам, ограничив его использование по времени (семестр, количество запусков). При наличии компьютерной техники в районах или филиалах можно быстро установить EXE-файлы (курсы) на все имеющиеся компьютеры путем простого копирования с диска.

Другой вариант – готовые EXE-файлы курсов можно выложить на сайт для студентов. Такое использование курсов в рамках Интернет-образования является более дешевым (для студентов).

Кроме рассмотренной выше комплексной обучающей системы TestBOX можно использовать следующие тестовые оболочки: UniTest, TestGold, VeralTest, NetOp School и др.

При переходе к информационному обществу в области образования меняются и требования к преподавателю. Теперь это уже не просто хороший специалист, но и профессиональный педагог, владеющий информационно-коммуникационными технологиями в образовании, готовый и умеющий создать информационные ресурсы: базы данных, электронные библиотеки, электронные гиперссылочные и мультимедийные учебные материалы, архивы, информационно-обучающие сайты. А для этого необходимо постоянное совершенствование информационной культуры педагога путем самообразования и повышения квалификации на курсах.

В. Ю. Колеватов

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ЗАДАЧАХ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СЕТЕЙ

В статье приведен обзор методов искусственного интеллекта, используемых в сфере сетевой безопасности, в том числе многоагентные системы, продукционные правила, нейронные сети и машины опорных векторов.

Введение

В настоящее время компьютерные сети представляют собой большие распределенные системы устройств, взаимодействующие между собой с целью обмена, хранения и обработки информации. Компьютерная сеть – прекрасная возможность для развития бизнеса компаний. Многие уже открыли свои сайты-представительства в Internet и практически все работники офисов используют для своей работы преимущества совместной работы. Сети объединяют разные типы устройств, соединенных по различным каналам связи. Усложнение структуры сетей, повышение нагрузки на них, регулярное появление новых методов нарушения корректной работы, способы ограничения доступа к данным их законных владельцев и целостности информации, превращение сферы создания нежелательного ПО (нежелательным программным обеспечением (malicious software) будем считать совокупность вредоносного, шпионского, рекламного ПО, спама и других нежелательных для пользователя событий) из любительского занятия в прибыльный бизнес определяют необходимость серьезного отношения к вопросам сетевой безопасности.

В целях обеспечения защищенности компьютерных сетей были созданы системы, которые классифицируют сетевую активность различных программ. В случае совпадения с определенной экспертом ситуацией такие системы предлагают пользователю прекратить действия возможно вредоносного ПО и откатить произведенные им изменения. Однако большинство современных систем сетевой безопасности не имеют возможности самообучения и оперируют только заданными в них вручную правилами. Частое появление нежелательного ПО, использующего новые уязвимости, повысили требования к системам сетевой безопасности.

Применение методов искусственного интеллекта (ИИ) позволяет ввести в системы защиты свойство самообучения и обеспечивает обнаружение угроз «на лету». В данной статье приво-

КОЛЕВАТОВ Василий Юрьевич – аспирант кафедры информатики и методики обучения информатике ВятГГУ

© Колеватов В. Ю., 2008

дится обзор методов ИИ, используемых для защиты компьютерных сетей.

Многоагентные системы

Долгое время ведутся разработки персональных средств защиты, предназначенных для одной единицы оборудования. Так как защита сети – задача комплексная, то помимо средств персональной защиты (антивирусы, сетевые экраны и т. д.) ведутся разработки многоагентных систем. Каждый агент отвечает за определенную часть задания, и общее решение возникает в результате их скоординированных действий. Агенты могут иметь реализованные элементы интеллекта, а могут и не иметь таковых. В процессе работы агенты обмениваются сообщениями по специальным протоколам. Обычно существует агент, который управляет действиями других. Перечислим особенности многоагентных систем, которые позволяют эффективно их использовать в системах сетевой безопасности:

– *Гибкость*. Агенты могут создавать себе подобных и размещать их на новых узлах сети, поэтому многоагентные системы легко адаптируются к любой сетевой архитектуре и адекватно отвечают на изменения в конфигурации сетевого оборудования.

– *Экономичность*. Система равномерно распределена по всему периметру защиты. Эта особенность обеспечивает оптимальное распределение вычислительных ресурсов сети.

– *Повышенная отказоустойчивость*. Так как агенты могут существовать самостоятельно и они распределены на всех узлах сети (т. е. система защиты не имеет центра), то атаковать ее будет сложнее, нежели сеть с централизованным сервером защиты. Распределенная по сети информация и защита требуют от злоумышленника проводить атаку многих узлов одновременно.

– *Возможность централизованного администрирования*. Внесение изменений в работу агентов может производиться централизованно и по протоколам взаимодействия агентов передаваться на все точки обеспечения безопасности.

Общая теория многоагентного подхода подробно рассмотрена в [1].

В [2] рассмотрены многоагентные системы защиты сети от внешних угроз. Особенности подхода, описанного в этих статьях, обеспечивают защиту от сложных угроз и позволяют наглядно представить текущее состояние всех агентов и сети в целом. Агенты разделены по роду деятельности и объединены в команды. Например, в [3] выделены следующие классы агентов команд защиты: обработки информации («сэмплеры»), обнаружения атаки («детекторы»), фильтрации («фильтры»), агенты расследования. Сэмплеры осуществляют сбор данных для последующего

обнаружения сетевых аномалий или злоупотреблений детектором. Фильтры ответственны за фильтрацию трафика по правилам, представленным детектором. Агент расследования пытается обезвредить агентов атаки. Команда агентов защиты совместно реализует механизм защиты и может взаимодействовать по различным схемам. В одной из схем при обнаружении начала атаки действует детектор команды агентов, на сеть которой направлена атака. Он посылает запрос агентам-сэмплерам других команд с целью получения информации, которая может быть релевантной указанной атаке. Сэмплеры отвечают на запрос, отправляя необходимые данные. Эта информация существенно повышает шансы на обнаружение атаки. В случае обнаружения вероятного источника атаки детектор сети-жертвы посылает информацию об адресе агента атаки детектору команды, в сети которой может находиться этот агент, с целью его деактивации.

Для тестирования защиты сети и открытия новых видов атаки необходимо уметь моделировать сложные комплексные атаки, например DDoS (распределенная атака отказа в обслуживании). Системы моделирования атаки описаны в [4].

«Осведомлен – значит защищен» – такой девиз подходит для специалистов в области сетевой безопасности, ведь гораздо дешевле и проще с помощью собственного средства анализа сети найти возможные недостатки архитектуры, настройки сети и ПО, чем ждать успешной атаки злоумышленника. Системы анализа защищенности сетей описаны в [5]. Они позволяют проверять различные уровни сети. Механизм их работы основан на сканировании устройств в сети и попытке осуществления известных атак на сеть [6].

Рассмотрим методы, применяемые для реализации защиты сетей и их адаптации к новым угрозам.

Производственные системы

Производственными называют системы, основанные на правилах вида «ЕСЛИ... ТО...». Часть «ЕСЛИ» описывает условие применения правила, а часть «ТО» – последствия использования правила. В результате применения правила может возникнуть событие одного из двух типов:

1. Получение нового знания.
2. Выполнение некоторого действия для изменения окружающей среды.

Правила хранятся в базе знаний, начальное заполнение которой производится экспертом. Правила еще называют эвристиками. К преимуществам такого подхода можно отнести устойчивость производственных систем и точность обнаружения известных типов угроз сетевой безопасности. К недостаткам относят поражение сети атаками zero-day (атака через уже обнаружен-

ные, но еще не закрытые бреши в программных продуктах).

Приведем пример правила: «ЕСЛИ [(процесс использует библиотеки для работы с сетью) и (количество обнаруженных сигнатур, типичных для отправки почты)>X] ТО записать в рабочую память факт <программа работает с электронной почтой>». В данном случае выполнение условия приводит к формированию нового знания, которое, в свою очередь, может быть использовано другими правилами.

Производители продукционных систем обычно поддерживают глобальную базу знаний, формируемую экспертами и программами, которую можно обновлять через Интернет.

Искусственные нейронные сети

Искусственная нейронная сеть является упрощенной моделью мозга человека и представляет набор нейронов, соединенных между собой определенным образом. Нейронные сети позволяют решать различные практические задачи, в частности связанные с распознаванием образов, поддержкой принятия решений, классификацией текстов. Основа сети – искусственный нейрон, являющийся упрощенной математической моделью нейрона головного мозга. Несомненные преимущества нейронных сетей в том, что они могут автоматически приобретать знания в процессе обучения и обладают способностью к обобщению.

Любое действие пользователя или приложения должно быть представлено в виде набора признаков, которые подаются на вход нейронной сети. В результате прохождения сигнала по нейронной сети мы получим число, по которому определяют, является ли действие вредоносным.

Входные сигналы $X_1...X_n$ умножаются на весовые коэффициенты w_i ($i = 1..m$). При $w_i > 0$ образуется возбуждающая связь нейронов, а при $w_i < 0$ – тормозящая связь. После умножения на весовые коэффициенты входные сигналы суммируются и подаются на вход нелинейного преобразователя, реализующего функцию одного аргумента, которую называют функцией активации, или передаточной функцией нейрона. Таким образом, происходит прохождение сигнала по сети.

Рассмотрим применение нейросети на небольшой практической задаче из области информационной безопасности (пример взят из [7]). При помощи нейросети необходимо детектировать доступ к базе данных со стороны некоего ПО, отличного от автоматизированного рабочего места (АРМ) пользователя, или определить аномалии в работе пользователя. Нейронная сеть имеет четыре входа, на которые подаются:

1. Объем информации, загружаемой из базы данных за контрольный период V (в килобай-

тах). Полученное значение необходимо нормализовать, поскольку считываемый из базы данных объем заранее не известен и индивидуален для каждой задачи и для каждого пользователя. В качестве нормализации можно применить оценку трафика по десятибалльной шкале (0 – объем равен нулю, 10 – максимальный объем трафика).

2. Количество транзакций в минуту C .

3. Количество операций модификации данных в минуту M . В этом примере АРМ использует «короткие транзакции», то есть в рамках одной транзакции обычно бывает 1–2 операции модификации данных.

4. Признаки обращения к словарю базы данных. Большинство клиентских АРМ к словарю не обращается, что отличает их от средств разработки и администрирования. Признаки будут дискретными (0 – нет обращений, 1 – есть), и их будет несколько – по одному на каждую из таблиц словаря базы.

В данном примере используется гомогенная двухслойная нейронная сеть без обратных связей, содержащая один скрытый слой из двух нейронов и выходной слой с одним нейроном.

Обучение нейронной сети можно произвести с помощью существующих пакетов (например, пакет Deductor Lite [8]) или известных алгоритмов (например, метод «обратного распространения ошибки»). Для качественного обучения такой сети необходимо около 300 обучающих примеров. Следует отметить, что подготовка обучающей выборки является достаточно сложным этапом.

Выход нейронной сети может быть интерпретирован как процентное соответствие текущих действий действиям хакера.

Таким же способом можно организовать определение различных атак и адаптацию к новым типам угроз.

Нейронные сети позволяют обеспечить функцию обучения системы безопасности и корректировку сети в случае возникновения ошибок в ее работе. Однако обучаемость нейронной сети одновременно является и недостатком – анализ обученной нейросети достаточно сложен, и, по сути, она является для пользователя «черным ящиком» с определенным количеством входов и выходов.

В отличие от продукционных систем, хранение нейронной сети в рабочей памяти вычислительной машины требует гораздо меньше памяти и определение вредоносных действий требует меньше вычислительных ресурсов, чем использование продукционных систем. Это является преимуществом, поскольку производители стремятся минимизировать размер возможных обновлений для систем безопасности.

Машины опорных векторов

Машины опорных векторов (Support vector machines, SVM) были описаны в работах В. Н. Вапника [9]. SVM – это математический метод получения функции, решающей задачу классификации. Пусть X – характеризующий событие вектор признаков. В процессе обучения формируется оптимальная гиперплоскость $f(X)$ в пространстве векторов-признаков, определяющая событие как допустимое или нежелательное.

Машины опорных векторов зарекомендовали себя в распознавании рукописного текста и задачах классификации. Ведутся разработки по использованию этого метода в системах обеспечения сетевой безопасности. Например, в [10] описана методика определения нежелательного программного обеспечения по метрической удаленности от геометрического центра векторов-признаков событий компьютерной сети с помощью машины опорных векторов.

Для определения атак нужно сформировать вектор признаков, подобный вектору, который формируется для искусственной нейронной сети. Затем с помощью программного обеспечения, например SVM Light [11], произвести обучение машины опорных векторов. В результате получится функция, которая будет производить классификацию векторов-признаков.

Методы использования и обучения SVM в сфере сетевой безопасности еще до конца не изучены. Это задача дальнейших исследований.

Заключение

В работе описаны архитектурные и методологические способы использования методов искусственного интеллекта для защиты от нежелательного программного обеспечения. В своих разработках производители средств защиты компьютерных сетей применяют разные подходы. Общей тенденцией является направление на интеграцию средств защиты различных уровней (например, персональный антивирус и сетевой экран уровня предприятия) с использованием средств искусственного интеллекта. Формирование многоагентной системы, которая имеет единый центр обработки информации, с реализованной функцией обучения на лету, работает на всех уровнях сетевой безопасности от отслеживания вредоносного программного обеспечения на пользовательском компьютере до отслеживания потоков данных корпоративной сети, является решением описанных проблем сетевой безопасности. Реализация этой идеи – задача будущих исследований.

Примечания

1. Люгер, Д. Ф. Искусственный интеллект, стратегии и методы решения сложных проблем

[Текст] / Д. Ф. Люгер. 4-е изд. Вильямс, 2003. 864 с.; Рассел, С. Искусственный интеллект, современный подход [Текст] / С. Рассел, П. Норvig. Вильямс, 2007. 1424 с.

2. Gorodetski, V. I. Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning [Text] / V. I. Gorodetski, I. V. Kotenko, O. Karsaev // International Journal of Computer Systems Science & Engineering. 2003. № 4. С. 191–200; Котенко, И. В. Кооперативная работа команд агентов при защите от сетевых атак нарушения доступности [Электронный ресурс] / И. В. Котенко, А. В. Уланов. Режим доступа: www.comsec.spb.ru

3. Котенко, И. В. Указ. соч.

4. Kotenko, I. V. Simulation of Internet DDoS Attacks and Defense [Electronic resource] / I. V. Kotenko, A. V. Ulanov. Режим доступа: www.comsec.spb.ru; Котенко, И. В. Модели действий хакеров-злоумышленников при реализации распределенных многошаговых атак [Электронный ресурс] / И. В. Котенко, М. В. Степашкин. Режим доступа: www.comsec.spb.ru; Котенко, И. В. Модели и методика интеллектуальной оценки уровня защищенности компьютерной сети [Электронный ресурс] / И. В. Котенко, М. В. Степашкин, В. С. Богданов. Режим доступа: www.comsec.spb.ru; Котенко, И. В. Моделирование противоборства программных агентов в Интернете: общий подход, среда моделирования и эксперименты [Текст] / И. В. Котенко, А. В. Уланов // Защита информации. INSIDE. 2006. № 4. С. 2–10.

5. Багиров, Т. К. Автоматизированная система анализа защищенности корпоративной вычислительной сети на основе многоагентного подхода [Электронный ресурс]: дис. ... канд. техн. наук: 05.13.19 / Т. К. Багиров. Уфа: РГБ, 2007. 204 с.; Богданов, В. С. Анализ выполнения политики безопасности в компьютерных сетях: проактивный подход [Электронный ресурс] / В. С. Богданов, И. В. Котенко. Режим доступа: www.comsec.spb.ru; Котенко, И. В. Многоагентные технологии анализа уязвимостей и обнаружения вторжений в компьютерных сетях [Текст] / И. В. Котенко // Новости искусственного интеллекта. 2004. № 1. С. 56–72.

6. Лукацкий, А. В. Как работает сканер безопасности? [Электронный ресурс] / А. В. Лукацкий. Режим доступа: www.citforum.ru

7. Зайцев, О. Нейросети в системах безопасности [Текст] / О. Зайцев // ИТ-Спец. 2007. № 6. С. 54–59.

8. www.basegroup.ru [Электронный ресурс]

9. Vapnik, V. N. The Nature of Statistical Learning Theory [Electronic resource] / V. Vapnik. Springer-Verlag, 1995.

10. Lashkov, P. Intrusion Detection in Unlabeled Data with Quarter-Sphere Support Vector Machine [Electronic resource] / P. Lashkov, C. Schäfer, I. V. Kotenko. Режим доступа: www.comsec.spb.ru

11. Thorsten, J. Making Large-Scale SVM Learning Practical // Lehrstuhl VIII, Kunstliche Intelligenz. Dortmund, 1998.

Е. С. Колупаева

ИЗУЧЕНИЕ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ И ИКТ

Статья посвящена проблемам изучения псевдослучайных чисел в школе. Рассматривается взаимосвязь разных понятий.

Федеральный компонент принятого в 2004–2005 учебном году Государственного образовательного стандарта по Информатике и ИКТ [1] предполагает на ступени среднего полного общего образования (профильный уровень) в теме «Системы счисления» изучение *псевдослучайных последовательностей*. Однако, возможно, что с понятием псевдослучайного числа учащимся придется встретиться намного раньше. Так, при введении статистического определения вероятности в качестве инструмента моделирования случайного эксперимента может использоваться компьютер. Именно такой подход реализован в пособии [2]. Здесь может возникнуть некоторое затруднение. Авторы данного пособия (возможно, не желая нагружать учащихся новыми терминами) утверждают: «Если у вас есть компьютер, то таблицу случайных чисел может заменить датчик случайных чисел, которым снабжены многие программы и языки программирования. Такой датчик по вашему требованию выдает случайное число». В результате такого упрощения у учащихся складывается неверное представление, что с помощью программы на компьютере можно получить не периодическую последовательность случайных чисел. Попробуем разобраться в этой ситуации. Но прежде проясним для себя понятие случайного числа.

На практике под *случайными числами* подразумевают числа, которые могут рассматриваться в качестве значений некоторой *случайной величины*. Как правило, имеются в виду значения случайной величины, равномерно распределенной на промежутке $(0, 1)$, или приближения к таким зна-

чениям, имеющие конечное число цифр в своем представлении.

Таким образом, *последовательность случайных чисел*, если говорить не строго, это такая последовательность чисел, в которой каждое число получено *случайно*, не имея ничего общего с другими числами в последовательности; и появляется с определенной заранее известной *вероятностью*.

Рассмотрим далее основные способы получения последовательностей случайных чисел с помощью компьютера.

Исторически первым являлся *табличный способ* организации случайных последовательностей, когда случайные числа, полученные экспериментальным путем, помещались в память ЭВМ. Недостатками этого способа были конечность таблицы случайных чисел и использование памяти компьютера. Технологический прогресс позволил вернуться к использованию таблиц в 90-е гг., когда достаточно большие случайные последовательности стало возможно размещать на компакт-дисках. В настоящее время для этих целей широко используется Интернет, где можно скачать файлы с тестированными случайными числами, кроме того, существуют серверы публичного доступа, наиболее известный подобный ресурс – <http://www.random.org/>.

Следующим способом получения случайных чисел является *аппаратный способ*. В этом случае специальное устройство – датчик или генератор случайных чисел (основанный в большинстве случаев на использовании шумов радиоэлектронных приборов) – подключается к компьютеру. Недостатками такого метода являются невозможность точно воспроизвести вычисления, кроме того, такие генераторы могут давать сбои, которые сложно выявить.

Алгоритмический способ – это наиболее интересный для нас способ генерации случайных чисел, так как именно с ним связано понятие псевдослучайного числа. В этом случае последовательность случайных чисел формируется по специальному алгоритму. Числа, получаемые таким образом, принято называть *псевдослучайными*, подчеркивая тем самым, что они только кажутся случайными, а на самом деле получены с помощью детерминированного (определенного) алгоритма [3]. Получаемые таким образом последовательности псевдослучайных чисел имеют период, что существенно отличает их от последовательностей случайных чисел. К недостаткам этого способа также относится то, что на генерации чисел затрачивается машинное время. Существенным достоинством является возможность повторения вычислений, поэтому на практике отдается предпочтение псевдослучайным числам, при условии, что метод генерирования был выбран удачно.

Рассмотрим далее несколько примеров формирования случайных последовательностей с помощью алгоритмов.

Джон фон Нейман в 1946 г. первым предложил алгоритмический подход к генерации случайных чисел. Его идея заключалась в том, чтобы возвести в квадрат предыдущее случайное число и выделить средние цифры, поэтому метод получил название *метод середины квадратов*. Например, пусть имеется 4-значное число $x_0 = 5613$. Возведём его в квадрат, получим 31505769, а затем отберем средние 4 разряда, которые и будут являться очередным числом псевдослучайной последовательности $x_1 = 5057$. Главным недостатком этого метода является наличие зависимости между числами последовательности, кроме того, этот метод, как правило, приводит к короткому циклу повторяющихся элементов. Например, каждое появление нуля как числа последовательности приведет к тому, что все последующие числа также будут нулями.

Заметим, что в учебнике [4] учащимся предлагается самостоятельно установить, как ведет себя последовательность, образованная по методу фон Неймана, для двузначных чисел в зависимости от начального числа.

В настоящее время чаще всего используется *линейный конгруэнтный метод*, когда для генерации псевдослучайных чисел используется рекуррентная формула

$$x_{n+1} = (a \cdot x_n + c) \bmod m, n \geq 0,$$

где a – множитель; $0 \leq a < m$; c – приращение; $0 \leq c < m$; m – модуль; $m > 0$; x_0 – начальное значение; $0 \leq x_0 < m$. Числа a , c , m и x_0 выбираются согласно некоторым правилам так, чтобы увеличить длину периода последовательности. В результате получаем числа, которые ведут себя как значения случайной величины, равномерно распределенной на полуинтервале $[0, 1)$. Генераторы такого типа используются во всех приложениях Microsoft [5]. Последовательности псевдослучайных чисел, подчиняющиеся другим законам распределения, получают косвенно, через равномерное распределение, так как прямые методы не практичны.

Чтобы убедиться, что данную последовательность псевдослучайных чисел можно использовать на практике, используют специальные *статистические критерии*. Различают два вида критериев: *эмпирические* критерии, при использовании которых компьютер манипулирует группами чисел последовательности и вычисляет определенные статистики, и *теоретические* критерии, для которых характеристики последовательности определяются с помощью теоретико-числовых методов, основанных на рекуррентных правилах, которые используются для образования последовательности.

Рассмотрим эмпирический критерий χ^2 (*хи-квадрат*). Он является основным методом, используемым в сочетании с другими статистическими критериями. Этот метод был предложен К. Пирсоном в 1900 г., поэтому его еще называют критерием Пирсона. Метод основан на *теореме К. Пирсона*.

Рассмотрим произвольную случайную величину, которая может быть одномерной или многомерной, дискретной или непрерывной. Множество возможных значений ξ обозначим X . Рассмотрим разбиение X на r попарно непересекающихся множеств X_1, \dots, X_r , таких, что

$$P(\xi \in X_j) = p_j > 0, \text{ при } j = 1, 2, \dots, r.$$

$$\text{Очевидно } p_1 + p_2 + \dots + p_r = P(\xi \in X) = 1.$$

Выберем N независимых значений $\xi_1, \xi_2, \dots, \xi_N$ случайной величины ξ и обозначим через v_j количество значений, принадлежащих X_j . Легко видеть, что v_j – случайная величина, распределенная по биномиальному закону, поэтому ее математическое ожидание $M(v_j) = N p_j$.

В качестве меры отклонения «истинных» значений v_j от «теоретических» $N p_j$ удобно выбирать величину, называемую статистикой хи-квадрата,

$$\chi^2_N = \sum_{j=1}^r \frac{(v_j - N p_j)^2}{N p_j}. \quad (*)$$

Теорема. Каковы бы ни были исходная величина ξ и разбиение $X = X_1 + \dots + X_r$ (такое, что все $p_j > 0$), при любом $x > 0$ верно

$$\lim_{N \rightarrow \infty} P(\chi^2_N < x) = \int_0^x k_{r-1}(x) dx,$$

где $k_{r-1}(x)$ – плотность распределения χ^2 с $(r-1)$ -й степенью свободы.

На практике эта теорема используется для проверки гипотезы о законе распределения случайной величины.

Пусть имеется конкретная гипотеза о законе распределения случайной величины ξ . В результате осуществления N независимых экспериментов были получены N значений $\xi_1, \xi_2, \dots, \xi_N$ этой случайной величины (N достаточно велико). Не противоречат ли эти N значений нашей гипотезе?

Будем рассуждать следующим образом. Возьмем какое-нибудь число r и разбиение множества возможных значений X случайной величины ξ на r попарно непересекающихся множеств X_1, \dots, X_r . Исходя из нашей гипотезы можно вычислить вероятности $p_j = P(\xi \in X_j)$. Предположим, что разбиение выбрано так, что все $p_j > 0$. Тогда по значениям $\xi_1, \xi_2, \dots, \xi_N$ нетрудно вычислить величины v_1, v_2, \dots, v_r (в статистике этот этап называется *группировкой значений*) и по форму-

ле (*) – величину χ^2_N . Если наша гипотеза справедлива, то (при достаточно большом N) эта величина достаточно хорошо подчиняется закону распределения χ^2 с $(r - 1)$ -й степенью свободы. Далее по таблице распределения χ^2 для $(r - 1)$ -й степени свободы (см., например, [6]) сравниваем значения χ^2 с вычисленным χ^2_N и определяем вероятность p' . В зависимости от полученного значения p' принимаем или нет гипотезу о законе распределения случайной величины ξ .

Рассмотрим *пример*. Будем бросать две монеты и подсчитывать количество выпавших орлов (эта задача часто приводится как пример, что известные математики тоже ошибаются при решении элементарных задач. Так, Д'Аламбер, решая эту задачу, принял исходы ОО, ОР и РР за равновероятные. На самом же деле равновероятных исходов здесь четыре: ОО, ОР, РО и РР).

Обозначим «орел» – О, а «решку» – Р, тогда возможны следующие варианты

исход опыта	ОО	ОР	РР
вероятность	1/4	1/2	1/4

Моделируем бросание двух монет, результаты занесем в табл. 1.

Таблица 1

Число бросаний двух монет (N)	Исходы опыта					
	ОО		ОР		РР	
	теор. (T1)	практ. (П1)	теор. (T2)	практ. (П2)	теор. (T3)	практ. (П3)
100	25	29	50	46	25	25
500	125	126	250	247	125	127
1000	250	258	500	512	250	230

Найдем для каждого N статистику хи-квадрат по формуле

$$\chi^2_N = \frac{(T1 - П1)^2}{T1} + \frac{(T2 - П2)^2}{T2} + \frac{(T3 - П3)^2}{T3}.$$

Таблица 2

Число бросаний (N)	Значение χ^2_N
100	0,96
500	0,076
1000	2,144

Найдем степень свободы $m = r - 1$, где r – количество разбиений множества возможных значений случайной величины (в нашем примере – число возможных исходов опыта). В нашем случае $m = 2$. Соответствующая строка в таблице распределения хи-квадрат выглядит следующим образом (табл. 3).

Все значения χ^2_N оказались больше 25%-й точки. То есть такие последовательности из «ор-

лов» и «решек» при бросании двух монет должны получиться более чем в 25% случаев. Обратим внимание на случай $N = 500$. Здесь вероятность появления последовательности лежит между 95 и 99%. То есть последовательность получилась «подозрительно» случайной. На практике последовательность с таким значением χ^2_N подвергают дополнительной проверке.

Доступное для учащихся изложение критерия Пирсона можно найти в учебнике [7]. Авторы, «обходя стороной, дебри теории вероятностей», сначала подводят учащихся к мысли, что для проверки датчика случайных чисел необходимо моделировать с его помощью случайный процесс, и только затем переходят к рассмотрению критерия хи-квадрата.

Отметим, что на практике число испытаний N и число разбиений r выбирают так, чтобы $N p_j \geq 5$. Это необходимо для «устойчивости» значений v_j .

Составим *алгоритм* критерия хи-квадрат для проверки гипотезы о равномерном распределении на полуинтервале $[a, b)$ непрерывной случайной величины, заданной последовательностью n значений. (То есть проверяем, что случайная величина задается интегральной функцией распределения $F(x) = x$, при $a \leq x < b$.)

Шаг 1. Получить массив X из n чисел и отсортировать его по-возрастанию.

Шаг 2. Присвоить значения переменным a, b, n, r, p , где r – число разбиений интервала $[a, b)$, p – вероятность ($p = (b - a) / r$).

Шаг 3. Выполнить группировку значений для массива X , результат записать в массив Y . (Разбить полуинтервал $[a, b)$ на r равных частей и подсчитать, сколько чисел данной последовательности попадает в каждую часть.)

Шаг 4. Вычислить значение V по формуле: $V := V + (Y[j] - n * p)^2 / n * p, j = 1...k$.

Шаг 5. Интерпретировать результат, сравнив с таблицей χ^2 -распределения при $m = r - 1$. Если V меньше 1%-й точки или больше 99%-й точки, отбрасываем рассматриваемую последовательность как недостаточно случайную. Если V лежит между 1%- и 5%-й точками или между 95%-й и 99%-й точками, то последовательность «подозрительна».

Описанный алгоритм подходит и для дискретной случайной величины, принимающей значения из полуинтервала $[a, b)$ с равными вероятностями $p = (b - a) / r$.

Проверка по χ^2 -критерию часто производится три раза (и более) с разными данными. Если по

Таблица 3

Вероятность появления последовательности	$p = 1\%$	$p = 5\%$	$p = 25\%$	$p = 50\%$	$p = 75\%$	$p = 95\%$	$p = 99\%$
$m = 2$	9,21	5,991	2,773	1,386	0,5753	0,1026	0,00201

Число испытаний	$n = 10$			$n = 100$		
	1	2	3	1	2	3
Запуск генератора						
Pascal	$V = 14$	$V = 8$	$V = 12$	$V = 5.6$	$V = 6.1$	$V = 6.4$
Delphi	$V = 14$	$V = 14$	$V = 7$	$V = 5.7$	$V = 8.8$	$V = 9.6$
Excel	$V = 12$	$V = 6$	$V = 14$	$V = 8.6$	$V = 9.8$	$V = 9.5$

Таблица 5

Вероятность появления последовательности	$p = 1\%$	$p = 5\%$	$p = 25\%$	$p = 50\%$	$p = 75\%$	$p = 95\%$	$p = 99\%$
$m = 9$	2,088	3,325	5,899	8,343	11,39	16,92	21,67

крайней мере два из трех результатов оказываются «подозрительными», то гипотеза о законе распределения случайной величины отвергается.

Следует отметить, что критерий хи-квадрата не является достаточным условием для проверки гипотезы о распределении случайной величины, поэтому на практике применяется несколько эмпирических критериев. В настоящее время наиболее мощным из известных критериев для проверки качества линейных конгруэнтных генераторов псевдослучайных чисел является *спектральный критерий*, он обладает свойствами как теоретических, так и эмпирических критериев.

Рассмотрим еще один *пример*. Применим критерий χ^2 для анализа встроенных генераторов псевдослучайных чисел в языках программирования Pascal и Delphi, а также в табличном процессоре Excel. Будем генерировать целые числа от 0 до 9 включительно. Результаты занесены в табл. 4.

В таблице χ^2 -распределения при $m = 9$ получаем следующие значения (табл. 5).

При $n = 10$ ни один из генераторов проверку не прошел. Это объясняется тем, что выбранное n было слишком мало (теоретические частоты $n \cdot p = 1 < 5$). При $n = 100$ все генераторы прошли проверку (ответ был предсказуем), причем генератор в Excel показал результат, близкий к 50%-й точке, такая случайная последовательность наиболее приемлема на практике. Отметим, что в рассмотренных программных продуктах используются разные способы генерирования псевдослучайных чисел. В Pascal и Delphi используется генератор псевдослучайных чисел с циклом, равным 2^{32} , начальное число в последовательности берется с учетом текущего времени системных часов (при условии использования функции randomize). В Excel используется линейный конгруэнтный метод генерации псевдослучайных чисел.

В заключение еще раз подчеркнем значимость в школьном курсе понятия «псевдослучайная последовательность». Прежде всего, это связано со все больше востребованным моделированием на компьютере случайных процессов. Введение этого понятия на непрофильном уровне, на наш взгляд, нецелесообразно, но необходимо грамотное формирование у учащихся представления о псевдослучайных числах, знания пределов использования получаемых с помощью программы псевдослучайных последовательностей. Такой подход будет способствовать соблюдению педагогического принципа научности содержания, кроме того, с учетом широкого распространения стохастических методов вполне вероятно, что эти знания пригодятся учащимся в будущем.

Примечания

1. Сборник нормативных документов. Информатика и ИКТ [Текст]. М.: Дрофа, 2007. 103 с.
2. Бунимович, Е. А. Вероятность и статистика. 5–9 кл. [Текст] : пособие для общеобраз. учеб. заведений / Е. А. Бунимович, В. А. Булычев. М.: Дрофа, 2005. 159 с.
3. Здесь слово «случайный» использовано скорее не в математическом, а в бытовом смысле. С точки зрения математики закон распределения случайной величины – это абстрактное понятие; и только опыт может убедить нас в том, что какая-либо конкретная последовательность чисел обладает интересующим нас свойством независимых случайных чисел. *Соболь, И. М. Численные методы Монте-Карло [Текст] / И. М. Соболь. М.: Наука, 1973. 311 с.*
4. *Сенокосов, А. И. Информатика [Текст] : учеб. для 8–9 кл. шк. с углуб. изуч. информатики / А. И. Сенокосов, А. Г. Гейн. М.: Просвещение, 1995. 255 с.*
5. [Электронный ресурс]. <http://msdn2.microsoft.com>.
6. *Кнут, Д. Искусство программирования. Т. 2. Получисленные алгоритмы [Текст] / Д. Кнут. М.: Вильямс, 2003. 832 с.*
7. *Сенокосов, А. И. Указ. соч.*

И. А. Корякина

ИННОВАЦИОННЫЙ ПОДХОД В ОБУЧЕНИИ ИТ-СПЕЦИАЛИСТОВ В ВЫСШЕЙ ШКОЛЕ

В статье рассматриваются новые подходы к подготовке ИТ-специалистов в вузе.

Логика развития современного общества (в условиях принципиальной ограниченности сырьевых ресурсов) требует повышения производительности труда даже в тех сферах, которые традиционно считались «неподдающимися» механизации и автоматизации. Реорганизация бизнес-процессов (по сути – экономических процессов) является наиболее доступным и очевидным резервом повышения производительности труда. Всплеск интереса к этому быстрому в освоении и эффективному по отдаче резерву привёл к дефициту ИТ-специалистов во всех отраслях экономики. Это объясняется опережающими (точнее – описываемыми законом Мура) темпами (по сравнению с темпами подготовки и переподготовки кадров) увеличения доступности вычислительных ресурсов, что приводит к прогрессирующей нехватке специалистов, умеющих осваивать всё возрастающие вычислительные мощности. С учётом уровня поступления выпускников школ в вузы, можно сделать вывод, что экстенсивный способ решения задачи подготовки ИТ-кадров исчерпан, и следует признать, что теперь нехватка вызвана качеством вузовской подготовки.

Методологические проблемы, возникающие в ходе решения задачи подготовки ИТ-специалистов. Феномен ИТ-технологий состоит в том, что разрыв во времени между исследовательскими проектами, лабораторными результатами и массовым серийным производством достиг небывало низких значений. В области производства программного обеспечения он фактически достиг нуля: свежеразработанный продукт может выставляться на продажу через Интернет и становиться массовым буквально за сутки. Это привело к тому, что старая парадигма университетского образования, ориентированная на воспроизводство академической элиты (и выпускавшая специалистов-практиков в качестве побочного продукта), требует некоторого переосмысления в направлении интеграции преподавания знаний и формирования деятельностных компетенций. Как отмечено в работе [1], «специфические требования к современному выпускнику обусловлены тем, что его профессиональная деятельность

будет проходить в условиях нестабильного экономического окружения. Поэтому ему необходимо уметь рационально мыслить, самостоятельно пополнять свои знания и навыки, владеть культурой рационального управления». Из приведённой цитаты следует, что главной компетенцией на ближайшую перспективу является скорость адаптации к изменению экономических условий, что требует изучения студентами дисциплин экономико-управленческого направления. Вместе с тем методика преподавания должна быть нацелена на преподавание знаний, умений, навыков (ЗУН) таким образом, чтобы одновременно сформировать у студента помимо практических ЗУН выбранной предметной области и умение быстрого освоения возникающих в предметной области новшеств. То есть у студента должны сформироваться некие надпредметные ЗУН (как бы метаЗУН) по быстрому самостоятельному освоению предметных ЗУН.

Подходы к решению методологических проблем подготовки ИТ-специалистов. В высшей школе в настоящее время наметились определенные направления изменений, связанные с разработкой и внедрением новых образовательных стандартов по подготовке ИТ-специалистов и выработкой единого понимания содержания образования по информатике: акцент делается на самостоятельную форму обучения студентов, при этом она переходит на качественно новый уровень. В основу данной формы обучения положена компетентностная модель специалиста, менее жестко привязанная к конкретному объекту и предмету труда. В данном случае под компетентностной моделью специалиста достаточно понимать рассмотрение специалиста как обладателя набора компетенций.

Поэтому включение экономических дисциплин в учебные планы технических специальностей вузов, во-первых, следует рассматривать как повышение компетентности выпускников, что актуально для современного рынка труда:

1) даёт специализированную компетентность как специалистов по активному внедрению ИТ-технологий в таких непромышленных отраслях экономики, как финансовая, консалтинговая и др.;

2) повышает специальную компетентность ИТ-специалистов как специалистов по приложению ИТ-технологий к вопросам экономического планирования/управления в любых отраслях экономики;

3) повышает общую компетентность специалистов как потенциальных кандидатов на руководящую должность.

Во-вторых, смысл преподавания экономики ИТ-специалистам ещё и в следующем:

1) экономисты плохо знают инженерию и поэтому методологически плохи как экономисты;

КОРЯКИНА Ирина Аркадьевна – старший преподаватель кафедры информатики ВятГГУ, начальник сектора обработки документов филиала в Кирове «ОАО ВТБ Северо-запад»

© Корякина И. А., 2008

2) в процессе до-обучения легче инженеру освоить экономику, чем экономисту освоить инженерию;

3) подготовленные таким образом инженеры, до-обученные до экономистов, заменят экономистов, не справившихся со своими до-обучением и работой.

В качестве главных направлений перехода к новой системе образования в настоящее время рассматривают также:

- внедрение методов инновационного и развивающего образования на основе использования перспективных информационных и Интернет-технологий;

- повышение доступности качественного образования путем развития системы дистанционного обучения и средств поддержки учебного процесса современными информационными и телекоммуникационными технологиями.

Инновационный подход к методике обучения ИТ-специалистов. Предлагаемый нами инновационный подход к методике обучения ИТ-специалистов охватывает все перечисленные тенденции в педагогике и позволит обеспечить эффективность решения вопросов соответствия квалификации выпускника требованиям к нему, предъявляемым современным обществом.

В основе инновационного подхода лежит ряд положений. Рассмотрим их.

1) Ряд дисциплин подготовки интегрируются в единое целое на основе неких системообразующих идей, влияющих на отбор и логику преподавания содержания.

Технологические подходы к обучению. Содержание обучения охватывает не только учебный материал, подлежащий усвоению, но и то, что и на каком уровне, каким образом, усвоено, т. е. включает в себя технологии и формы организации учебной деятельности, методы обучения и диагностику его результатов – в той мере, в которой все это влияет на обученность, образованность, развитие личности.

В современном смысле под технологией обучения понимают всю совокупность проблем, связанных с целями, содержанием, организацией и проведением учебного процесса, принципы и приемы его оптимизации и управляемости, повышения эффективности, систему методических, дидактических, психологических и педагогических процедур, осуществляющих требуемые изменения в деятельности обучаемых; особые комбинации методов и приемов обучения, гарантирующих достижение планируемых результатов; системный метод создания, применения и определения всего процесса преподавания и усвоения знаний с учетом технических и человеческих ресурсов и их взаимодействия, ставящий своей задачей оптимизацию форм образования [2].

Современным общеобразовательным технологиям посвящены многочисленные монографии. Опыт использования технологий образования накоплен в высшей технической школе. Но все они базируются на понимании под технологией обучения использования технических средств обучения (ТСО), применения компьютеров в учебном процессе и контроля как системообразующего фактора технологии на уровне учебного занятия и учебной дисциплины. Дальнейшее развитие исследований в области педагогической технологии несколько расширило ее содержание. Под педагогической технологией стали понимать не просто исследования в сфере использования технических средств или компьютеров, а исследования с целью выявления принципов и разработки приемов оптимизации образовательного процесса путем анализа факторов, повышающих образовательную эффективность путем конструирования и применения приемов и материалов, а также посредством оценки применяемых методов.

В условиях того, что руководство Вятского государственного гуманитарного университета идет навстречу в реализации нашей инновационной методики, наша идея состоит в попытке раздвинуть рамки других упомянутых ограничений и применить технологический подход к обучению ИТ-специалистов в плане использования на практических занятиях современных инструментов обучения. Современные инструменты обучения – это реальные информационные системы по дисциплинам «Информационные системы в банковском деле» и «Информационные системы в страховом деле», обеспеченные технической поддержкой и обновлением версий системы со стороны фирм-разработчиков. В результате учащиеся будут в курсе всех последних нормативных изменений в предметных областях, которые отражаются со стороны разработчика в новых версиях систем.

Обучение этим системам ведётся на основе методических разработок, которые учитывают реальные ситуации в предметных областях, а также требуют от обучаемых самостоятельной, творческой работы на основе полученных знаний на экономических дисциплинах «Банковское дело» и «Страховое дело». Именно в этом заключается реальная стыковка предыдущих этапов обучения с последующими, что также является особенностью нашего технологического подхода.

Инженерный подход. Один из актуальнейших ныне вопросов – методы и содержание подготовки ИТ-специалиста как инженера. Как перестроить сферу образования в этом направлении, чтобы она вобрала и соединила в себе цели и задачи инженерного и технического образования? Мы строим действительно новые средства решения этой проблемы и тем самым смотрим на

такое явление, как образование, как на «искусственное» и «техническое», то есть сотворенное не столько природой и историей, сколько нашими устремлениями.

«Искусственное» и «техническое» отношения к миру свойственны особой социальной и профессиональной группе – инженерам. Умение переносить способы работы из одной области деятельности в другую, из одного времени в другое – это одна из важнейших особенностей инженеров.

Образовательные учреждения следует рассматривать как производства. Вот в этом и заключается инновация: мы позволяем богатейшему опыту, накопленному в области организации промышленного производства, начать работать в области педагогики.

Мы получаем вывод о применении инженерного подхода к организации учебного процесса. Пока этот подход в начальном состоянии: на уровне автоматизации составления ведомостей и расписаний в деканате, подготовки компьютерных тестов для контроля знаний.

Второй инновационный момент состоит в том, что применением инженерного подхода можно раскрыть большой запас потенциальных возможностей, которые имеют когнитивные технологии обучения.

Этот подход может состоять в основном в применении к методике преподавания (так как программа обучения задана федеральными стандартами):

- 1) инженерной психологии;
- 2) онтологии с использованием ИИ (искусственного интеллекта);
- 3) проектного подхода.

2) *Прагматическая направленность подготовки – инновационные образовательные программы – реализация идеи технопарка.* В то время как на рынке ощущается серьезная нехватка ИТ-специалистов, в программах вузов по-прежнему не хватает учебного времени на преподавание предметов, направленных на получение практических знаний и навыков. Это не дает возможности вузам гибко менять программы обучения, отражая изменения в области ИТ, которые в силу высоких темпов развития отрасли происходят очень часто. В результате выпускников приходится доучивать в компании не менее полугода. В связи с этим актуальной становится реализация идеи технопарка на вузовском уровне. Взаимодействие фирмы-разработчика систем, производственного предприятия и вуза в течение учебного процесса – вот что необходимо реализовать в рамках этой идеи. И уже известны единичные случаи такого взаимодействия: В мае 2008 г. ОАО КБ «Центр-инвест», компания SAP СНГ и Южный

Федеральный Университет заключили соглашение о сотрудничестве в рамках SAP University Alliance Program («Университетский Альянс»). Реализация программы предполагает создание на базе ЮФУ нового Центра, который будет осуществлять выпуск специалистов в области решений SAP, и разработку учебных планов по курсам изучения продуктов SAP. Первый набор студентов планируется осуществить уже в этом году. Банк «Центр-инвест» стал первым банком, проявившим инициативу поддержки преподавания передовых информационных технологий SAP в ЮФО. «Центр-инвест» планирует не только участвовать в создании условий для получения теоретических знаний в области решений SAP, но и готов предложить будущим студентам стажировку в банке, а в дальнейшем – возможность стать одним из своих сотрудников [3].

Все предпосылки к подобному сотрудничеству есть и у ВятГГУ, это – использование в учебном процессе продуктов фирмы R-Style Softlab (в настоящее время достигнута договоренность о приобретении дополнительных подсистем к существующей в университете системе RS-bank v5.0.), наличие в г. Кирове нескольких банков, работающих на АБС этой же фирмы, опыт производственной практики студентов специальности «ПИ в экономике» в одном из банков города.

3) *Особенности изучения материала в рамках инновационного подхода.* Осмысление исторического опыта в области высшего профессионального образования (университеты->академгородки->научограды->технопарки, университеты->вузы->заводы-вузы->технопарки) показывает, что имеются долгосрочные тенденции (так называемые «длинные тренды») в области образования, а именно: сближение форм академического и производственного обучения по формуле университеты->...->технопарки. Исходя из этой тенденции, в статье предлагается инновационный подход к организации курса экономики для ИТ-специалистов. Инновация состоит в том, что изучаемый материал рассматривается как предметная область, подлежащая системному анализу. Таким образом, изучение экономических дисциплин становится не только академически-учебным процессом, но и познавательной деятельностью «в условиях, максимально приближенных к трудовым», так как наиболее востребованной профессиональной компетенцией ИТ-инженера является умение проводить организацию, формализацию и автоматизацию [бизнес/производственных] процессов того предприятия, на котором он трудится.

Специалист в области ИТ должен уметь использовать модели деятельности любого предприятия. Теория управления бизнес-процессами

уже давно используется в российской практике, на многих примерах доказала свою эффективность, и год от года число компаний, которые строят у себя систему процессного управления, увеличивается. Многие компании, пять-шесть лет назад пробовавшие внедрить процессный подход как дань моде, теперь четко осознали необходимость управления бизнес-процессами и не только описывают их, но и начинают совершенствование и автоматизацию бизнес-процессов. По результатам опроса клиентов компании IDS Scheer (www.ids-scheer.ru), 55% из них в настоящее время активно ведут работы по совершенствованию бизнес-процессов.

В России активность компаний в области внедрения процессного подхода зависит от их отраслевой принадлежности. Банковский и телекоммуникационный секторы уже активно используют элементы процессного управления, автоматизируя процессы с помощью ВРМ-систем.

В условиях необходимости совершенствования внутренней деятельности банки вынуждены применять современные технологии управления и автоматизации, а именно процессный подход. В то же время для телекоммуникационного рынка характерны активные слияния и поглощения при высоком уровне конкуренции на рынке, что заставляет телекоммуникационные компании не только совершенствовать свои внутренние бизнес-процессы, но и согласовывать их с внутренними процессами партнеров. Именно поэтому большинство телекоммуникационных компаний уже работают над описанием и автоматизацией своих бизнес-процессов.

Второй инновацией является практика применения преподавателем в организации учебного процесса системного анализа не только как собственно инструмента повышения эффективности своего труда, но и как педагогического приёма. То есть сам преподаватель оказывается вовлечённым в процесс системного анализа дважды:

- во-первых, он выступает в роли системного аналитика в отношении преподаваемого экономического материала;

- во-вторых, он выступает в роли системного аналитика в отношении своей собственной педагогической деятельности, организуя, формализуя и автоматизируя её с использованием тех же самых методологических подходов, которые студенты осваивают в применении к экономике.

С точки зрения педагогики это выглядит так:

- в первом случае студент под руководством преподавателя учится системному подходу к экономике, преподаватель в этом случае выступает как бы в роли руководителя/«технического лидера» группы специалистов/студентов, осваивающих экономические знания. Педагогический принцип – «делай, как я, делай вместе со мной»;

- во втором случае студент выступает в роли рядового участника учебного процесса (как бы клиента), преподаватель же выступает в роли системного аналитика некоторой (в данном случае – педагогической) предметной области. Педагогический принцип – «делай аналогично тому, как делаю я». Это соответствует более высокому уровню освоения материала.

Третьей инновацией является использование синергетического эффекта от 2 первых инноваций. Этот эффект состоит в том, что преподаватель как бы объединяет обе роли системного аналитика и начинает выступать в роли инженера-когнитивиста, формулирующего единый подход к описанию 2 предметных областей. Таким образом, студенты «бесшовно» вовлекаются в освоение компетенции онтологического инжиниринга [4].

Примечания

1. Рубцов, С. В. Целевое управление корпорациями. 2001 [Электронный ресурс] / С. В. Рубцов // http://or-rsv.narod.ru/Book/Book_0_1_1.htm#v1

2. Гайфуллин, В. Г. Технологические подходы в педагогике [Текст]: ученые записки / В. Г. Гайфуллин. Казань: Изд-во ТГГИ, 2000.

3. [Электронный ресурс]. <http://bankir.ru/news/newslne/18.06.2008/135239>

4. Там же.

И. А. Корякина, О. Н. Тупицына

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО ПРОДУКТА «1С:ПРЕДПРИЯТИЕ 8 УПРАВЛЕНИЕ СТРАХОВОЙ КОМПАНИЕЙ» В УЧЕБНОМ ПРОЦЕССЕ ВятГГУ

Статья посвящена обзору возможностей программного средства «1С:Предприятие 8 Управление страховой компанией», которое используется для проведения практических занятий в ВятГГУ на факультете информатики по специальности 010502.65 (351400) Прикладная информатика (в экономике) по курсу «Информационные системы в страховом деле».

В соответствии с учебным планом специальности 010502.65 (351400) Прикладная информатика (в экономике) курс «Информационные системы в страховом деле» предполагает практические занятия. В качестве методического программного средства для проведения практических занятий по курсу используется программный про-

КОРЯКИНА Ирина Аркадьевна – старший преподаватель кафедры информатики ВятГГУ, начальник сектора обработки документов филиала в Кирове «ОАО ВТБ Северо-запад»

ТУПИЦЫНА Оксана Николаевна – старший преподаватель кафедры информатики ВятГГУ
© Корякина И. А., Тупицына О. Н., 2008

дукт «1С:Предприятие 8 Управление страховой компанией», учебная версия которого была приобретена Вятским государственным гуманитарным университетом в октябре 2007 г.

Программный продукт «1С:Предприятие 8 Управление страховой компанией» – отраслевое тиражное решение на платформе «1С:Предприятие 8», предназначено для комплексной автоматизации управления бизнесом страховых компаний, позволяет автоматизировать основные участки управленческого и регламентированного учетов. Данный продукт является совместным решением фирмы «1С» и компании «ОРТИКОН».

Группа компаний «Ортикон» – официальный партнер фирмы 1С по внедрению программ системы «1С:Предприятие» образовалась в результате объединения компаний:

– «Ортикон», специализирующейся на разработке и внедрении отраслевых решений на основе «1С:Предприятие» для страховых, инвестиционных компаний и негосударственных пенсионных фондов, а также других организаций финансового рынка;

– консалтинговой группы СофтОфис, специализирующейся на разработке и внедрении нетиповых автоматизированных систем управления и учета для предприятий различного профиля на платформе «1С:Предприятие 8».

За прошедшее время компания «Ортикон» разработала и внедрила различные решения для таких компаний, как СК «РУКСО», СК «Фиделити», СК «Москва», СК «Ариадна», СК «Природа», СК «Информстрах», и многих других. Среди клиентов компании около 100 страховых компаний. Компания является признанным лидером среди разработчиков отраслевого программного обеспечения на основе 1С для страховых компаний. Компания «ОРТИКОН» сертифицирована по международным стандартам качества ISO 9001:2000 по внедренческой деятельности.

Цель практической части курса состоит в том, чтобы дать студентам практические знания:

1) в предметной области управления и ведения учета в страховой компании;

2) об общей структуре конфигурации системы «1С:Предприятие 8 Управление страховой компанией».

В ходе курса рассматриваются архитектура конфигурации «Управление страховой компанией», методики использования управленческого функционала, а также взаимодействие подсистем в части управления портфелем договоров страхования, сострахования, управление портфелем договоров перестрахования, управление урегулированием убытков, управления денежными средствами, регламентированного учета.

Функциональные возможности конфигурации «1С:Управление страховой компанией»:

1. Управление портфелем договоров прямого страхования, сострахования

Учет и сопровождение договоров страхования – основа работы системы.

Данный функционал системы позволит студентам:

1) ознакомиться с управлением договорами страхования, сострахования, что включает ведение полной информации по договору. При этом практически вся информация является периодической, т. е. может быть изменена и при этом сохранится история изменений;

2) наблюдать состояние каждого договора на каждую конкретную дату. Все изменения по договору вносятся с помощью специального документа – дополнительного соглашения. По договору возможно также досрочное прекращение, возобновление и пролонгация;

3) отследить все движения по бухгалтерскому, налоговому и управленческому учету, которые осуществляются автоматически (имеется возможность настроить разные варианты начислений). Но также есть возможность ручного ввода необходимых начислений;

4) научиться управлять отношениями с посредниками. Список посредников по договору не ограничен. Возможны варианты ведения расчетов как через посредников, так и напрямую с контрагентом (страхователем).

Для успешной работы в системе необходимо хорошее знание таких понятий из базового курса по страховому делу, как виды страхования, правила страхования, объекты страхования, страховой продукт, страховой риск.

2. Управление продажами страховых продуктов

Страховой продукт может быть произвольным – т. е. пользователь просто настраивает сочетание типов объектов и рисков. Кроме этого в системе можно выбрать predetermined тип для страхового продукта, например ОСАГО.

Данный функционал системы позволит студентам освоить следующие этапы продажи страхового продукта:

1) сопровождение договоров. На основе первичных документов формируется электронный документооборот, при этом каждая операция, влияющая на реквизиты договора, сохраняется в системе;

2) многосекционный учет. Учет неограниченного количества видов, правил страхования в рамках одного договора обуславливается тем, что все суммовые показатели (финансового учета) указываются в разрезе каждого объекта и риска;

3) построение простых и сложных (периодических) графиков платежей.

Система предоставляет возможность указания графика платежей по договору страхования. Для облегчения ввода графика существует специальный механизм «Помощник заполнения графика платежей». С его помощью пользователь может создать график любой сложности за несколько секунд;

4) управление взаиморасчетами по комиссионному вознаграждению с посредниками. В конфигурации имеется возможность управлять отношениями с посредниками. Список посредников по договору не ограничен. Возможны варианты ведения расчетов как через посредников, так и напрямую с контрагентом (страхователем);

5) учет бланков строгой отчетности. Конфигурация предоставляет возможность управления бланками строгой отчетности, при этом возможно вести номерной учет бланков. Далее можно осуществить перемещение бланков между материальными лицами и подразделениями, выдать бланки агенту и получить обратно неиспользованные бланки от агента.

3. Управление урегулированием убытков

При этом студенты знакомятся со следующими основными функциональными возможностями подсистемы:

1) ведение извещений об убытке;

2) учет заявлений об убытках. По одному заявлению могут быть учтены сразу несколько договоров, относящихся к одному страховому событию;

3) убытки детализируются по получателям последующей выплаты;

4) для вида страхования «ОСАГО» убытки классифицируются по разрезам, необходимым для специализированной отчетности;

5) страховой акт может быть составлен сразу по нескольким заявлениям об убытках;

6) учет ДТП по виду страхования «ОСАГО»;

7) управление регрессными исками.

4. Управление финансами

Подсистема Управление финансами обеспечивает ведение учета в соответствии с российским законодательством по всем участкам учета, а также ведение внутреннего оперативного учета с большой детализацией данных.

Функциональными особенностями данной подсистемы являются:

1) обеспечение многофирменного (и/или многофилиального) учета в рамках одной информационной системы (и/или базы данных);

2) план счетов бухгалтерского и налогового учета страховой организации;

3) отражение операций по бухгалтерскому, налоговому и управленческому учету;

4) разделение вариантов начисления премий и комиссии для целей бухгалтерского и налогового учета;

5) детализация совершенных операций по управленческому учету для составления внутренней оперативной отчетности.

Функциональные возможности подсистемы могут использоваться финансовым директором, сотрудниками бухгалтерии и планово-экономического отдела, а также другими финансовыми службами предприятия.

5. Управление денежными средствами

В конфигурации реализовано управление движением наличных и безналичных денежных средств. Поддерживается ввод и печать платежных поручений, приходных и расходных кассовых ордеров.

6. Бухгалтерский учет

Возможности ведения бухгалтерского учета, реализованные в системе, призваны обеспечить полное соответствие как российскому законодательству, так и потребностям реального бизнеса. Организация подсистемы бухгалтерского учета обеспечивает высокую степень автоматизации формирования бухгалтерской отчетности.

Поддерживается ведение бухгалтерского учета для нескольких юридических лиц в единой информационной базе. Такая организация учета позволяет продемонстрировать студентам автоматизацию предприятия с достаточно сложной организационной структурой.

7. Налоговый учет

Налоговый учет по налогу на прибыль в конфигурации ведется независимо от бухгалтерского учета. Хозяйственные операции отражаются параллельно в бухгалтерском и налоговом учете.

8. Формирование регламентированной и аналитической отчетности

В конфигурации имеется развитая система отчетности, которая включает как универсальные настраиваемые под любые запросы пользователя отчеты, так и регламентированные (журнал договоров, журнал убытков). Возможности системы позволяют познакомить студентов с полным комплектом бухгалтерских отчетов.

Таким образом, рассмотренные возможности системы «1С:Управление страховой компанией» характеризуют её с точки зрения полноты охвата бизнес-процессов страховой компании, а также характеристики, как функционально-модульная структура конфигурации, масштабируемость, доступная цена позволяют сделать вывод о грамотном подходе к достижению практических целей курса «Информационные системы в страховом деле» с помощью данной системы и подготовить достаточно компетентных специалистов в этой предметной области.

Е. В. Котельников, Т. А. Стародубова

РАСПАРАЛЛЕЛИВАНИЕ МЕТОДОВ ОБУЧЕНИЯ В ЗАДАЧАХ МНОГОКЛАССОВОЙ КЛАССИФИКАЦИИ

В статье рассмотрены стратегии многоклассовой классификации на основе метода опорных векторов (SVM). Предложены параллельные реализации указанных стратегий с использованием технологии интерфейса передачи сообщений (MPI). Приведены результаты экспериментов по сравнению эффективности последовательных и параллельных реализаций.

Метод опорных векторов

В последнее время остается актуальной задача классификации данных. В общем случае эта задача заключается в нахождении правила, которое относит некоторый объект к одному из нескольких заранее известных классов по определенным признакам. К задачам классификации относятся проблемы распознавания символов, голоса, лиц, тематическая категоризация текстов, анализ последовательностей событий в компьютерных сетях для диагностики неисправностей и др.

Одним из наиболее распространенных подходов, применяемых для решения задач классификации, является метод опорных векторов (SVM – Support Vector Machines). Теоретический фундамент метода был заложен Владимиром Наумовичем Вапником и Андреем Яковлевичем Червоненкисом в начале 70-х гг. [1], а прикладные аспекты стали исследоваться с начала 90-х гг. [2] В течение последних пятнадцати лет метод SVM получил широкое распространение, найдя применение в целом ряде задач классификации и распознавания образов, потеснив традиционные нейросетевые технологии.

Изначально метод опорных векторов был разработан для решения задач распознавания с двумя классами. Применение SVM сводится к двум основным этапам: обучению и классификации (или распознаванию). Обучение происходит на основе набора векторов (обучающей выборки). Каждый вектор представляет собой набор признаков, характеризующих конкретный объект или ситуацию предметной области. Например, в задаче тематической категоризации текстов элементами векторов будут веса слов, входящих в определенный текст. Кроме того, каждому вектору

сопоставлено число, обозначающее класс, к которому относится вектор.

В процессе обучения в пространстве признаков строится линейная или нелинейная гиперплоскость, разделяющая векторы разных классов, причем размерность пространства может быть очень большой (до сотен тысяч в текстовой классификации). При этом SVM выделяет так называемые *опорные векторы* – такие векторы, которые находятся ближе всего к разделяющей гиперплоскости. Только опорные векторы несут всю информацию о разделении классов, так что остальные векторы могут в дальнейшем не учитываться.

Обучение SVM сводится к решению задачи квадратичной оптимизации с предельными ограничивающими условиями и одним ограничением линейного равенства. Существуют различные методы решения данной задачи, но наиболее часто используется стратегия декомпозиции, когда исходную обучающую выборку разбивают на неактивную и активную части. Главным преимуществом данной стратегии является то, что она предлагает алгоритмы с требованиями памяти, которые линейны относительно количества обучающих примеров и количества опорных векторов. Примером такого подхода является алгоритм последовательной минимальной оптимизации (Sequential Minimal Optimization, SMO), предложенный Дж. Платтом [3], в котором размер активной части всегда равен двум, а для поиска оптимальной активной части используется набор эвристик.

На этапе классификации решение об отнесении распознаваемого вектора к тому или иному классу выносятся на основании того, как расположен этот вектор относительно разделяющей гиперплоскости.

Метод SVM можно обобщить и на задачи классификации с числом классов, большим двух. Соответствующие алгоритмы будут рассмотрены ниже.

Метод опорных векторов обладает высокой точностью распознавания, однако время обучения на реальных наборах данных весьма существенно и увеличивается пропорционально квадрату числа обучающих векторов. Ситуация усугубляется при большом числе классов.

Решить проблему недостаточной скорости обучения SVM возможно при использовании многопроцессорных вычислительных систем.

Многоклассовая классификация

Существуют два основных подхода к решению проблемы многоклассовой классификации в SVM. Первый называется «*решение за один шаг*» или «*все вместе*» (*all-together*) [4]. В этом

КОТЕЛЬНИКОВ Евгений Вячеславович – кандидат технических наук, зав. кафедрой информатики и методики обучения информатике ВятГГУ
СТАРОДУБОВА Татьяна Анатольевна – аспирант кафедры информатики и методики обучения информатике ВятГГУ

© Котельников Е. В., Стародубова Т. А., 2008

подходе задача квадратичной оптимизации усложняется за счет увеличения размерности дополнительных переменных. При этом алгоритмы декомпозиции (например, SMO), быстро решающие задачу обучения в случае бинарной классификации, здесь становятся неэффективными.

В другом подходе задача многоклассовой классификации сводится к последовательности задач с двумя классами. Как правило, оказывается, что обучить и скомбинировать несколько простых распознавателей намного быстрее, чем один сложный. При этом наиболее эффективны три стратегии конструирования и обучения простых распознавателей: 1) «один против всех» («one-against-all»); 2) «каждый против каждого» («one-against-one»); 3) «турнир на выбывание», или «ориентированный ациклический граф SVM» (Directed Acyclic Graph SVM, DAGSVM).

В стратегии «один против всех» [5] для N классов обучается N классификаторов, каждый из которых отделяет «свой» класс от всех остальных классов. На этапе распознавания неизвестный вектор X подается на все N классификаторов. Принадлежность вектора X определяется тем классификатором, который выдал наибольшую оценку $f(X)$.

Стратегия «каждый против каждого» [6] выделяет $N(N-1)/2$ классификаторов, обучающихся отличать все возможные пары классов друг от друга. Для распознаваемого вектора каждый классификатор выдает оценку $f_{ij}(X)$, отражающую принадлежность к классам i и j . Результатом является класс с максимальной суммой

$$\sum_{i \neq j} g(f_{ij}(X)),$$

где g – монотонно неубывающая функция, например тождественная или логистическая.

Стратегия «турнир на выбывание» [7] также предполагает обучение $N(N-1)/2$ классификаторов, различающих все возможные пары классов. Отличие от предыдущей стратегии заключается в том, что на этапе классификации вектора X между классами устраивается турнир. При этом на каждом шаге распознавание вектора X осуществляет только один классификатор – «победивший» класс продолжает борьбу и определяет следующий используемый классификатор. Процесс осуществляется до тех пор, пока не останется один победивший класс, который и будет считаться результатом распознавания. Классификация вектора X будет выполнена за $N-1$ шаг. По сравнению с двумя предыдущими, данная стратегия позволяет ускорить как обучение классификаторов, так и процесс распознавания, при равной, а иногда и превосходящей точности вычислений. Но к особенностям данной стратегии

можно отнести то, что если вектор X относится сразу к нескольким классам (например, в задаче тематической категоризации текстов один и тот же текст может относиться к нескольким рубрикам), то в результате будет найден только тот класс, признаками которого он обладает в наибольшей степени.

Сравнение различных методов обоих подходов многоклассовой классификации приведено в [8]. В целом можно сделать следующие выводы: методы сведения многоклассовой классификации к бинарной обучаются быстрее и дают меньшее число ошибок, в то время как в подходе «решение за один шаг» получается меньшее число опорных векторов.

Так как целью нашей работы является получение максимального быстродействия, для исследования возможностей распараллеливания были выбраны методы второго подхода – «один против всех», «каждый против каждого» и «турнир на выбывание».

Параллельная реализация

Классификации больших объемов данных со значительным количеством атрибутов по нескольким классам оказывается достаточно ресурсоемкой задачей. Использование современных многопроцессорных вычислительных систем, а также применение методов и средств параллельного программирования для организации вычислений позволяет получать результат требуемой точности за приемлемое время.

Идея параллельной реализации стратегии «каждый против всех» заключается в том, что среди процессов (их должно быть не больше количества классов N) распределяются классы. Каждый узел определяет правило классификации для каждого из классов, которые были разосланы ему корневым процессом программы. В результате класс получает правило, отличающее его от остальных. По окончании параллельных вычислений узловой процесс проверяет наличие ошибки в правилах классификации.

В случае стратегий «каждый против каждого» и «турнир на выбывание» используются уже не более $N(N-1)/2$ процессов. Процессам передаются пары классов. Их задача – найти правило, которое различает эти два класса между собой. Затем эти правила в качестве ответа возвращаются в корневой процесс. При использовании стратегии «каждый против каждого» классификация происходит следующим образом: для каждого классифицируемого объекта находят значения всех $N(N-1)/2$ функции. В турнирной же стратегии на каждом шаге выбывает один из классов, поэтому определяются значения только $N-1$ функции.

Результаты экспериментов

На основе предложенного подхода к распараллеливанию была реализована тестирующая программа. Она позволяет вводить обучающий набор данных, классифицировать данные с использованием алгоритма SMO, замерять временные затраты.

Реализация параллельной программы осуществлялась на языке программирования C# с применением многопоточности и функций библиотеки передачи сообщений MPI.NET [9].

Для экспериментального исследования использовался набор данных, представляющий рукописные цифры (Pen-Based Recognition of Handwritten Digits), полученный из репозитория UCI Machine Learning Repository [10]. Набор обучающих данных содержит 7494 вектора, размерность каждого вектора равна 16, количество классов – 10.

При обучении SVM применялось RBF-ядро с параметром $\sigma = 0,3$, управляющая константа $C = 4$, точность вычислений менялась от 0,1 до 0,001.

Экспериментальные расчеты для кластерной архитектуры проводились на вычислительном кластере Вятского государственного гуманитарного университета, состоящем из 12 вычислительных узлов. Каждый вычислительный узел представляет собой персональный компьютер с процессором AMD Athlon 64 2 ГГц и 512 Мб оперативной памяти. Узлы связаны сетью Fast Ethernet. Кластер функционирует на базе операционной системы Microsoft Compute Cluster Server 2003 SP1, на каждом узле установлена среда выполнения MPI.NET Runtime версии 0.9.0.

В ходе тестирования сравнивалась производительность кластера и отдельного компьютера (AMD Athlon 64 2 ГГц и 512 Мб ОЗУ). Были получены следующие результаты (см. таблицу).

Приведенная таблица позволяет сделать следующие выводы. Во-первых, произведенное распараллеливание задачи обучения многоклассовой классификации оказалось достаточно результативным и позволило достичь ускорения в среднем 9,4, а эффективности – 0,78 (при 12 процессорах).

Во-вторых, время обучения в методе «один против всех» при точности 0,1 в 5 раз превышает время обучения в методах «каждый против каждого» и «турнир на выбывание», а при увеличении точности эта разница становится десятикратной. Результаты экспериментов позволяют рекомендовать последние два метода к использованию на больших и сверхбольших наборах данных.

Эксперименты с многопоточным приложением проводились на двух персональных компьютерах: первый – одноядерный процессор Intel Pentium M, 1,6 ГГц, 512 Мб ОЗУ; второй – двухъядерный процессор AMD Athlon 64 X2 Dual, 2,29 ГГц, 2 Гб ОЗУ.

Использовался тот же набор данных (Pen-Based Recognition of Handwritten Digits) и такие же параметры ядра. Поскольку ставилась задача тестирования эффективности применения многопоточного программирования для обучения, то тестировался только метод турнирного отбора при точности вычислений 0,001. Результаты приведены ниже.

Число потоков	Время обучения на одноядерном компьютере, с	Время обучения на двухъядерном компьютере, с
1	798	583
2	744	310
3	749	302
4	750	294
5	751	306

Точность	Время обучения на одном компьютере	Время обучения на кластере	Ускорение	Эффективность
Метод «один против всех»				
0,1	1430	161	8,88	0,74
0,01	3485	427	8,16	0,68
0,001	5350	627	8,53	0,71
В среднем			8,53	0,71
Метод «каждый против каждого»				
0,1	305	32	9,53	0,79
0,01	432	44	9,82	0,82
0,001	531	55	9,65	0,80
В среднем			9,67	0,81
Метод «турнир на выбывание»				
0,1	303	31	9,77	0,81
0,01	429	41	10,46	0,87
0,001	530	54	9,81	0,82
В среднем			10,02	0,83

Из таблицы видно, что на двухъядерном процессоре применение двух потоков вместо одного дает ускорение в 1,88 раз. Даже на одноядерном процессоре использование двух потоков позволяет уменьшить время обучения на 7%. Однако если на двухъядерном процессоре число потоков больше 4, а на одноядерном – больше двух, затраты процессорного времени на создание потоков превышают эффект от распараллеливания.

Заключение

В статье были предложены параллельные реализации алгоритмов многоклассовой классификации. Как показал анализ результатов проведенного исследования, наиболее точный результат за наименьшее время из рассмотренных стратегий многоклассовой классификации дает «турнир на выбывание».

Особенность полученного алгоритма в том, что он является масштабируемым (число задействованных процессов – параметр программы). Но его масштабируемость ограничена и зависит от количества классов.

В дальнейшем предполагается продолжить разработку параллельных алгоритмов обучения SVM, провести изучение возможностей их применения на сверхбольших наборах данных, а также исследование параллельных реализаций методов подбора оптимальных параметров SVM.

Примечания

1. *Вапник, В. Н.* Теория распознавания образов [Текст] / В. Н. Вапник, А. Я. Червоненкис. М.: Наука, 1974.

2. *Boser, B. E.* A training algorithm for optimal margin classifiers [Electronic resource] / B. E. Boser, I. M. Guyon, V. N. Vapnik. In Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. ACM Press, 1992. P. 144–152.

3. *Platt, J.* Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines [Text] / J. Platt // Advances in Kernel Methods Support Vector Machine. Cambridge: MIT Press, 1999. P. 185–208.

4. *Vapnik, V.* Statistical learning theory [Text] / V. Vapnik. N. Y.: Wiley, 1998; *Weston, J.* Multi-class support vector machines. Technical Report CSD-TR-98-04 [Text] / J. Weston, C. Watkins; Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

5. *Bottou, L.* Comparison of classifier methods: a case study in handwriting digit recognition [Text] / L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. Le Cun, U. Muller, E. Sackinger, P. Simard, V. Vapnik; In International Conference on Pattern Recognition. IEEE Computer Press, 1994. P. 77–87.

6. *Krebel, B.* Pairwise classification and support vector machines [Text] / B. Krebel; in B. Schölkopf, C. J. C. Burges, A. J. Smola, editors. Advances in Kernel Methods – Support Vector Learning. Cambridge, MA, 1999. MIT Press. P. 255–268.

7. *Platt, J.* Large Margin DAGS for Multiclass Classification [Text] / J. Platt, N. Cristianini, J. Shawe-Taylor. In Advances in Neural Information Processing Systems, 12 ed. S. A. Solla, T. K. Leen and K.-R. Muller, MIT Press, 2000. P. 547–553.

8. *Weston, J.* Op. cit.; *Hsu, C.-W.* A comparison of methods for multi-class support vector machines [Text] / C.-W. Hsu, C.-J. Lin. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

9. Project MPL.NET [Electronic resource] // The Open Systems Lab, Indiana University. Режим доступа: <http://www.osl.iu.edu/research/mpi.net>.

10. *Asuncion, A.* UCI Machine Learning Repository [Text] / A. Asuncion, D. J. Newman (<http://www.ics.uci.edu/~mllearn/MLRepository.html>). Irvine, CA: University of California, School of Information and Computer Science, 2007.

А. В. Котельникова

КЛАСТЕР РАБОЧИХ СТАНЦИЙ В ВУЗЕ НА ОСНОВЕ MICROSOFT COMPUTE CLUSTER SERVER 2003

В статье описана архитектура кластера рабочих станций, представлен обзор особенностей специализированной операционной системы для управления кластером Microsoft Compute Cluster Server 2003, показаны основные варианты её установки.

Введение

Актуальные на сегодняшний день дисциплины, связанные с многопроцессорными ЭВМ, распределенными вычислениями, параллельной обработкой данных, изучаются студентами многих специальностей, имеющих отношение к информатике и компьютерной технике. Немаловажной составляющей таких дисциплин является разработка параллельных алгоритмов, для реализации которых требуется многопроцессорная техника, недоступная многим учебным заведениям по причине высокой стоимости. Поскольку многопроцессорные вычислительные системы необходимы для изучения лишь одной-двух дисциплин, а для остальных курсов достаточно и однопроцессорных машин, на их покупку редко выделяют средства, хотя эта техника может обеспечить условия для выполнения сложных расчетов в любой области науки, для развития научных школ, ведущих разработку фундаментальных и прикладных научных проблем в рамках международных, российских и региональных программ.

Вузы по-разному выходят из этой ситуации. Некоторые для минимизации материальных за-

КОТЕЛЬНИКОВА Анастасия Валерьевна – кандидат педагогических наук, старший преподаватель кафедры информатики и методики обучения информатике ВятГГУ

© Котельникова А. В., 2008

трат используют обычную однопроцессорную технику и на ней эмулируют многочисленные процессы с помощью потоков, сокетов и т. п. Это неплохой выход, но он не позволяет полностью ощутить выгоду от использования многопроцессорности.

Также можно использовать такой, требующий не слишком больших затрат, вариант, как кластер рабочих станций [1]. По сути – это просто объединение возможностей отдельных компьютеров, связанных коммуникационной сетью, и использование их как единого вычислительного узла.

Выделяют два подхода к созданию кластера. Первый подход применяется для мощных вычислительных систем. В этом случае процессоры компактно размещаются в специальных стойках, а для управления системой и запуском задач выделяется один или несколько полнофункциональных компьютеров, называемых хост-компьютерами. Второй подход используется в небольших кластерных системах. В кластер объединяются полнофункциональные компьютеры, которые продолжают работать и как самостоятельные единицы. Такой кластер можно рассматривать как вычислительную систему с распределенной памятью, распределенным управлением и т. д., он может обладать неплохими характеристиками по быстродействию, сравними с показателями специализированных суперкомпьютеров, а стоимость его относительно невелика, поскольку рабочие станции для кластера уже установлены, используются и не всегда загружены работой. Таким образом, можно, например, объединить вычислительные мощности компьютеров учебной аудитории, при этом они смогут функционировать и как отдельные компьютеры, и как единый кластер рабочих станций. Данный подход наилучшим образом подходит для вузов, поэтому он и рассматривается более подробно в этой статье.

Отдельные вычислительные узлы кластера могут обладать разными характеристиками, на них могут быть установлены различные операционные системы (гетерогенный кластер), но удобнее работать с однородным кластером, включающим одинаковые компьютеры с одной операционной системой. Поскольку наиболее распространенной операционной системой в нашей стране является Windows, многие пользователи имеют большой опыт работы в ней, то и на кластере можно установить операционную систему Windows – Windows Compute Cluster Server 2003 (CCS) [2]. Это одно из комплексных решений для параллельных вычислительных систем, интегрированная платформа для поддержки высокопроизводительных вычислений на кластерных системах.

Основные сведения о Windows Compute Cluster Server 2003

До выхода CCS пользователям Windows-кластеров приходилось одновременно использовать программное обеспечение нескольких производителей, а это могло быть причиной проблем с совместимостью различных программ друг с другом. После объявления о запуске Windows Compute Cluster Server 2003 9 июня 2006 г. [3] стало доступно новое кластерное решение, в котором компания Microsoft предоставляет полный спектр программного обеспечения, необходимый для эффективной эксплуатации кластера и разработки программ, в полной мере использующих имеющиеся вычислительные мощности. Оно позволяет эффективно эксплуатировать высокопроизводительные кластерные установки.

Системный кластер от Microsoft представляет собой надежную платформу для параллельных и высокопроизводительных вычислений. К основным преимуществам, отличающим кластерную систему Microsoft от других высокопроизводительных вычислительных систем, можно отнести следующие:

- упрощенная система развертывания и управления кластером;
- улучшенный процесс интеграции с существующей ИТ-инфраструктурой;
- поддержка широкого ряда приложений и знакомая для разработчиков ИТ-среда.

CCS состоит из операционной системы Microsoft Windows Server 2003 и Microsoft Compute Cluster Pack (CCP) – набора интерфейсов, утилит и инфраструктуры управления. Вместе с CCP поставляется SDK, содержащий необходимые инструменты разработки программ для CCS. Кроме того, с Microsoft Compute Cluster Server 2003 логически связан Microsoft Visual Studio 2005, являющийся интегрированной средой разработки параллельных программ, содержащий компилятор и отладчик программ, разработанных с использованием технологий MPI и OpenMP.

Корпорация Майкрософт на протяжении многих лет поддерживала и продолжает поддерживать отказоустойчивые кластерные системы с высоким уровнем эксплуатационной готовности [4]. Однако появление Windows Compute Cluster Server 2003 означало вывод высокопроизводительных вычислений на уровень индивидуальных пользователей и рабочих групп.

В списке самых высокопроизводительных суперкомпьютеров мира TOP500 [5] приводится 500 наиболее мощных известных компьютерных систем. Этот проект начался в 1993 г. и позволяет выявить основные направления развития высокопроизводительных вычислений. Таблица обновляется дважды в год: в июне и ноябре. В списке

TOP500 за ноябрь 2007 г. 406 из 500 приведенных в ней систем по архитектуре представляют собой кластеры, причем в 6 из них в качестве операционной системы используется Windows Compute Cluster Server 2003. Все эти шесть систем были разработаны в 2007 г. и занимают места: 116, 239, 309, 406, 407 и 422 из 500. Число процессоров в них от 1024 до 3000. Хотя места систем, использующих Windows Compute Cluster Server 2003, не очень высоки, все же стоит помнить, что, во-первых, CCS появился совсем недавно, лишь в июне 2006 г., и еще не в полную силу реализован его потенциал, а во-вторых, в таблице представлены наиболее производительные системы мира и даже 422-е место – хороший показатель.

В TOP50 [6] – списке наиболее производительных компьютерных систем России и стран СНГ, в 8-й редакции от 27.03.2008 г. также имеются системы, использующие операционную систему Windows Compute Cluster Server 2003. Одна из них занимает 21-е место и установлена в Нижегородском государственном университете им. Н. И. Лобачевского в 2007 г. [7] Данная система содержит 256 ядер. Другая занимает 32-е место, содержит 128 ядер и установлена в Центре наноструктурных материалов и нанотехнологий – центре коллективного пользования в Белгородском государственном университете в 2007 г.

Установка Microsoft Windows Compute Cluster Server 2003 – это не очень сложная операция, но, как и любая технология, требует предварительного планирования. Следовательно, важно понять основные параметры, в том числе архитектуру вычислительного кластера Windows, требования аппаратного и программного обеспечения, поддерживаемые сетевые топологии, особенности установки.

Требования к оборудованию и ПО

Целью использования CCS является обслуживание приложений, осуществляющих высокопроизводительные вычисления. Windows Compute Cluster Server 2003 поддерживает работу ряда популярных технологий, таких, как 64-процессорные вычислительные узлы; интерфейс передачи сообщений v2 (MPI2); коммуникационные сети – гигабитный Ethernet, Ethernet по удаленному и прямому доступу к памяти (RDMA), InfiniBand, Myrinet; компиляторы и библиотеки других фирм.

Минимальные требования к оборудованию [8] практически не отличаются от тех, которые предъявляются 64-разрядной версией ОС Windows Server 2003 Standard Edition. Максимальный размер ОЗУ для Windows Compute Cluster Server 2003 составляет 32 ГБ, минимальные требования –

512 МБ. Поддерживаются следующие процессоры: AMD Opteron, AMD Athlon 64; Intel Xeon и Intel Pentium с поддержкой технологии Intel Extended Memory 64 (EM64T); или другие совместимые процессоры. Что касается поддержки многопроцессорной обработки, Windows Compute Cluster Server 2003 и 64-разрядная версия Windows Server 2003 Standard Edition поддерживают до четырех процессоров на сервер, 64-разрядная версия Windows Server 2003 Enterprise Edition поддерживает до восьми процессоров на сервер. Необходимое свободное место на диске для установки – 4 ГБ.

Если используются службы удаленной установки (Remote Installation Services, RIS), то на ведущем узле должно быть два тома (C:\ и D:\) – один для системного раздела, а второй для служб RIS. Если ведущий сервер выполняет еще какие-то функции (например, выступает в роли файлового сервера или сервера печати), рекомендуется создавать дополнительные разделы. На вычислительных узлах требуется только системный раздел. Массивы независимых дисков с избыточностью (RAID) поддерживаются, но их использование не обязательно.

Для установки требуется по меньшей мере одна сетевая плата. Если используется частная сеть, на ведущем узле должно быть не менее двух сетевых плат (одна сетевая плата подключается к общей (корпоративной) сети, а другая – к выделенной частной сети кластера), а на вычислительных узлах – по крайней мере одна. Кроме того, в определенных случаях каждому узлу необходимо иметь высокоскоростную сетевую плату для сети MPI (Message Passing Interface).

На ведущем и вычислительных узлах Windows Compute Cluster Server 2003 должна быть установлена одна из следующих операционных систем: Windows Server 2003 Compute Cluster Edition, 64-разрядная версия Windows Server 2003 Standard Edition, 64-разрядная версия Windows Server 2003 Enterprise Edition, 64-разрядная версия Windows Server 2003 R2. Компоненты удаленного администрирования и планирования заданий автоматически устанавливаются на ведущем узле вычислительного кластера; чтобы упростить администрирование и планирование, их можно установить на удаленной рабочей станции. Эти компоненты поддерживают следующие операционные системы: Windows Server 2003 с пакетом обновления 1 (SP1) или Windows Server 2003 R2, Windows XP Professional с пакетом обновления 2 (SP2), 64-разрядная версия Windows XP Professional.

В целом Windows Compute Cluster Server 2003 позволяет создать экономически эффективное и мощное решение для высокопроизводительных вычислений, работающее на стандартных компь-

ютерах с архитектурой x64 и поддающееся быстрому расширению и развертыванию с помощью знакомых средств и технологий.

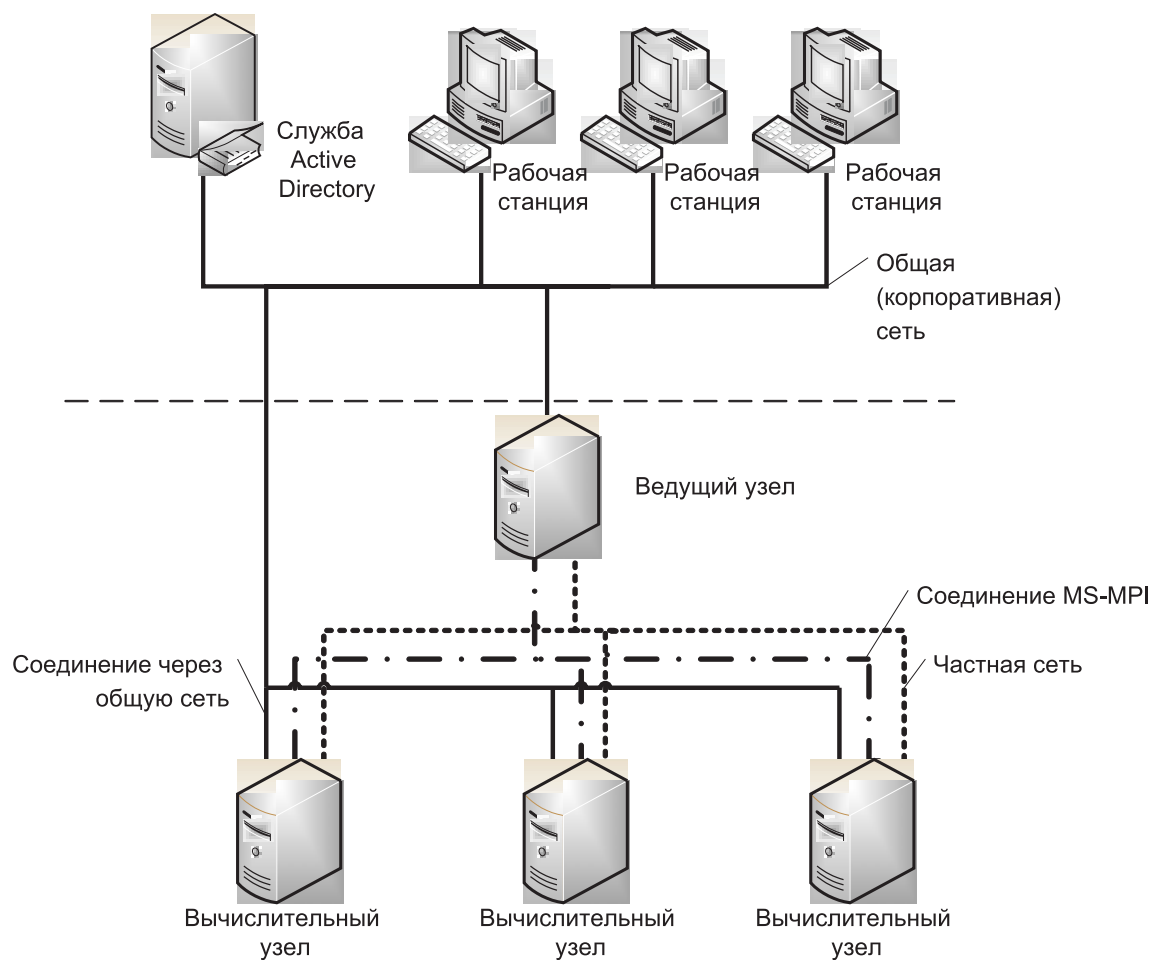
Архитектура и особенности установки Windows Compute Cluster Server 2003

Windows Compute Cluster Server 2003 – это кластер, включающий один ведущий узел и один или несколько вычислительных узлов [9]. Ведущий узел контролирует доступ ко всем ресурсам кластера, выступает посредником при получении этого доступа и является единственной точкой управления, развертывания и планирования заданий для вычислительного кластера. Windows Compute Cluster Server 2003 использует существующую корпоративную инфраструктуру Active Directory для обеспечения безопасности, управления учетными записями и общего управления операциями с помощью таких средств, как Microsoft Operations Manager 2005 и Microsoft Systems Management Server 2003.

На рисунке представлена общая схема архитектуры CCS. Далее рассмотрены компоненты Windows Compute Cluster Server 2003: Active Directory, ведущий узел, вычислительные узлы, планировщик заданий, Message Passing Interface, а также общие и частные сети.

– *Microsoft Active Directory* – служба каталога. Каждый узел кластера должен быть элементом домена Active Directory, предоставляющего службы авторизации и аутентификации для Windows Compute Cluster Server 2003. Домен Active Directory может быть независим от кластера или может работать в кластере на ведущем узле.

– *Ведущий узел* обеспечивает пользовательские интерфейсы развертывания и администрирования, а также службы управления вычислительным кластером. Основные интерфейсы: Compute Cluster Administrator (администратор вычислительного кластера), Compute Cluster Job Manager (планировщик заданий вычислительно-



Общая архитектура Windows Compute Cluster Server 2003

го кластера) и Command Line Interface (CLI, интерфейс командной строки). Службы управления отвечают за планирование заданий, а также за управление заданиями и ресурсами. Администратор имеет возможность при желании использовать для автоматического развертывания узлов службы удаленной установки (RIS). Если у вычислительных узлов нет интерфейса к общей сети (есть только частные интерфейсы), то для настройки Network Address Translation (NAT) между узлами и общей сетью используется Internet Connection Sharing (ICS). В этом случае ведущий узел выступает в роли шлюза между общей и частной сетью, составляющими кластер, обеспечивает работу служб Dynamic Host Configuration Protocol (DHCP) и Domain Name System (DNS) в частной сети. Администратор также имеет возможность назначать вычислительным узлам статические адреса.

– *Вычислительный узел* – это любой компьютер, настроенный на предоставление вычислительных ресурсов как часть вычислительного кластера. Здесь пользователи запускают вычислительные задания. На этих узлах должна быть установлена поддерживаемая операционная система, но не требуется, чтобы на всех была одна и та же ОС, и даже аппаратная конфигурация может различаться. Если на вычислительных узлах одинаковая конфигурация, то упрощается развертывание, администрирование и особенно управление ресурсами. Кластер, состоящий из различных конфигураций аппаратного обеспечения, будет ограничен в возможностях, потому что скорость выполнения заданий в параллельном режиме на разных узлах будет определяться быстродействием самого медленного процессора.

– *Планировщик заданий* компьютерного кластера выполняется на ведущем узле и управляет планированием заданий, выделением им необходимых ресурсов, изменением сопоставленных заданиям действий и свойств, а также выполнением всех задач благодаря взаимодействию со службой Node Manager Service, выполняющейся на каждом вычислительном узле. Планировщик заданий имеет интерфейс командной строки и поддерживает ряд языков, в том числе Perl, Fortran, C/C++, C# и Java. Задания могут состоять из одного или нескольких действий, а также содержать сведения о необходимом количестве процессоров и указание, должны ли эти процессоры предоставляться монополюбно или допустимо их совместное использование с другими заданиями/действиями.

Планировщик заданий имеет несколько важных особенностей.

1) *Восстановление после ошибок* – автоматическое повторение попытки выполнения неудавшихся действий и заданий, а также автоматичес-

кое игнорирование зависших узлов; обеспечение автоматического обнаружения узлов, вернувшихся в работоспособное состояние.

2) *Автоматическая очистка*. Каждый процесс, сопоставленный заданию или действию, отслеживается и завершается после выполнения задания или действия на всех узлах, что предотвращает появление неконтролируемых процессов на вычислительных узлах.

3) *Безопасность*. Каждое задание или действие выполняется в контексте отправившего пользователя с обеспечением безопасности на протяжении всего процесса. Поскольку часто высокопроизводительные кластеры внедряются с целью поддержки критически важных приложений, значение безопасности и интеграции с существующей инфраструктурой высоко. Используя службу каталогов Active Directory, CCS обеспечивает систему безопасности на основе ролей для администрирования кластера и всех заданий. Планировщик запускает каждое задание в контексте и с учетными данными отправителя, а не привилегированного пользователя; все учетные данные хранятся с заданием и удаляются после его завершения. Это позволяет вычислительным заданиям получать доступ к сетевым ресурсам (например, файловым серверам и серверам баз данных) в контексте пользователя, а системным администраторам – применять и проверять политики безопасности, используя существующие и хорошо знакомые механизмы Active Directory. Обмен данными в рамках управления заданиями осуществляется по шифрованным каналам с проверкой подлинности, учетные данные известны только диспетчеру узла на время выполнения задания. Изолированию учетных данных и защите их целостности способствует то, что вычислительному процессу виден только маркер входа в систему, а не фактические учетные данные. Комплексная безопасность при работе с протоколом MS-MPI обеспечивается в CCS путем применения защищенных и шифрованных каналов на всем протяжении выполнения заданий. После того как диспетчер узла планирует и назначает задание и порождаются действия, задание всегда выполняется в контексте планирующего пользователя.

– *Microsoft Message Passing Interface (MS-MPI)*. Протокол MS-MPI широко используется существующими кластерами для высокопроизводительных вычислений и представляет собой высокоскоростной сетевой интерфейс. Этот протокол построен на основе стандарта MPI2 Арагонской Национальной Лаборатории и полностью совместим с ним и другими реализациями MPI, а также поддерживает комплексный интерфейс API, обеспечивающий вызов более 160 функций. В Windows Compute Cluster Server 2003 протокол MS-MPI для повышения произво-

дительности и эффективности работы центрального процессора использует протокол WinSock Direct. Через драйверы Winsock Direct, предоставляемые производителями оборудования, протокол MS-MPI может работать по любому каналу связи Ethernet, который поддерживается Windows Server 2003, а также по каналам связи с небольшими задержками и высокой пропускной способностью, например InfiniBand и Myrinet. Технология Gigabit Ethernet обеспечивает создание высокоскоростной и экономически эффективной схемы соединений, а InfiniBand подходит для чувствительных к задержкам приложений, которым требуется высокая пропускная способность. Протокол MS-MPI включает поддержку для языков программирования C, Fortran77 и Fortran90, а Microsoft Visual Studio имеет в своем составе параллельный отладчик для MS-MPI. Разработчик может запустить приложение MPI на нескольких вычислительных узлах прямо из среды Visual Studio, а затем Visual Studio автоматически соединяет процессы на каждом узле, позволяя разработчику индивидуально приостанавливать и проверять программные переменные на отдельных узлах.

– *Общая и частная сети.* Вычислительные узлы часто соединены друг с другом через несколько сетевых интерфейсов. С помощью частной сети администраторы могут конфигурировать компьютерные кластеры для управления и развертывания узлов. Трафик MPI может использовать частную сеть управления, но наибольший уровень быстродействия достигается при использовании выделенной сети, поддерживающей только трафик MPI.

Windows Compute Cluster Server 2003 поддерживает пять различных сетевых топологий с количеством сетевых интерфейсных плат на каждом узле от одной до трех. Топология определяет способ установки.

Вариант 1: Три сетевые платы на каждом узле

Одна сетевая плата подключена к общей (корпоративной) сети, другая – к выделенной частной сети для управления кластером, а третья – к выделенной высокоскоростной сети Message Passing Interface (MPI).

Этот вариант подходит для организаций, которым требуется выполнять тесно связанные параллельные приложения, потому что вычислительные данные передаются по выделенной сети MS-MPI, а управление и развертывание могут производиться по частной сети. Кроме того, для отладки кода приложения, когда происходят ошибки, разработчики могут использовать общую сеть для связи с вычислительными узлами.

Вариант 2: Три сетевые платы на ведущем узле, по две на каждом вычислительном узле

Ведущий узел обеспечивает преобразование сетевых адресов между вычислительными узлами и общей сетью; каждый вычислительный узел имеет подключение к частной сети и к высокоскоростному протоколу, например MPI.

Этот вариант подходит для тесно связанных параллельных приложений, потому что их вычислительный трафик проходит по выделенной частной сети. При такой конфигурации при выполнении параллельных приложений отсутствует сетевая задержка. Ведущий узел поддерживает ICS между вычислительными узлами и общей сетью, а также развертывание RIS.

Поддержка отладки ограничена, потому что доступ к вычислительным узлам происходит через сам ведущий узел.

В качестве недостатка первых двух вариантов можно отметить, что администраторы должны вручную устанавливать IP-адреса на каждом интерфейсе MS-MPI каждого узла, лучше всего в тот момент, когда узел подключен к кластеру, но еще не активирован.

Вариант 3: Две сетевые платы на каждом узле

Одна сетевая плата подключена к общей (корпоративной) сети, а другая – к выделенной частной сети кластера.

При такой конфигурации частная сеть используется для управления кластером, например RIS развертывания образов узлов, и трафика MS-MPI. Поскольку каждый вычислительный узел также напрямую связан с общей сетью, отладка приложений, выполняющихся на узлах, осуществляется проще: программисты при необходимости могут связываться с узлами напрямую, что делает отладку более эффективной.

Этот вариант, как и некоторые другие, поддерживает тесно связанные параллельные приложения, но здесь имеется достоинство, состоящее в том, что ведущий узел не является узким местом, пропускающим все вычислительные результаты через себя при передаче работающей программе.

В этом варианте может использоваться интегрированная коммуникативная система (Integrated Communications System, ICS), потому что каждый вычислительный узел может напрямую взаимодействовать с обслуживающими серверами DHCP и DNS через общую сеть. Если используется RIS, на ведущем узле для частной сети должна быть настроена полнофункциональная служба DHCP для поддержки RIS, потому что для RIS требуется автоматическое выделение IP-адресов в процессе удаленной установки.

Вариант 4: Две сетевые платы на ведущем узле, по одной на вычислительных узлах

В этой конфигурации ведущий узел обеспечивает ICS между вычислительными узлами и общей сетью. Сетевая плата общей сети на веду-

щем узле регистрируется в DNS общей сети, а сетевая плата частной сети контролирует все взаимодействия в кластере. Ведущий узел играет роль шлюза для всех взаимодействий между общей сетью и вычислительными узлами кластера. Частная сеть используется для управления и развертывания всех вычислительных узлов; ее также можно использовать для высокоскоростного вычислительного трафика MS-MPI.

Поскольку эта конфигурация опирается на интегрированную коммуникативную систему, вычислительные узлы скрыты ведущим узлом. Эта конфигурация может поддерживать RIS для упрощения развертывания вычислительных узлов и может использоваться для выполнения тесно связанных параллельных приложений. Поскольку между общей сетью и вычислительными узлами нет прямого взаимодействия, администраторы должны выполнять приложения, производя отладку непосредственно на вычислительных узлах или на отдельных системах частной сети.

Для первых четырех вариантов, если на ведущем узле установлен Active Directory, также требуется установка обновлений интегрированной коммуникативной системы ICS. Кроме того, при использовании RIS нужны обновления RIS.

Вариант 5: По одной сетевой плате на каждом узле

Весь сетевой трафик проходит по общей сети. В этом сценарии с ограниченными сетевыми возможностями развертывание вычислительных узлов с помощью служб RIS не поддерживается, т. е. каждый вычислительный узел необходимо установить и активировать вручную.

Этот вариант подходит для тестирования и получения общего представления о вычислительных кластерах или для сценариев, когда вычисления на одном узле не зависят от других узлов. Эта конфигурация не очень подходит для параллельных вычислений, потому что они могут вызвать большое количество взаимодействий внутри кластера, что может негативно повлиять на работу в сети.

Кроме того, поскольку на каждом узле используется по одной сетевой плате, развертывание компьютерных узлов RIS не поддерживается. Каждый вычислительный узел должен быть установлен вручную.

Проанализировав все топологии, можно прийти к выводу, что вариантом, требующим наименьших материальных затрат и подходящим для установки на небольшом числе узлов, является пятый, но он обладает рядом недостатков, связанных с прохождением всего сетевого трафика по общей сети. На такой конфигурации практически невозможно получить высокие показатели производительности и ускорения, однако для получения общего представления о кластере может быть достаточно и такого варианта. Конеч-

но, при наличии материальных средств, лучше использовать варианты 1–4 (чем меньше номер варианта, тем он дороже, но тем больше достоинств такой кластер имеет).

Модель кластера, выбранная администратором, зависит от того, какая вычислительная цель стоит перед кластером, а также от конфигурации серверов. Оптимальное развертывание зависит от тех приложений, которые работают на кластере.

Для установки Windows Compute Cluster Server 2003 требуется два компакт-диска. Первый CD включает операционную систему Windows Server 2003, Compute Cluster Edition. Второй CD содержит Compute Cluster Pack – комбинацию интерфейсов, утилит и инфраструктуры управления, составляющие основу Windows Compute Cluster Server 2003. Администратор использует эти диски для конфигурации ведущего и вычислительных узлов.

В процессе установки Windows Compute Cluster Server 2003 можно выделить два этапа. На первом происходит установка операционной системы на ведущем узле, он присоединяется к существующему домену Active Directory, а также устанавливается пакет Compute Cluster Pack. После этого, если планируется автоматическое развертывание вычислительных узлов с помощью служб RIS, в рамках выполнения списка поручений устанавливаются и настраиваются службы RIS. После установки пакета Compute Cluster Pack отображается список поручений, т. е. действий, необходимых для окончательной настройки вычислительного кластера, включая определение топологии сети, настройку служб RIS с помощью соответствующего мастера, добавление вычислительных узлов в состав кластера, а также настройку пользователей и администраторов кластера [10].

Заключение

За счет объединения возможностей стандартных компьютеров с архитектурой x64, удобства и безопасности службы каталогов Active Directory, а также преимуществ операционной системы Windows платформа Windows Compute Cluster Server 2003 позволяет создавать недорогие, обладающие повышенной безопасностью решения для высокопроизводительных вычислений. Windows Compute Cluster Server 2003 быстро и без труда развертывается с помощью стандартных технологий развертывания Windows, а дополнительные вычислительные узлы можно добавлять просто путем их включения в сеть и присоединения к кластеру. Протокол Microsoft Message Passing Interface полностью совместим со стандартом MPICH2. Интеграция со службой Active Directory позволяет организовать систему безопасности на основе ролей, а консоль управ-

ления обеспечивает знакомый интерфейс администрирования и планирования.

Таким образом, Windows Compute Cluster Server 2003 хорошо подходит для кластера рабочих станций в вузе. CCS позволяет создавать высокоскоростные кластеры при использовании популярной платформы Windows, отличающиеся по размеру от всего нескольких до сотен вычислительных узлов. Следование рекомендациям, представленным в статье, упростит проектирование, установку, развертывание и эксплуатацию кластера рабочих станций в вузе. Самым простейшим вариантом кластера на основе CCS 2003, который можно применять в учебном процессе, является конфигурация, использующая по одной сетевой плате на каждом узле.

Примечания

1. *Лацис, А.* Как построить и использовать суперкомпьютер [Текст] / А. Лацис. М.: Бестселлер, 2003; Страницка для начинающих пользователей вычислительных кластеров [Электронный ресурс]. Режим доступа: www.parallel.ru/cluster/beginner_guide.html; *Комолкин, А. В.* Программирование для высокопроизводительных ЭВМ [Электронный ресурс] / А. В. Комолкин, С. А. Немнюгин. Режим доступа: www.hpc.nw.ru/COURSES/НРС; *Яровой, Г. П.* Параллельные информационные технологии на базе вычислительного кластера [Электронный ресурс] / Г. П. Яровой, Ю. Н. Радаев, Ю. А. Родичев, В. А. Салеев. Режим доступа: <http://lvv.ssu.samara.ru/upload/article.doc>; Кластер МГИУ [Электронный ресурс]. Режим доступа: <http://www.ctc.msiu.ru/CLUSTER/>; *Барский, А. Б.* Параллельные информационные технологии [Текст] / А. Б. Барский. М.: Интернет-университет информационных технологий – ИНТУИТ.ру, Бином. Лаборатория знаний, 2007; *Гофф, М. К.* Сетевые распределенные вычисления: достижения и проблемы [Текст] / Макс К. Гофф. М.: КУДИЦ-ОБРАЗ, 2005; *Эндрюс, Г. Р.* Основы многопоточного, параллельного и распределенного программирования [Текст] / Грегори Р. Эндрюс. М.: Изд. дом «Вильямс», 2003.

2. Microsoft выпускает Windows Compute Cluster Server 2003 [Электронный ресурс]. Режим доступа: <http://newsdesk.pcmag.ru/node/1079>; Microsoft Windows Compute Cluster Server 2003 – Overview & Features [Электронный ресурс]. Режим доступа: HP.com.; *Рассел, Ч.* Обзор Microsoft Windows Compute Cluster Server 2003 [Текст] / Чарли Рассел. 2005; ИТ Центр Рыбасова [Электронный ресурс]. Режим доступа: <http://www.it-cr.ru/soft/server-solutions/microsoft/wccs/>; Microsoft Windows Compute Cluster Server 2003 [Электронный ресурс]. Режим доступа: <http://www.microsoft.com/rus/hpc>; Системные требования для Windows Compute Cluster Server 2003 [Электронный ресурс]. Режим доступа: <http://www.microsoft.com/rus/windowsserver2003/ccs/sysreqs.aspx>; Deploying and Managing Microsoft Windows Compute Cluster Server 2003 [Электронный ресурс]. Режим доступа: <http://207.46.196.114/windowsserver/en/library/9330fd8-c680-425f-8583-c46ee77306981033.msp?mfr=true>.

3. Microsoft выпускает Windows Compute Cluster Server 2003...

4. Microsoft Windows Compute Cluster Server 2003...

5. Суперкомпьютеры TOP500 [Электронный ресурс]. Режим доступа: <http://www.top500.org/>.

6. Суперкомпьютеры TOP50 [Электронный ресурс]. Режим доступа: <http://supercomputers.ru/?page=rating>.

7. Учебный курс: CS338. Многопроцессорные вычислительные системы и параллельное программирование [Электронный ресурс]. Режим доступа: http://www.software.unn.ac.ru/ccam/mskurs/RUS/HTML/cs338_ppr_materials.htm.

8. *Рассел, Ч.* Указ. соч.; Системные требования для Windows Compute Cluster Server 2003...

9. *Рассел, Ч.* Указ. соч.; Deploying and Managing Microsoft Windows Compute Cluster Server 2003...

10. Deploying and Managing Microsoft Windows Compute Cluster Server 2003...; *Гергель, В. П.* Теория и практика параллельных вычислений [Текст] / В. П. Гергель. М.: Интернет-университет информационных технологий – ИНТУИТ.ру, Бином. Лаборатория знаний, 2007.

А. В. Лялин

БИТОВАЯ АРИФМЕТИКА И ПОИСК ПОДСТРОКИ В СТРОКЕ

В статье рассматривается один из многочисленных алгоритмов поиска подстроки в строке, так называемый алгоритм Сдвиг-И, активно использующий битовую арифметику. С одной стороны, это пример того, как низкоуровневые операции находят применение в высокоуровневом программировании. С другой – что более важно – пример взаимосвязи информатики и математики в решении реальных прикладных проблем. Материал построен по принципу «от простого к сложному», развивается от задачи к задаче и представляет собой конспект занятий.

Подобно тому, как все искусства тяготеют к музыке, все науки стремятся к математике.

Джордж Сантаяна

Каждый из нас лишь выиграет, создавая время от времени «игрушечные» программы с заданными искусственными ограничениями, заставляющими нас до предела напрягать свои способности... Искусство решения мини-задач на пределе своих возможностей оттачивает наше умение для реальных задач.

Дональд Кнут

Казалось бы, элементарная задача – поиск слова в тексте (или подстроки в строке), а решение было предложено десятки и десятки. Сформировался даже новый раздел информатики – методы обработки строк. Алгоритмы над стро-

ЛЯЛИН Андрей Васильевич – аспирант кафедры информатики и методики обучения информатике ВятГУ
© Лялин А. В., 2008

ками находят применение как в простейших текстовых редакторах, так и в молекулярной биологии. Один из таких алгоритмов, а именно Сдвиг-И, мы и рассмотрим.

План следующий. *Первое.* Решаем задачи на битовую арифметику. Вроде бы «игрушечные» и бесполезные. Но, как окажется вскоре, вся эффективность алгоритма Сдвиг-И построена как раз на битовых операциях. *Второе.* Чётко формулируем проблему поиска подстроки в строке, приводим «лобовое» решение и изучаем новый алгоритм. *И третье.* Пожалуй, самое интересное. Расширяем возможности алгоритма. Дело в том, что он не только быстро работает при точном поиске, но и легко обобщается на случай нечёткого поиска. Например, если слово было написано с ошибками или ошибки содержались в тексте.

1. Битовая арифметика

Случайные открытия делают только подготовленные умы.

Блез Паскаль

Битовые операции очень близки логическим в классической логике. 1 рассматривается как «истина», а 0 – как «ложь». Они применимы только к целым числам и выполняются поразрядно над их двоичными представлениями.

• Для каждой пары битов результат операции *And* (битовое и) есть 1, если оба бита равны 1, и 0 в остальных случаях. *Пример.* 00110101 *And* 01100110 = 00100100.

• Для каждой пары битов результат операции *Or* (битовое или) есть 0, если оба бита равны 0, и 1 в остальных случаях. *Пример.* 00110101 *Or* 01100110 = 01110111.

• Для каждой пары битов результат операции *Xor* (битовое исключающее или) есть 1, если биты различны, и 0, если они совпадают. *Пример.* 00110101 *Xor* 01100110 = 01010011.

• Операция *Not* (битовое отрицание) инвертирует каждый бит числа. *Пример.* *Not*(01001011) = 10110100.

• К битовым операциям также относятся сдвиг влево *Sbl* и сдвиг вправо *Sbr*. Операция $x \text{ Sbl } n$ сдвигает двоичное представление числа x влево на n разрядов, а операция $x \text{ Sbr } n$ – вправо. При этом биты, уходящие за пределы разрядной сетки, теряются, а освободившиеся – заполняются нулями. *Пример.* 00010110 *Sbl* 2 = 01011000. 00010110 *Sbr* 3 = 00000010.

Освоимся в новой арифметике. Задачи. Решаются только битовые операции и операции сложения-вычитания. Циклы и ветвления запрещаются. Числа – неотрицательные, и для их представления используется 1 байт. Все примеры необходимо проиллюстрировать.

1. Определите результат выполнения следующих операций. 31 *And* 240, 81 *Or* 32, 240 *Xor* 128, *Not*(102), 18 *Sbl* 2, 37 *Sbr* 3.

2. Установите равным единице младший бит числа. *Ответ:* $x' = x \text{ Or } 1$.

3. Определите значение младшего бита числа. Какой операции с целыми числами это действие эквивалентно? *Ответ:* $\text{bit} = x \text{ And } 1 = x \text{ Mod } 2$.

4. Инвертируйте младший бит числа. *Ответ:* $x' = x \text{ Xor } 1$.

5. Обнулите младший бит числа. *Ответ:* $x' = x \text{ And } \text{Not}(1)$.

6. Обнулите четыре старших бита. *Ответ:* $x' = x \text{ And } 15$.

7. Обнулите четыре младших бита. *Ответ:* $x' = x \text{ And } 240$.

8. Умножьте число на k -ю степень двойки. *Ответ:* $x \cdot 2^k = x \text{ Sbl } k$ (сдвиг влево на один разряд эквивалентен умножению на два).

9. Вычислите целую часть от деления неотрицательного числа на степень двойки 2^k . *Ответ:* $x \text{ Div } 2^k = x \text{ Sbr } k$ (сдвиг вправо на один разряд эквивалентен делению нацело на два).

10. Даны два неравных неотрицательных числа n и m , не превосходящие 7. Найдите значение суммы $2^n + 2^m$, без операции сложения. *Ответ:* $s = (1 \text{ Sbl } n) \text{ Or } (1 \text{ Sbl } m)$.

11. Постройте число, в котором k младших бит нулевые, остальные единичные. Например, при $k = 4$ это будет число 11111000. *Ответ:* $x = \text{Not}(0) \text{ Sbl } k$.

12. Постройте число, в котором k младших бит единичные, остальные нулевые. Например, при $k = 3$ это будет число 00000111. *Ответ:* $x = \text{Not}(\text{Not}(0) \text{ Sbl } k)$. А если без отрицания? *Ответ:* $x = 1 \text{ Sbl } k - 1$.

13. Составьте выражение, принимающее значение 0, если положительное число имеет вид $2^n - 1$, иначе принимающее значение, не равное 0. Так, число 00001111 – такого вида, а число 01101011 – нет. *Ответ:* $x \text{ And } (x + 1)$.

14. Составьте выражение, принимающее значение 0, если положительное число – степень двойки, иначе принимающее значение, не равное 0. Так, число 00001000 – такого вида, а число 01001010 – нет. *Ответ:* $x \text{ And } (x - 1)$.

15. Пусть t – степень двойки. Вычислите остаток при делении числа на t . *Ответ:* $x' = x \text{ And } (t - 1)$.

16. Установите равным единице крайний справа нулевой бит. Например, было число 11000111, а стало 11001111. *Ответ:* $x' = x \text{ Or } (x + 1)$.

17. Распространите вправо крайний правый единичный бит числа. Так, число 10101000 преобразуется к виду 10101111. *Ответ:* $x' = x \text{ Or } (x - 1)$.

18. Распространите вправо крайний правый единичный бит, обнулив при этом все остальные биты. Так, число 10101000 преобразуется к виду 00001111. *Ответ:* $x' = x \text{ Xor } (x - 1)$.

19. Установите равными единице «завершающие» нулевые биты числа, обнулив при этом все остальные биты. Так, число 01101000 преобразуется к виду 00000111. *Ответ:* $x' = \text{Not}(x) \text{ And } (x - 1)$.

20. Обнулите все биты числа, кроме k младших. *Ответ:* $x' = x \text{ And } (1 \text{ Sbl } k - 1)$.

21. Обнулите k младших битов положительного целого числа. Так, при $k = 3$ из 01001100 получаем 01001000. *Ответ:* $x' = x \text{ And } (\text{Not}(0) \text{ Sbl } k)$. И это всё равно что округлить число в меньшую сторону к ближайшему числу, кратному степени двойки 2^k .

22. Установите равным единице k -й бит числа. Так, при $k = 4$, число 01000101 преобразуется в 01001101. *Ответ:* $x' = x \text{ Or } (1 \text{ Sbl } (k - 1))$.

23. Инвертируйте k -й бит числа. Так, при $k = 4$, 01001101 \rightarrow 01000101. *Ответ:* $x' = x \text{ Xor } (1 \text{ Sbl } (k - 1))$.

24. Обнулите k -й бит числа. Так, при $k = 4$, 01001101 \rightarrow 01000101. *Ответ:* $x' = x \text{ And Not } (1 \text{ Sbl } (k - 1))$.

25. Определите k -й бит числа. Так, третий бит числа 11001111 равен 1. *Ответ:* $\text{bit} = (x \text{ Sbr } (k - 1)) \text{ And } 1$.

26. Выделите крайний справа нулевой бит числа. Необходимо, например, из числа 11001111 получить 00010000. *Ответ:* $x' = \text{Not}(x) \text{ And } (x + 1)$.

27. Выделите крайний справа единичный бит числа. Необходимо, например, из числа 01101000 получить 00001000. *Ответ:* $x' = x \text{ And } (\text{Not}(x) + 1)$.

Примеры можно продолжить. Но переходим к главному – реальной практической проблеме, поиску подстроки в строке. Как ни странно, битовые операции здесь найдут своё достойное применение...

2. Алгоритм Сдвиг-И

Стену можно пробить только головой. Все остальное – только орудия.

Лешек Кумор

Прежде четко сформулируем проблему. Строка – это последовательность символов из некоторого алфавита. Имеется две строки w и t . Длина первой – m символов, второй – n символов. Назовём w словом, а t – текстом. Задача состоит в поиске всех позиций, начиная с которых слово входит в текст.

Так, если $w = \text{cocos}$, а $t = \text{cocosxhcocosc}$, то вхождения слова будут начиная с первого и девятого символов.

Формулировка проста. И первое решение, что приходит на ум (прямое, «лобовое», грубой силы), – это перебирать все позиции текста t и смотреть, не с этой ли позиции слово w входит в текст.

```

Procedure SimpleSearch(w,t:String);
Var n,m,i,j:Byte;
Begin
  n:=length(t); m:=length(w);
  For i:=1 To n-m+1 Do begin
    j:=1;
    While (j<=m) And (w[j]=t[i+j-1]) Do
      Inc(j);
    If j>m Then Writeln(i);
  End;
End;

```

Для каждого i сравниваются не более m символов, а всего сравнений не более $m*(n - m + 1)$. И их ровно $m*(n - m + 1)$ лишь для некоторых вырожденных случаев. Скажем, при $w = \text{aaaab}$ и $t = \text{aaaaaaaaaaaaaaaaaab}$. Так что в худшем случае временная сложность $O(mn)$.

Но при поиске в обычных текстах такие примеры встречаются редко, и в среднем потребуется $O(n + m)$ времени. Ведь несопадающие символы, как правило, стоят не на последнем месте, а гораздо раньше. И всё же прямой алгоритм работает медленно. Существуют более удачные и оригинальные решения. Одно из них – алгоритм Сдвиг-И...

Смелые мысли играют роль передовых шашек: они гибнут, но обеспечивают победу.

И. Гёте

Пусть снова слово $w = \text{cocos}$, а текст $t = \text{cocosxhcocosc}$. Будем искать несколько больше, чем от нас требуется – не только все вхождения w в t , но и вхождения в t всех возможных префиксов w . А префикс – это любое из начал слова. Так, w имеет пять префиксов: c , co , coc , $coco$ и $cocos$.

Если точнее, для каждой позиции текста нам необходимо знать, является ли она концом вхождения не только слова, но и любого из его префиксов.

С этой целью для каждой позиции текста построим m -элементный массив из нулей и единиц. В нём j -й элемент равен 1, если j -й префикс слова входит в предыдущий текст и заканчивается на этой позиции, иначе – j -й элемент равен 0. В итоге имеем таблицу из n строк и m столбцов:

	c	o	c	o	s
c	1	0	0	0	0
o	0	1	0	0	0
c	1	0	1	0	0
o	0	1	0	1	0
s	0	0	0	0	1
x	0	0	0	0	0
c	1	0	0	0	0
x	0	0	0	0	0
c	1	0	0	0	0
o	0	1	0	0	0
c	1	0	1	0	0
o	0	1	0	1	0
s	0	0	0	0	1
c	1	0	0	0	0

Например, как заполнялся массив для четвёртой позиции текста, четвёртая строка таблицы. Первый префикс c не входит в предыдущий текст, заканчиваясь на данной позиции, поэтому первый элемент массива равен 0. Второй префикс co входит в текст и заканчивается на данной позиции, поэтому второй элемент равен 1. Далее аналогично рассматриваются префиксы coc (не входит), $coco$ (входит) и $cocos$ (не входит) и заполняются третий, четвёртый и пятый элементы. Они равны 0, 1 и 0 соответственно.

Нас интересуют прежде всего, конечно, пятая и тринадцатая строки таблицы. Единица в их конце означает, что в текст входит всё слово, а данная позиция является концом его вхождения.

Остается научиться строить такую таблицу быстро, строка за строкой. Пусть R_i – i -я строка таблицы. На самом деле, каждая строка R_i зависит только от предыдущей строки R_{i-1} , слова и текущего символа текста $t[i]$.

Итак, пусть у нас уже имеется строка R_{i-1} . На обработку поступил символ текста $t[i]$. Строим следующую строку R_i . Так, $R_i[5] = 1$, то есть пятый префикс $cocos$ входит в текст и заканчивается на текущей позиции i , только если одновременно:

(1) в текст входит и заканчивается на предыдущей позиции $i-1$ меньший префикс $coco$, или $R_{i-1}[4] = 1$.

(2) текущий символ текста $t[i]$ совпадает с пятым символом слова, или $t[i] = w[5]$.

Первое условие означает, что в текст входят все первые символы префикса $cocos$ до последнего, а второе условие, что в текст входит и последний символ префикса $cocos$.

Аналогично, $R_i[4] = 1$ только тогда, когда одновременно $R_{i-1}[3] = 1$ и $t[i] = w[4]$ и т. д. Другими словами, очередная строка таблицы строится из предыдущей по такой формуле:

$$R_i[j] = \begin{cases} 1, & \text{если } R_{i-1}[j-1] = 1 \text{ и } t[i] = w[j], \text{ для} \\ 0, & \text{в остальных случаях} \end{cases}$$

всех $1 \leq j \leq m$.

Единственное исключение возникает при заполнении первой позиции $R_i[1]$, для которой условие (1) очевидно всегда истинно, поскольку предыдущих символов попросту нет. Чтобы формула работала и в этом случае, будем хранить в дополнительном нулевом элементе каждой строки единицу. Пусть $R_i[0] = 1$ для всех $1 \leq i \leq n$.

В итоге имеем следующий алгоритм поиска подстроки в строке. «Бежим» по тексту. Для каждой его позиции i строим битовый массив, используя битовый массив для предыдущей позиции. Проверим, появилась ли 1 в конце нового массива. Если да, то слово входит в текст и заканчивается на текущей позиции i . Следовательно, начинается с позиции $i - m + 1$.

{Ищет слово w в тексте t }

Procedure Search(w, t :String);

Const Wsize=...; {Максимальный размер слова w }

Var R:Array [0..Wsize] of **Byte**; {Битовый массив}

n, m, i, j :**Byte**;

Begin

$n := \text{length}(t)$; $m := \text{length}(w)$;

$R[0] := 1$; **For** $j := 1$ **To** m **Do** $R[j] := 0$;

For $i := 1$ **To** n **Do Begin**

For $j := m$ **DownTo** 1 **Do** {Вычисляем битовый массив для текущего символа текста, используя старый массив.

Кстати, почему вычисляем, начиная с последнего элемента?}

If ($R[j-1] = 1$) **And** ($t[i] = w[j]$)

Then $R[j] := 1$

Else $R[j] := 0$;

If $R[m] = 1$ **Then** WriteLn($i - m + 1$);

{Если последний элемент единичный, слово найдено}

End;

End;

Да, на каждом шаге мы можем получить гораздо больше информации – не только о вхождении всего слова, но и любого из его префиксов. Однако по эффективности новый алгоритм даже хуже прямого. Для каждого i всегда выполняется ровно m сравнений. Временная сложность в любом случае $O(mn)$. На первый взгляд, попытка оказалась неудачной, но сделаем следующий шаг...

Ты не забывай, что у меня в голове опилки и длинные слова меня только огорчают...

Алан Милн «Винни-Пух и все-все-все»

Если длина искомого слова не превышает 32, то любую из строк таблицы, а это массив из нулей и единиц, можно представить в виде одной целочисленной переменной типа *LongInt*. Целое число ведь тоже состоит из нулей и единиц, в его двоичном представлении. Вот здесь-то и пригодится битовая арифметика. Оказывается, очередную битовую строку можно будет вычислять всю сразу, без цикла по ней, за несколько битовых операций.

Рассмотрим, например, как по третьей строке 10100 строится четвёртая 01010. Напомним, единица в четвёртой строке на некоторой позиции j может появиться только при одновременном выполнении двух условий:

(1) в третьей строке слева от позиции j стоит единица (в текст входят все первые символы j -го префикса до последнего).

(2) текущий символ текста (для нашего примера это o) совпадает с j -м символом слова

(в текст входит и последний символ j -го префикса).

Для проверки условия (1) достаточно сдвинуть вправо третью строку 10100. А так как в дополнительном нулевом элементе всегда хранится единица, получаем 11010. Вспомним, что обычный сдвиг *Shr* заполняет освободившийся разряд нулём. Поэтому после сдвига ещё необходимо установить единицу в первый элемент. В тех позициях новой строки, где стоят единицы, возможно, будут стоять единицы и четвёртой строки. Для всех префиксов, помеченных этими единицами, их первые символы до последнего уже точно входят в текст.

Для проверки условия (2) заранее для каждого символа алфавита подготавливаем характеристический вектор длиной m . В нём j -й элемент равен 1, если данный символ совпадает с последним символом j -го префикса (или просто совпадает с j -м символом слова), и равен 0 в противном случае. Имеем

	c	o	c	o	s
c	1	0	1	0	0
o	0	1	0	1	0
s	0	0	0	0	1

Для всех же остальных символов алфавита характеристические векторы будут нулевыми 00000, так как этих символов вообще нет в слове.

Итак, нам известен характеристический вектор текущего символа текста. Для символа *o* это 01010. В тех позициях вектора, где стоят единицы, возможно, будут стоять единицы и четвёртой строки. Для всех префиксов, помеченных этими единицами, последний символ совпадает с текущим символом текста.

Осталось объединить оба условия, они должны выполняться одновременно. Позиции, для которых и в третьей строке, сдвинутой вправо, и в векторе стоит единица (префикс полностью входит в текст), получают также единичное значение, остальные обнуляются. Используется, конечно, битовая операция *And* (И). Эта строка и становится четвёртой. 11010 *And* 01010 = 01010.

Всё готово, чтобы сформулировать окончательный вариант алгоритма Сдвиг-И. Вначале для каждого символа алфавита вычисляем его характеристический вектор – целое число типа *LongInt*. Далее «бежим» по тексту.

Как и раньше, для каждой его позиции i строим битовую строку. Но храним её уже в одной переменной типа *LongInt*. Сдвигаем старую строку вправо, в её первый элемент устанавливаем единицу и объединяем через *And* с характеристическим вектором текущего символа текста. Опять же, но с помощью битовых операций, проверяем, появилась ли 1 в конце новой строки. Если да, то слово входит в текст и заканчивается

на текущей позиции i . А значит, начинается с позиции $i - m + 1$. Первоначально же битовая строка нулевая.

{Алгоритм Сдвиг-И ищет слово w в тексте t }

Procedure SearchShiftAnd(w, t :String);

Var V:Array [chr(0)..chr(255)] Of **LongInt**; {Массив для хранения характеристических векторов всех символов таблицы ASCII, от 0-го до 255-го.}

n, m, i :Byte; R, first: **LongInt**;

j :Char;

Begin

n :=length(t); m :=length(w);

For j :=chr(0) **To** chr(255) **Do**

V[j]:=0;

For i :=1 **To** m **Do** V[w [i]]:=V[w [i]] **Or** (1 **Shl** ($m-i$)); {Вычисляем характеристические вектора.}

$first$:=1 **Shl** ($m-1$); {Число вида 10...0, используется для установки 1 в первую позицию битовой строки.}

R:=0;

For i :=1 **To** n **Do** **Begin**

R:=(R **Shr** 1) **Or** $first$) **And**

V[t [i]]; {Вычисляем битовую строку для текущей позиции.}

If R **And** 1=1 **Then** Writeln($i-m+1$);

{Если последний бит единичный, слово найдено}

End;

End;

Алгоритм по существу получился линейным, с временной сложностью $O(n)$. Внутренний цикл мы заменили всего тремя битовыми операциями, а они выполняются даже быстрее обычных арифметических. Очередная строка таблицы вычисляется сразу!

А что касается ограничения на длину слова – максимум 32 символа, то на практике это вполне приемлемо. Не будем же мы искать слова длиной более 32 в обычном тексте.

Всё на первый взгляд очень запутанно. Поначалу даже не верится, что эта программа ищет слово в тексте. Но это так. Когда приходит понимание, всё кажется уже простым и красивым. И это ещё не предел. Алгоритм имеет замечательную возможность – его легко расширить для нечёткого поиска, поиска слов с ошибками и не только. Но прежде несколько упражнений.

1. Проведите трассировку алгоритма Сдвиг-И. Скажем, при $w = abac$ и $t = aabaccababacab$.

2. Экспериментально сравните скорости прямого и нового алгоритма.

3. В алгоритме Сдвиг-И на вхождение префиксов указывает 1. Пусть теперь всё наоборот – на вхождение указывает 0. На самом деле это уже будет не алгоритм Сдвиг-И, а Сдвиг-Или.

Запрограммируйте его и вы увидите, что в основном цикле на одну битовую операцию станет меньше. Дополнение единицей при сдвиге не требуется.

{Алгоритм Сдвиг-Или ищет слово w в тексте t }

```

Procedure SearchShiftOr( $w, t$ :String);
  Var V:Array [chr(0)..chr(255)] Of
  LongInt;
       $n, m, i$ :Byte;  $h, R$ : LongInt;
   $j$ :Char;
Begin
   $n$ :=length( $t$ );  $m$ :=length( $w$ );
   $h$ :=1 Shl  $m$  -1;
  For  $j$ := chr(0) To chr(255) Do V[ $j$ ]:=h;
  For  $i$ :=1 To  $m$  Do V[ $w[i]$ ]:=V[ $w[i]$ ]
And (Not(1 Shl ( $m-i$ )));
   $R$ :=h;
  For  $i$ :=1 To  $n$  Do Begin
     $R$ :=( $R$  Shr 1) Or V[ $t[i]$ ];
    If  $R$  And 1=0 Then Writeln( $i-m+1$ );
  End;
End;

```

3. Расширения алгоритма Сдвиг-И

Нам выпало – стремиться за пределы.
Зря, что ли, существуют небеса?!

Роберт Браунинг

Алгоритм Сдвиг-И работает быстрее прямого (надемся, вы это проверили), но, к сожалению, как отмечают его авторы (Baeza-Yates R. A., Gonnet H. G. New Approach to Text Searching // Communications of the ACM. 1992. October. 35(10). P. 74–82), при точном поиске уступает другим более быстрым алгоритмам. Но его достоинство в том, что он легко обобщается на случаи нечёткого, приблизительного поиска. Вот некоторые из таких задач.

1. Организуйте «регистронезависимый» поиск, то есть поиск слова вне зависимости от регистра букв.

Для этого достаточно для одних и тех же букв разного регистра построить одинаковые характеристические векторы. Основная же часть алгоритма остаётся без изменений.

2. Известно, что в тексте без цифр некоторые буквы были заменены цифрами. Найдите слово в таком «испорченном» тексте.

Следует всего лишь характеристические векторы цифр сделать единичными. Это будет означать, что цифра может появиться в любой позиции слова.

3. Автомобильные номера имеют вид – буква, три цифры, две буквы. Например, м815тк. Найдите в тексте все автомобильные номера.

Здесь достаточно характеристические векторы для всех букв положить равными 100011, для

цифр – 011100, а для всех остальных символов – нулевыми.

4. В тексте, например, некоторые буквы o были заменены буквой a . Как найти слово в таком «испорченном» тексте?

Как обычно строим характеристические векторы для всех символов алфавита. А после лишь корректируем вектор для буквы a , соединяя его через Or с вектором для буквы o . Тем самым даём знать алгоритму, что « a » может стоять в слове и на месте « o ».

5. Мы можем искать не конкретно слово, а слова по некоторому шаблону. Например, $c*s*s$, где «звёздочка» заменяет любой символ. По этому запросу требуется найти все слова из пяти букв, на первой и третьей позициях которых находится « c », а на последней « s ». Как необходимо изменить алгоритм для обработки подобных запросов?

Опять же после построения характеристических векторов остаётся только их скорректировать. На второй и четвёртой позициях (то есть на позициях со «звёздочками») во *всех* векторах поставить единицы. Это будет означать, что на этих позициях может находиться любой символ.

6. Пусть квадратные скобки указывают группу символов для поиска. Например, по запросу «б[аи]ржа» должны быть найдены слова «баржа» и «биржа». То есть слова, которые начинаются с «б», за которым следуют «а» или «и», и заканчиваются на «ржа».

Квадратные скобки определяют некоторую позицию слова. В характеристических векторах для всех символов в квадратных скобках устанавливаем единицу на данной позиции.

7. Пусть фигурные скобки указывают группу символов, которых не должно содержаться в искомом слове. Так, по запросу {ABC}ГГУ необходимо найти все слова из четырёх букв, не начинающиеся на «А», «В» и «С» и заканчивающиеся на «ГГУ».

Аналогично, фигурные скобки определяют некоторую позицию слова. В характеристических векторах для всех символов в фигурных скобках ставим ноль на данной позиции.

8. Предположим, что искомое слово в тексте содержится, возможно, с одним несовпадающим символом. Задача – несмотря на эту ошибку найти слово.

Для примера возьмём слово $w = cocos$ и текст $t = cocosxcocosc$. Тогда вхождения w в t будут начиная с первого и седьмого символов. Первое будет неточным, а второе – точным.

Для поиска точных вхождений строим таблицу $R1$, аналогичную описанной ранее. Для поиска же неточных вхождений – таблицу $R2$, которая похожа на первую, с той лишь разницей, что отражает как точные вхождения, так и вхождения при одном изменённом символе.

Снова для каждой позиции текста составляем битовую строку длиной m . В ней j -й элемент равен 1, если j -й префикс слова входит в предыдущий текст, возможно, с одной ошибкой замены, и заканчивается на этой позиции, иначе j -й элемент равен 0.

R1	с	о	с	о	с
с	1	0	0	0	0
о	0	1	0	0	0
с	1	0	1	0	0
о	0	1	0	1	0
с	1	0	1	0	0
х	0	0	0	0	0
с	1	0	0	0	0
о	0	1	0	0	0
с	1	0	1	0	0
о	0	1	0	1	0
с	1	0	0	0	1
с	1	0	0	0	0

R2	с	о	с	о	с
с	1	0	0	0	0
о	1	1	0	0	0
с	1	0	1	0	0
о	1	1	0	1	0
с	1	0	1	0	1
х	1	1	0	1	0
с	1	0	1	0	0
о	1	1	0	1	0
с	1	0	1	0	0
о	1	1	0	1	0
с	1	0	1	0	0
о	1	1	0	1	0
с	1	0	1	0	1
с	1	0	0	0	0

Единица в конце пятой и одиннадцатой строки $R2$ означает, что в текст, быть может, с одной ошибкой замены, входит всё слово, и данная позиция является концом его вхождения.

Как же быстро построить таблицу $R2$? Итак, пусть у нас уже имеется строка $R2_{i-1}$, а также строка $R1_{i-1}$. На обработку поступил символ текста $t[i]$. Строим следующую строку $R2_i$.

Единица в новой строке $R2$ для некоторого j -го префикса может появиться только в двух случаях.

(1) Все первые символы j -го префикса, кроме последнего, точно совпадают с текстом, то есть $R1_{i-1}[j-1] = 1$. Проверять на совпадение последний символ не требуется, он может быть и ошибочным, это позволительно.

Или...

(2) В первых символах j -го префикса уже допущена одна ошибка замены, но его последний символ совпадает с текущим символом текста, то есть $R2_{i-1}[j-1] = 1$ и $t[i] = w[j]$.

Если мы сдвинем вправо (с дополнением единицей) предыдущую строку таблицы $R1$, то получим все единицы новой строки $R2$ по первому случаю. Для всех префиксов, помеченных этими единицами, ошибка, быть может, только в последнем символе.

Если же мы сдвинем вправо (с дополнением единицей) предыдущую строку $R2$ и соединим её через And с характеристическим вектором текущего символа текста, то получим все единицы новой строки $R2$ по второму случаю. Для всех префиксов, помеченных этими единицами, ошибка, возможно, произошла где-то внутри.

Остаётся только собрать все единицы в новой строке $R2$ при помощи операции Or (Или).

Рассмотрим для примера, как строится восьмая строка $R2$. Седьмая строка $R1$ 10000 после сдвига и дополнения единицей переходит в 11000. Часть единиц восьмой строки $R2$ есть. Седьмая строка $R2$ 10100 после сдвига, дополнения единицей и операции And с характеристическим вектором текущего символа «о» 01010 переходит в 01010. Есть и другая часть единиц. Соединяем полученные строки через Or , так как нам нужны единицы и той и другой, $11000 Or 01010 = 11010$. Результат и будет восьмой строкой $R2$.

{Алгоритм Сдвиг-И ищет слово w в тексте t с одной ошибкой замены.}

```
Procedure SearchShiftCh(w,t:String);
  Var V:Array [chr(0)..chr(255)] Of
  LongInt;
```

```
  n,m,i:Byte; R1,R2,first:
  LongInt; j:Char;
```

```
Begin
  n:=length(t); m:=length(w);
  For j:=chr(0) To chr(255) Do V[j]:=0;
  For i:=1 To m Do V[w[i]]:=V[w[i]] Or
  (1 Shl (m-i));
```

```
  R1:=0; R2:=0; first:=1 Shl (m-1);
  For i:=1 To n Do Begin {Вычисляем
  битовые строки как первой, так и второ-
  вой таблицы.}
```

```
    R1:=(R1 Shr 1) Or first;
    R2:= R1 Or ((R2 Shr 1) And
  V[t[i]]);{Здесь дополнение единицей
  можно не делать, так как она уже есть
  в R1.}
```

```
    R1:=R1 And V[t[i]];
    If R2 And 1=1 Then Writeln(i-m+1);
  End;
```

9. Предположим, что искомое слово в тексте содержится, возможно, с одним лишним символом. Как, несмотря на эту ошибку вставки, найти слово?

Рассуждения будут во многом повторять предыдущие, об ошибке замены. Попробуйте их восстановить, прорисовать таблицы, придумать способ их построения и т. д. Для проверки приведём готовую программу.

{Алгоритм Сдвиг-И ищет слово w в тексте t с одной ошибкой вставки.}

```
Procedure SearchShiftIns(w,t:String);
  Var V:Array [chr(0)..chr(255)] Of
  LongInt;
```

```
  n,m,i:Byte; R1,R2,first:LongInt;
  j:Char;
```

```
Begin
  n:=length(t); m:=length(w);
  For j:=chr(0) To chr(255) Do
  V[j]:=0;
```

```

For i:=1 To m Do V[w[i]]:=V[w[i]] Or
(1 Shl (m-i));
R1:=0; R2:=0; first:=1 Shl (m-1);
For i:=1 To n Do Begin
  R2:= R1 Or ((R2 Shr 1) Or first)
And V[t[i]];
  R1:=((R1 Shr 1) Or first) And
V[t[i]];
  If R2 And 1=1 Then Writeln(i-m);
End;
End;

```

10. Предположим теперь, что искомое слово в тексте содержится, возможно, с одним удалённым символом. Как, несмотря на эту ошибку удаления, найти слово?

Опять же восстановите ход рассуждений самостоятельно. Программа есть.

{Алгоритм Сдвиг-И ищет слово *w* в тексте *t* с одной ошибкой удаления.}

```

Procedure SearchShiftDel(w,t:String);
  Var V:Array [chr(0)..chr(255)] Of
LongInt;
  n,m,i:Byte; R1,R2,first:LongInt;
  j:Char;
Begin
  n:=length(t); m:=length(w);
  For j:=chr(0) To chr(255) Do
V[j]:=0;
  For i:=1 To m Do V[w[i]]:=V[w[i]] Or
(1 Shl (m-i));
  R1:=0; R2:=0; first:=1 Shl (m-1);
  For i:=1 To n Do Begin
    R1:=((R1 Shr 1) Or first) And
V[t[i]];
    R2:= ((R1 Shr 1) Or first) Or ((R2
Shr 1) Or first) And V[t[i]];
    If R2 And 1=1 Then Writeln(i-m+2);
  End;
End;

```

11. Расстояние Хемминга между двумя строками одинаковой длины определяется как число позиций, в которых символы не совпадают. Дано слово и некоторое число *d*. Необходимо найти все слова в тексте, удалённые от данного слова не более чем на расстояние *d*.

Если переформулировать задачу, то надо найти слово, при этом в тексте оно содержится не более чем с *d* ошибками замены. Случай с одной ошибкой замены уже рассмотрен. На самом деле, обобщить его не составит большой проблемы. Нужно только ввести по одной дополнительной таблице на каждую ошибку и проводить аналогичные преобразования одной таблицы в другую. Для проверки приводим программу.

{Алгоритм Сдвиг-И ищет слово *w* в тексте *t* с не более чем с *d* ошибками замены.}

Procedure

```

SearchShiftChk(w,t:String;d:Byte);
  Const Maxd=...;
  Var V:Array [chr(0)..chr(255)] Of
LongInt;
  R:Array [0..Maxd] of LongInt;
  n,m,k,i:Byte;
  first,OldR0,tmp:LongInt; j:Char;
Begin
  n:=length(t); m:=length(w);
  For j:=chr(0) To chr(255) Do
V[j]:=0;
  For i:=1 To m Do V[w[i]]:=V[w[i]] Or
(1 Shl (m-i));

  For k:=0 To d Do R[k]:=0;
  first:=1 Shl (m-1);

  For i:=1 To n Do Begin
    OldR0:=R[0];
    R[0]:=((R[0] Shr 1) Or first) And
V[t[i]];
    For k:=1 To d Do Begin
      tmp:=R[k];
      R[k]:=((R[k] Shr 1) And V[t[i]])
Or (OldR0 Shr 1) Or first;
      OldR0:=tmp;
    End;
    If R[d] And 1=1 Then
Writeln(i-m+1);
  End;
End;

```

А если необходимо найти все слова, удалённые от данного слова *ровно* на расстояние *d*? Достаточно будет единственное условие в программе заменить таким: **If** (R[d] **And** 1=1) **And** (R[d-1] **And** 1=0) **Then**...

12. В слове возможны одна замена, одно удаление, одна вставка символа, то есть хоть все три ошибки сразу, но только по одной каждого типа. Какие изменения потребует алгоритм Сдвиг-И при поиске таких слов?

Как и прежде, строится таблица *R1*, отражающая точные совпадения, а по ней *R2*, отражающая совпадения с поправкой на ошибки, но теперь возможны все три вида ошибок. Программа будет следующей:

```

Procedure SearchShiftAll(w,t:String);
  Var V:Array [chr(0)..chr(255)] Of
LongInt;
  n,m,i:Byte;
  R1,R2,pR0,first:LongInt; j:Char;
Begin
  n:=length(t); m:=length(w);
  For j:=chr(0) To chr(255) Do
V[j]:=0;
  For i:=1 To m Do V[w[i]]:=V[w[i]] Or
(1 Shl (m-i));
  R1:=0; R2:=0;

```

```
first:=1 Shl (m-1);
For i:=1 To n Do Begin
  pR1:=R1;
  R1:=((R1 Shr 1) Or first) And
V[t[i]];
  R2:=((R2 Shr 1) Or first) And
V[t[i]]) Or pR1
  Or ((pR1 Shr 1) Or first)
  Or ((R1 Shr 1) Or first);
  If R2 And 1=1 Then Writeln(i); {Вы-
водится конец вхождения слова.}
End;
End;
```

Понятно, что эта программа решает любую и из трёх предыдущих задач. Разберитесь в ней, выясните, какие битовые операции за учёт каких ошибок отвечают...

Кстати, *R2* можно вычислить и экономнее:
R2:=(((R2 Shr 1) Or first) And
V[t[i]]) Or pR1 Or ((pR1 Or R1) Shr
1) Or first).

В программе выбран другой способ лишь для наглядности.

13. Во всех вариантах алгоритма Сдвиг-И на вхождение префиксов указывает 1. Пусть теперь на вхождение указывает 0. Перепишите программы из задач 5, 8, 9, 10, 11 и 12 с учётом такой замены, 1 на 0. Пример подобной модификации у нас уже был, для точного поиска. При этом дополнение единичей оказалось лишним. Быть может, и здесь это приведёт к сокращению числа битовых операций.

И в заключение ещё две задачи на исследование, без ответов и подсказок.

14. Пусть треугольные скобки указывают группу символов, в которых, возможно, есть одна ошибка замены. Например, нам необходимо найти автомобильный номер м815тк. Мы уверены, что буквы вспомнили правильно, а вот в цифрах, может, есть одна ошибка. Тогда запрос запишется как м<815>тк. Как модифицировать алгоритм Сдвиг-И для поиска слов по такому шаблону?

15. Другой стандартный знак в шаблоне – это «решётка» #, на место которой может быть подставлено любое слово. Например, шаблон ab#cd означает, что мы ищем подслово ab, за которым следует что угодно, а затем на любом расстоянии идёт cd. Способен ли алгоритм Сдвиг-И проверить, есть ли в тексте слова, соответствующие таким шаблонам?

М. Ю. Мельникова

СПЕЦИФИКА ВНЕДРЕНИЯ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

В статье рассматриваются вопросы, связанные с внедрением разработанных корпоративных информационных систем фирмами на предприятиях или в организациях.

В условиях рынка всё большее число компаний, предприятий и корпораций осознают преимущества использования корпоративных информационных систем. Современные предприятия (корпорации) имеют сложную структуру, обусловленную многопрофильностью деятельности, территориальной распределённостью подразделений, большим числом кооперативных связей с партнёрами.

Дисциплина «Корпоративные экономические информационные системы» является частью учебного плана студентов по специальности Прикладная информатика (в экономике), поэтому актуальность данной темы является ключевым аспектом, так как на практических занятиях студенты знакомятся не только с готовыми проектами корпоративных информационных систем, а так же прилагают усилия и проектируют и разрабатывают собственные прототипы систем данного класса.

Корпоративная информационная система призвана автоматизировать управление всеми ресурсами и деловыми процессами такого территориально-распределённого предприятия, при этом автоматизация управленческих процессов направлена не просто на сокращение затрат на обработку информации, а на динамическую оптимизацию организационной структуры и деловых процессов или бизнес-процессов, обеспечивающих функционирование предприятия.

Создание, внедрение и эксплуатация корпоративной системы – задача сложная и дорогая, да к тому же требующая серьезной организационной реформы.

Внедрение корпоративной системы воистину сопоставимо с подвигом Геракла. Действительно, для этого требуется обеспечить бесконфликтную поддержку и интеграцию всех бизнес-процессов, объединение изолированных ранее функциональных модулей и доступность данных в пределах всей организации в режиме реального времени. Неудивительно, что программное обеспечение таких производителей, как PeopleSoft, SAP и Oracle, громоздко, дорого и сложно в реализа-

ции. Еще менее удивительно, что такая реализация требует организационной реформы.

Но если компании готовы к высоким затратам на реализацию корпоративной системы и сопутствующим сложностям, то их ожидает значительное увеличение производительности и эффективности работы. Создавая свои внутренние информационные системы, организации предъявляют определенные требования к их функциональности, причем зачастую для выполнения каждой функции приходится разрабатывать отдельную систему, и их число растет с ужасающей скоростью. Так, например в одной из крупных фирм для поддержки выполнения заказов использовалось 58 разных систем. Очевидно, что такая трясина мешает свободному обмену информацией внутри бизнес-подразделений и между ними, порождает массу других проблем, а это, в конечном счете, весьма негативно сказывается на качестве данных. Поэтому для решения таких проблем более эффективно использовать интегрированную систему, созданную на основе одной базы данных. В целом сотрудники получают более эффективный доступ к информации, критически важной для их работы. Некоторые компании утверждают, что им удалось сократить уровень товарных запасов на складах и время обработки более чем на 50%.

Зачастую большинство фирм, занимающихся разработкой, сопровождением и внедрением своих продуктов, опираются, как правило, на неформальный план внедрения. В чём заключается он и на каких принципах базируется?

Планирование проекта

Планирование корпоративной системы может занять практически столько же времени, сколько понадобится на ее реализацию. На определение полного списка требований и поиск приемлемого производителя зачастую уходит несколько месяцев.

Один из ключевых шагов – решение о поэтапном внедрении. Большинство корпоративных систем слишком масштабны, поэтому сложно предусмотреть период опытной эксплуатации, во время которого новая система работает параллельно с унаследованными. С другой стороны, одномоментный ввод в эксплуатацию корпоративной системы может поставить компанию «на грань между жизнью и смертью». При вводе в эксплуатацию корпоративной системы, что обычно называют «продуктивным стартом», организация отключает унаследованные системы и не может вернуться к их использованию.

Чтобы снизить необходимость в параллельной поддержке работы корпоративной и унаследованных систем, компании часто выбирают поэтапный подход – в сущности, серию продуктивных стартов. Каждый продуктивный старт мож-

но использовать либо для ввода в эксплуатацию разных функциональных модулей (например, сначала финансового, затем производственного и кадрового, а потом модуля получения заказов), либо для поочередного внедрения корпоративной системы в филиалах.

Однако поэтапный ввод в действие функциональных модулей не позволяет сразу воспользоваться преимуществами интеграции, которые, собственно, и являются основной причиной внедрения корпоративной системы. Кроме того, он требует создания временных интерфейсов между уже реализованными модулями корпоративной системы и еще действующими унаследованными системами. Поэтапное внедрение системы в филиалах также требует временных интерфейсов для поддержки интеграции и может оказаться проблематичным, если филиалы в значительной степени взаимозависимы.

Еще одно ключевое решение – определить, использовать ли один экземпляр программного обеспечения (одинаковая конфигурация обслуживает все филиалы) или несколько. В последнем случае разные операционные модули будут конфигурировать программное обеспечение так, чтобы оно удовлетворяло их уникальным требованиям, что, в свою очередь, сведет на нет ряд преимуществ интегрированной системы. Если филиалы придерживаются кардинально разных методов работы, в силу чего унификация попросту невозможна, имеет смысл задействовать несколько экземпляров программного обеспечения. Но в большинстве случаев лучше применять конфигурацию единого экземпляра, поскольку такой подход обеспечивает единообразие, расцениваемое как серьезное преимущество корпоративной системы.

Стратегия внедрения проекта

Не менее важное правило – энергичное руководство проектом и тщательное следование методологии. Поскольку при внедрении корпоративной системы организация неизбежно меняется, даже незначительные отклонения от любой из следующих стратегий могут привести к серьезным проблемам и, в конце концов, к краху проекта.

Еще одна опробованная стратегия заключается в создании управляющего комитета. Соответствующая конфигурация требует принятия множества бизнес-решений, а в случае интегрированной системы эти решения взаимозависимы. Управляющий комитет должен анализировать не согласующиеся и взаимоисключающие требования разных бизнес-подразделений и быстро принимать решения, поскольку любое промедление снижает скорость движения в других областях. От состава группы, занимающейся реализацией корпоративной системы, также зависит успех или

провал проекта. Лучше всего выбирать членов группы с учетом перспективы, то есть наряду с техническими консультантами в ее состав должны входить самые лучшие, квалифицированные пользователи из бизнес-подразделений.

Чтобы проект оказался успешным, компании следует тщательно подбирать производителей и консультантов. Компания должна убедиться в том, что консультанты полностью понимают ее бизнес-требования и передают технические знания персоналу, который осуществляет текущую поддержку корпоративной системы.

Минимизация настроек

Конфигурация корпоративной системы – это основной процесс адаптации программного обеспечения к требованиям компании. По возможности, она должна быть единственным средством настройки корпоративной системы. Проектной группе следует избегать более радикальных подходов (таких, как создание интерфейсов для унаследованных систем или разработка дополнительных систем), хотя иногда они неизбежны.

Проектная группа никогда не должна менять базовый код корпоративной системы. Крайне важно использовать только конфигурацию (подход минимизации) и избегать более радикальных способов настройки.

Учёт всевозможных деталей

Так, например, большинство организаций недооценивают время, которое требуется для очистки данных и тестирования процесса преобразований. Как показывает практика внедрения комплексной корпоративной системы «Галактика» в одной из компаний, было потрачено несколько месяцев на очистку данных, а затем еще девять месяцев тестирования преобразований в целом. Даже при наличии повторного тестирования на одном из этапов было некорректно преобразовано около 1 тыс. незавершенных заказов. Для выполнения всей необходимой работы в рамках процесса корректного преобразования данных пришлось остановить последующий ввод системы в эксплуатацию.

Еще одна задача, которую организации далеко не всегда решают адекватно, – создание профилей для аутентификации пользователей. При реализации интегрированной системы каждый, теоретически, может получить доступ к любому полю общей базы данных, а это весьма нежелательно с точки зрения поддержки качества данных. По этой причине сначала следует запретить доступ всем, а затем объяснить каждому пользователю корпоративной системы, какие операции он может выполнять в рамках его служебных обязанностей.

Новые роли и процессы

Крупные корпоративные системы строятся на явных и неявных предположениях о бизнес-процессах и о том, как их выполняют сотрудники.

Одна из категорий таких предположений – рабочие роли, определяющие, как распределяется работа, кто и за какую ее часть отвечает. Например, SAP-технология предполагает конкретные роли сотрудников, которые покупают и планируют материалы, и сотрудников, которые управляют конфигурацией.

Новые системы контроля

С появлением новых процессов и служебных обязанностей становятся необходимыми новые способы управления и контроля за процессами. Корпоративная система собирает информацию о каждой транзакции в режиме реального времени. Поскольку сотрудники не могут выполнять свою работу без соответствующих транзакций корпоративной системы, они автоматически сообщают всей организации (через единую базу данных) о том, что именно они делают. Затем любой сотрудник способен воспользоваться этой информацией для мониторинга и контроля за процессами.

Корпоративная система, объединяющая операционные и финансовые сведения в единой базе данных, также может предоставить информацию о затратах практически в режиме реального времени. Во многих компаниях процедуры использования информации в процессах мониторинга и контроля только формируются, но при большей доступности данных им неизбежно требуются новые формы контроля.

Точность либо своевременность

В корпоративной системе все пользователи применяют одни и те же сведения, но по-прежнему существует противоречие между точностью и актуальностью данных. Рассмотрим этот вопрос в рамках трех ситуаций, связанных со следующими типами заказов клиентов:

- оцениваемые менеджерами по продажам и руководством компании за два года до их выполнения как часть процесса планирования;
- оговоренные за год до их выполнения и, скорее всего, не имеющие полной спецификации;
- поступающие за полгода до выполнения, для которых у организации есть заказанные компоненты и календарь производства.

Раньше первые две категории заказов обрабатывались в отдельной унаследованной системе планирования, и информация о них была доступна только планировщикам. Теперь доступ к соответствующим сведениям оказался открытым для всех, и тот, кто не занимается планированием, часто считает их точными (каковыми они не являются) и предпринимает нецелесообразные действия на основе этого предположения. Подход некоторых организаций – вводить данные в систему лишь тогда, когда они достаточно точны для того, чтобы на них можно было опираться при принятии решений. Однако в таком случае

информация часто оказывалась не вполне своевременной и, в силу этого, менее полезной для планировщиков.

Эффективно использовать новую структуру

Успешная реализация технической системы и проведение организационной реформы – это всего лишь начало. При внедрении корпоративной системы формируется интегрированная внутренняя структура информационных технологий, которую организации могут использовать по крайней мере двумя способами.

Во-первых, она служит основой для дальнейшего совершенствования процессов. Как только компания интегрирует процессы, способы их совершенствования становятся более очевидными и кажутся более выполнимыми.

Во-вторых, инфраструктура создает основу для управления отношениями с поставщиками (SRM) и заказчиками (CRM). Примером управления отношениями с поставщиками является использование корпоративной системы для публикации требований к поставщикам компонентов. Пример управления отношениями с заказчиками – использование корпоративной системы для предоставления сотрудникам, работающим с клиентами, информации, которая необходима для более качественного обслуживания.

В конечном счете, успешный проект внедрения корпоративной системы позволяет увеличить эффективность работы компании. Это возможно даже в том случае, если корпоративная система внедрена с опозданием, с превышением бюджета или не обеспечила всей ожидавшейся функциональности. Напротив, если корпоративная система реализована вовремя и в рамках выделенных средств, но эффективность бизнеса не возросла, можно считать, что проект потерпел неудачу.

Итак, внедрение корпоративных информационных систем базируется изначально на двух идеях. Во-первых, внедрение корпоративной системы – это не просто реализация крупного проекта в сфере информационных технологий, но и масштабная организационная реформа. А организационные преимущества не будут достигнуты, если предприятие сосредоточится исключительно на технических аспектах. Во-вторых, противоречия возникают неизбежно, поскольку внедрение корпоративной системы – проект весьма значительного масштаба, и увязать в его рамках требования всех заинтересованных сторон очень сложно. Нужно осознавать, что эти противоречия можно лишь уменьшать и контролировать, не рассчитывая на их кардинальное решение. А раз так, они оказываются постоянными сложностями при управлении интегрированными организациями, использующими корпоративные системы.

МЕТОДИЧЕСКИЕ АСПЕКТЫ ИЗУЧЕНИЯ СТУДЕНТАМИ ДИСЦИПЛИНЫ «КОРПОРАТИВНЫЕ ЭКОНОМИЧЕСКИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ»

В статье подробно рассмотрено содержание дисциплины «Корпоративные экономические информационные системы», в процессе изучения которой изучаются аспекты, связанные с их разработкой и проектированием, а также используется методическое программное средство для проведения практических занятий – учебная версия автоматизированного комплекса «Галактика».

Учебная дисциплина «Корпоративные экономические информационные системы» – часть общего учебного плана профессиональной подготовки студентов по специальности Прикладная информатика (в экономике).

Актуальность данной учебной дисциплины обусловлена тем, что знания и навыки, полученные в результате ее изучения, могут быть применены:

- при разработке корпоративных информационных систем в банках, производственных и торговых компаниях, консалтинговых и страховых фирмах, туристических агентствах и др.;
- при внедрении типовых корпоративных информационных систем, таких, как «1С:ПРЕДПРИЯТИЕ», «ГАЛАКТИКА», ВААН IV и других в различных фирмах и компаниях;
- при проведении реорганизации бизнеса на основании корпоративных информационных технологий;
- при оказании консультационных услуг в области применения корпоративных информационных систем различного назначения.

В Государственном стандарте высшего профессионального образования данная дисциплина находится в разделе «Дисциплины специализации в области» и соответственно её содержание достаточно вариативно в разных вузах, так, например, в Пензенской технологической академии эта дисциплина рассматривается не как самостоятельная, а её аспекты охватываются и рассматриваются в курсе другой дисциплины «Предметно-ориентированные экономические информационные системы».

В рамках данного курса должна соответственно рассматриваться как пример некоторая комплексная автоматизированная система по управлению как предприятием, так и корпорацией. Так плодотворно ведёт сотрудничество с высшими учебными заведениями и предлагает учебные вер-

МЕЛЬНИКОВА Марина Юрьевна – старший преподаватель филиала ВятГГУ в г. Кирово-Чепецке
© Мельникова М. Ю., 2008

сии систем данного типа зарекомендовавшая себя на информационном рынке корпорация «Галактика». Программа под эгидой «Галактика и вузы» предусматривает сотрудничество с данной организацией, вузам предоставляется возможность установки учебной версии данной программы и целый комплекс методических материалов по её изучению и освоению.

Проанализировав учебные программы различных вузов по данной дисциплине, можно сделать вывод о недостаточно глубокой проработанности содержания, что связано с новизной курса.

В качестве основы для дальнейшего совершенствования можно предложить апробированное как на факультете информатики ВятГГУ, так и филиале ВятГГУ в г. Кирово-Чепецке содержание курса «Корпоративные экономические информационные системы».

Конечной целью данного курса является как формирование у будущих специалистов представления об особенностях структуры корпоративных экономических систем, их проектировании, так и получение практических навыков по их созданию и использованию конкретной корпоративной экономической информационной системы.

В ходе изучения дисциплины были поставлены следующие задачи:

- изучить историю развития корпоративных экономических информационных систем;
- изучить классификацию экономических управляющих систем;
- изучить основные методологии создания экономических управляющих систем: MRP, CRP, MRPII, ERP;
- изучить современное состояние рынка экономических управляющих систем и тенденции его развития;
- изучить структуру и общую схему функционирования корпоративной экономической информационной системы;
- изучить особенности моделирования и проектирования корпоративной информационной системы (КИС) с помощью инструментальных средств проектирования систем данного класса;

В результате изучения дисциплины студенты должны:

1) иметь представление о корпоративных экономических информационных системах, их роли в качестве средства автоматизации предприятия или организации на всех уровнях управления, от бухгалтерских подразделений до кадровых служб, отделов логистики, управления производства в целом; структуре и подходах к проектированию и моделированию корпоративных информационных систем; вопросах администрирования систем данного класса;

2) уметь выбрать технологию, подходы и инструментальные средства создания корпоратив-

ной экономической системы на основе знаний об основных параметрах современных средств разработки КИС;

3) владеть навыками использования инструментальных средств в процессе проектирования корпоративных информационных систем;

4) работать с основными модулями системы «Галактика», такими, как настройка, управление персоналом, логистика, финансово-расчётные операции, типовые расчётные операции, касса, управление производством; заполнять каталоги, вводить данные и сведения о клиентах, а также перечень необходимых документов, создавать необходимые в процессе работы отчётные документы и формы.

Изучение данной дисциплины по видам учебной работы осуществляется в следующем объёме часов:

- общая трудоёмкость часов (по ГОС ВПО) – 84;
- аудиторные занятия – 42;
- лекции – 14;
- практические занятия (семинары) – 28;
- самостоятельная работа – 42.

Основным видом контроля по окончании изучения данной дисциплины является зачёт.

Рассмотрим основные темы данного курса. Лекционный материал, в соответствии с поставленными задачами, должен включать в себя следующие темы:

1. Теоретические основы Корпоративных экономических информационных систем. Корпоративные экономические информационные системы. Роль корпоративных информационных систем в автоматизации процессов управления. Классификация корпоративных экономических информационных систем.

2. Технология проектирования Корпоративных экономических информационных систем. Требования, предъявляемые к архитектуре Корпоративных экономических информационных систем. Особенности проектирования клиент-серверных систем.

3. Инструментальные средства для разработки Корпоративных экономических информационных систем. Методологии разработки Корпоративных экономических информационных систем – MRP, MRPII.

4. Методология ERP разработки Корпоративных экономических информационных систем. Новые направления развития Корпоративных экономических информационных систем.

5. Практическое использование Корпоративных экономических информационных систем. Анализ и тенденции развития рынка корпоративных информационных систем.

6. Проблемы внедрения Корпоративных экономических информационных систем в России.

КИС «Галактика». Структура базовой концепции КИС «Галактика».

7. Контуры и модули КИС «Галактика». Краткий анализ и особенности, функциональные особенности. Новые направления системы «Галактика».

Большинство этих тем хорошо освещены в различных пособиях и в Интернете. Однако, как показал анализ программ различных вузов, возникает много сложностей. Как было отмечено ранее, практикум по данной дисциплине состоит из двух частей: первая часть посвящена изучению основных приёмов разработки, проектирования и администрирования КИС с учётом выбора инструментальных средств, а вторая часть – знакомство с конкретной КИС на основе учебного примера.

Для реализации первой части практикума нами был взят за основу подход коллег из МЭСИ, который заключается в поэтапной разработке коллективной КИС на учебном примере. Студенты в ходе работы чётко формулируют постановку задачи, производят декомпозицию системы на подсистемы, выбирают и используют по назначению case – средства, знакомятся с такими понятиями, как словарь данных и метаинформация, целенаправленно используют инструментальные средства разработки и поддержания КИС, применяют к ней приёмы администрирования. В качестве учебного примера студенты занимаются разработкой и построением КИС предприятия по оптовой торговле кабелями, на основе выданного типового технического задания и сформулированных требований, что должна обеспечивать система. В качестве поддержки программного обеспечения могут быть выбраны: BPWIN, ERWIN, MS SQL SERVER, Delphi, ADABAS.

Для реализации второй части практикума по данной дисциплине на факультете информатики ВятГГУ в качестве методического программного средства для проведения второй части практических занятий была выбрана учебная версия автоматизированной системы по управлению предприятием «Галактика» от корпорации «Галактика». «Галактика» – российско-белорусская корпорация, одна из наиболее популярных отечественных КИС класса ERP, внедрена более чем на 7000 предприятиях России и других стран СНГ, в том числе в нефтяной компании «СИДАНКО», ОАО «Русский продукт», ОАО «Ангарская нефтехимическая компания», ОАО «Саратовские обои», ООО «Сыктывкарский фанерный завод». «Галактика» построена в архитектуре клиент-сервер и является интероперабельной системой, способной функционировать в связке с различными серверами баз данных, поддерживается Oracle, /MS SQL. Галактика

позиционируется как система для предприятий любой сферы экономической деятельности: средних, крупных, в том числе как без территориально распределённой инфраструктуры, так и со сложной распределённой инфраструктурой. Продукт тиражно-заказной, настраиваемый на конкретное предприятие с помощью большого числа настроек. Именно данной корпорацией создан спектр решений для предприятий самых различных отраслей – нефтегазовой, химической, пищевой промышленности, связи и телекоммуникаций, торговли, энергетики, металлургии. Имеются примеры удачных внедрений в бюджетной сфере, а также на предприятиях научно-технического сервиса, в высших учебных заведениях, в органах государственного и муниципального управления.

Некоторые разработки корпорации «Галактика» для сектора ИТ-рынка – «недорогие решения для аналитической обработки информации» – получили известность на североамериканском континенте.

Особым шагом по сближению с вузами заслуживает образовательная политика корпорации «Галактика», которая в рамках программы «Галактика и вузы» предоставляет учебным заведениям высшего и среднего профессионального образования, имеющим необходимые условия, полнофункциональные демоверсии системы для внедрения в учебный процесс.

Всё это и предопределило выбор Корпоративной экономической информационной системы «Галактика» в качестве базовой системы для второй части лабораторного практикума.

Знакомство с системой «Галактика» в ходе лабораторного практикума производится на учебном примере, где есть уже готовая база данных с виртуальным предприятием, набором документов и частью выполненных операций. В качестве методического учебного пособия по данной части практикума выступает разработанное и апробированное в ряде вузов, в том числе и в МЭСИ, именно под учебную базу данных, которая поставляется вместе с демоверсией системы «Галактика» – лабораторный практикум на базе системы «Галактика» под ред. Е. П. Бочаровой, А. И. Колдина.

Переходя к рассмотрению тем практикума, нельзя обойти его цели, каковыми являются следующие:

– знакомство студентов с установкой, настройкой и функционированием модулей, составляющих ядро системы «Галактика»;

– выполнение операций, связанных с кадровой деятельностью, в том числе это операции по работе со штатным расписанием, вакантными ставками, приём на работу сотрудника, переводы, увольнение, выполнение отчётов;

– выполнение операций по закупке и сбыту продукции, реализации продукции через прайс-лист;

– изучение финансовых и бухгалтерских операций, проводки, формирование счетов – фактур, платёжных поручений, выполнение отчётов, авансовые отчёты и учёт кассовых операций;

– изучение процесса заказа продукции на конкретный срок выполнения, формирование технического процесса на изготовление заказа, изготовление и отпуск заказа покупателю.

В соответствии с данными целями темами практикума являются следующие:

1. Установка и настройка КИС «Галактика».

2. Контур «Управление персоналом». Штатное расписание, приём на работу, переводы сотрудников из подразделений, увольнение, тарифная сетка. Индивидуальная карточка сотрудника. Отчёты.

3. Контур «Логистика». Функции. Особенности функции, структура. Модуль учёта снабжения по закупке товаров, ценовая политика, прайс-листы. Модуль учёта сбытом. Документы – основания, накладные, заказы.

4. Контур «Бухгалтерский учёт». Функции контура. Модуль «Финансовые расчётные операции» (ФРО). Типовые хозяйственные операции (ТХО). Примеры ведения операций, счета-фактуры, факты оплаты.

5. Контур «Управление производством». Основные функции контура. Изготовление продукции. Формирование производственных запасов. Особенности учёта производства продукции.

По данному курсу разработан дополнительный методический материал в виде инструкций, то есть методических указаний. Данный материал содержит практические работы по освоению модулей: Управление персоналом, Сбыт, Снабжение, Складской учёт, ХозОперации, ФРО – Финансово-расчётные операции, Управление производством.

Управление персоналом – это модуль, который позволяет конкретно просмотреть одну из функций управления предприятием – на уровне кадровой службы. Студенты знакомятся с такими формами работы, как автоматизированное ведение личных дел сотрудников, формирование и управление штатным расписанием, наличием и внесением вакантных ставок, приём на работу сотрудников, переводы, увольнение сотрудников, учёт рабочего времени, а также получение отчётов по кадровой информации.

Управление снабжением, сбытом и складской учёт – это три модуля, которые образуют один из контуров системы «Галактика» – Логистику. Студенты, знакомясь с ними, изучают всю цепочку методологии MRP, моделируют процессы закупки сырья, особенности заключения договоров с поставщиками и покупателями. Так, в мо-

дуле Управление снабжением студенты осуществляют: ввод документов-оснований на закупку, ввод накладных на приход и возврат по рекламации, актов на работы и услуги; выписки доверенностей, формируют бартерные договоры, формируют ордера по накладным, платёжные документы по закупкам и отчёты по закупкам. Работая в модуле Управление сбытом, студенты знакомятся со следующими операциями: выписка документов-оснований на продажу, оформление договоров; оформление накладных на отпуск и возврат по рекламациям, актов на услуги и работы, бартерные договоры, формирование платёжных поручений, прайс-листов, а также формирование отчётов по реализации. При работе с модулем Складской учёт студенты знакомятся с такими операциями, как оформление природных или расходных складских ордеров; оформление накладных на внутреннее перемещение; расход текущих остатков материальных ценностей и формирование отчётов по движению и наличию материальных ценностей.

Финансово-расчётные операции и ХозОперации – это два модуля, основу которых составляют бухгалтерские расчёты. При работе с данными модулями чётко прослеживается взаимосвязь с предыдущими модулями контура Логистика. Так, в модуле ХозОперации студенты знакомятся с новыми для них понятиями и операциями: типовая хозяйственная операция, финпроводка, бухпроводка, журнал хозяйственных операций. А модуль Финансово-расчётных операций регистрирует движение финансовых средств и тесно связан с операциями из модуля ХозОперации. При работе с данными модулями студенты охватывают весь спектр учётных задач по экономическому профилю.

И в заключение цикла практических работ студенты знакомятся с контуром Управление производством, в частности рассматривают вопросы объёмно-календарного планировании хозяйственной деятельности, основу которого составляет методология MPS, учёт материальных ценностей и услуг в производстве, ведение спецификаций. Изучение этого контура является завершающим и охватывает вопросы, решаемые студентами при рассмотрении предыдущих модулей.

Таким образом, по окончании изучения дисциплины «Корпоративные экономические информационные системы», имеющей вышеописанное содержание, у студентов будут сформированы теоретические знания о корпоративных экономических системах и практические навыки по особенностям их проектирования и разработки и по использованию на конкретном примере – автоматизированная информационная система по управлению «Галактика».

М. А. Морданов

О РАЗЛОЖЕНИИ НАТУРАЛЬНЫХ ЧИСЕЛ НА СУММУ ПРОСТОГО И ДВУХ КВАДРАТОВ

Приводится экспериментальная оценка времени работы алгоритма представления натурального числа в виде суммы простого числа и двух квадратов целых чисел.

В 1959 г. Юрий Владимирович Линник доказал, что каждое достаточно большое натуральное число n может быть представлено в виде суммы простого числа и двух квадратов целых чисел, т. е. в виде $n = p + k^2 + l^2$. Рассмотрим алгоритм нахождения такого разложения.

Функция проверки простоты числа имеет вид:

```
function check(z:int64):boolean;
var i,ii:int64;
    h:boolean;
begin
    i:=2;h:=true;
    ii:=round(sqrt(z*1.0))+1;
    while (i<ii) and h do
        begin
            if (z mod i=0) then h:=false;
            inc(i);
        end;
        check:=h;
    end;
```

Функция трудоемка (по времени), ибо это не что иное, как проверка всех натуральных чисел из интервала от 2 до \sqrt{n} на предмет того, являются ли они делителями числа n . Чтобы уменьшить время работы, необходимо обращаться к функции check как можно реже и с минимально возможными числами. Для этого начинаем поиск разложения с максимально возможными квадратами и постепенно уменьшаем их, пока ответ не будет найден (т. е. число $rez = n - i*i - j*j$ не будет простым).

```
procedure main;
var i,j,rez:int64;
    k:boolean;
begin
    k:=false; i:=round(sqrt(n*1.0));
    j:=i;
    while (i>=2) and (not k) do
        begin
            if n-i*i>0 then j:=trunc(sqrt( (
n-(i*i) *1.0) )) else j:=0;
            while (j>=2) and (not k) do
                begin
```

```
                rez:=n-i*i-j*j;
                if rez>1 then
                    if check(rez) then k:=true;
                dec(j);
            end;
        dec(i);
    end;
    inc(i);inc(j);
    if k then
        writeln(n,'=',rez,'+',i,'*',i,'+',j,'*',j);
    end;
```

При проведении эксперимента для работы с многоразрядными числами кроме приведенной реализации использовался модуль Long Arithmetic. Его описание не приводится, ибо он не что иное, как реализация арифметических операций, описанных в работе С. М. Окулова (Окулов С. М. Программирование в алгоритмах. М.: БИНОМ. Лаборатория знаний, 2002).

Количество разрядов в числе	Среднее время работы	
	Int64	Long Arithmetic
10	0 сек.	≈ 0,5 сек.
15	0 сек.	≈ 12,2 сек.
19	0 сек.	≈ 2*10 ³ сек.
30	–	6,9*10 ⁸ сек. ≈ 22 года
50	–	1,8*10 ¹⁷ сек. ≈ 5,7*10 ⁹ лет
100	–	2,2*10 ³⁸ сек. ≈ 6,9*10 ³⁰ лет
200	–	3,2*10 ⁸⁰ сек. ≈ 9,9*10 ⁷² лет
500	–	9,9*10 ²⁰⁶ сек. ≈ 3,1*10 ¹⁹⁹ лет
1000	–	6,2*10 ⁴¹⁷ сек. ≈ 1,9*10 ⁴¹⁰ лет

Из таблицы видно, что при использовании типа Int64 для хранения чисел разложение проходит быстро, но в этом случае есть один недостаток – максимальное число, которое может быть представлено в этом типе – 9 223 372 036 854 775 807 (19 разрядов).

Если же использовать для вычислений модуль Long Arithmetic, то в программе будет возможно раскладывать и числа, состоящие из тысяч разрядов. Но и в этом случае есть серьезный недостаток – время, необходимое компьютеру для выполнения вычислений. Согласно таблице одна и та же программа при использовании Int64 работает меньше секунды, а при использовании Long Arithmetic – несколько минут.

Таким образом, ни один из двух алгоритмов не дал приемлемых результатов.

С. М. Окулов

ЕЩЕ РАЗ ПРО ИНФОРМАТИКУ

В статье дается обоснование фундаментальных основ школьного курса информатики.

Пиши, не мудрствуя лукаво.
А. С. Пушкин

Взгляд на информатику как предмет в общеобразовательной школе за немногим более чем двадцатилетнее развитие обладал некоей аномальной изменчивостью. С одной стороны, эта ситуация обусловлена объективными причинами (динамика развития данной области действительности и т. д.), а с другой – все ли ладно в Датском королевстве? Автор в ряде работ [1] пытается обосновать те фундаментальные понятия, на которых должен, с его точки зрения, строиться школьный курс информатики. В данной статье, несколько с других позиций, обосновывается тот же взгляд на суть предмета обсуждения. Забудем на время обо всех существующих концепциях, позициях, учебниках [2] и т. д. и, опираясь только на более чем тридцатилетний опыт работы как в разработке реальных систем обработки информации, так и в образовательной информатике, выстроим цепь логических рассуждений в обоснование нашей позиции.

Зададим себе первый наивный вопрос – «что такое информатика»? И оказывается, что дать простой и ясный ответ не очень просто. «Лужайки», например, под названием «математика», «физика» имеют достаточно четкие границы, а в данном случае? Конечно, допустимо сказать что-то, связанное с информационными процессами, с обработкой информации и т. д., но в этом случае одно понятие определяется через другие, образуется длинная понятийная цепочка, часто тавтологическая, и не очень ясно (прагматику), где начало и где конец этого понятийного многообразия. Упростим (или усложним) первоначальный вопрос – «возможна ли информатика без компьютера»? Если ответить «да», то границы «лужайки» становятся еще более неопределенными, понятие «информатика» переходит в разряд сверхоткрытых, и впору задать вопрос – «а что не есть информатика»? Предположим, что мы сказали «нет». В этом случае появляется граница, пусть пока только пунктирная. Но есть новое понятие – «компьютер», и тут же возникают вопросы, как минимум, о взаимосвязи по-

нятий и о том, что есть компьютер – это объект изучения в информатике или средство (инструмент) изучения информатики?

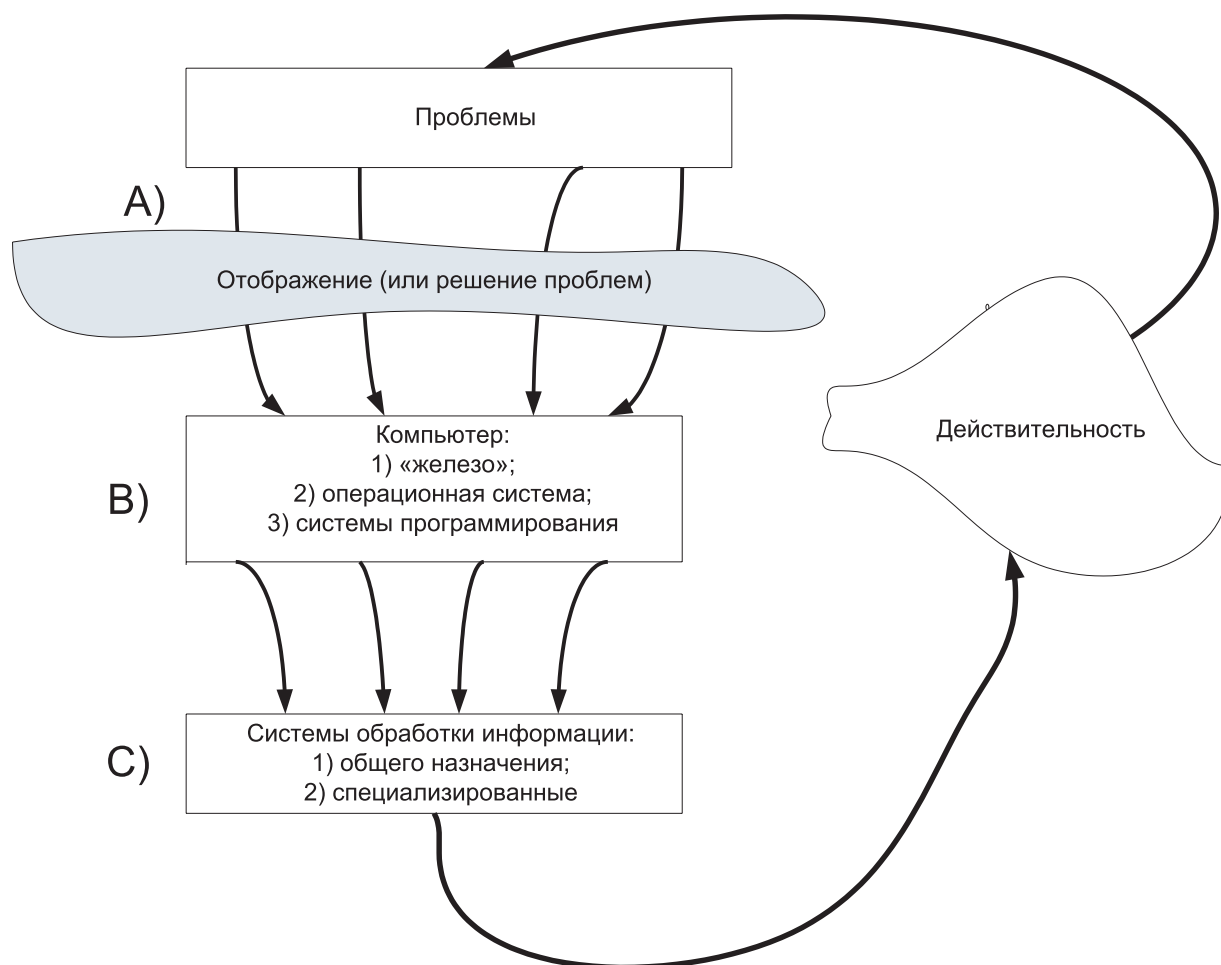
Определимся с условными границами понятия «компьютер». Самый низший уровень – «железо» (hardware). Безусловно, понимание того, как функционирует аппаратный уровень, необходимо. Хорошо, например, знание того, что для реализации всех возможностей этого уровня компьютера ему достаточно уметь выполнять только операции инверсии, сдвига на один разряд и прибавление единицы. Все остальное конструируется из этих примитивов. Но полезность компьютера при таком ограничении меньше, чем у молотка, последним хотя бы гвозди можно забивать. Таким образом, в понятие «компьютер» следует «вложить», как минимум, операционную систему (её основные функциональные возможности и структуру) и простейшую систему программирования. После этих вложений компьютер уже выполняет свое основное предназначение – вычисляет и перебирает, то есть произошло то чудесное оживление, которое делает компьютер универсальным устройством, универсальным вычислителем и переборщиком вариантов. Итак, граница очерчена, и просматривается, в каком объеме (минимальном) компьютер является объектом изучения. А дальше? Дальше два пути. Первый – расширяя понятие компьютер, добавляя к нему уже некий готовый функционал, например графический редактор, мы можем изучать его и утверждать о том, что изучается информатика – редакторы же связаны с компьютером. Второй – ответить на вопрос, что произошло в цепочке от потребности человека рисовать к появлению у компьютера такого функционала, удовлетворяющего этой потребности. Оставим на время второй путь и закончим рассуждения о нашем понимании компьютера. Повторим мысль о том, что возможности компьютера ограничены. Простейший пример (чуть-чуть абсурдный). Любой из нас, если потребуются, выполнит операцию 10^{1000} , выписывая соответствующее количество нулей. Компьютер, без специальных ухищрений, этого не сделает. Естественно, что возможности компьютера увеличиваются со временем – сравним компьютеры двадцатилетней давности и настоящие. Расширение возможностей позволяет решать все новые проблемы и совершенствовать решения старых. Расширяются возможности компьютера, но ограниченность компьютера остается [3].

Однако вернемся к нашим рассуждениям об информатике и продолжим их с помощью примера. Человек и до появления компьютера работал с текстом. В зависимости от вида деятельности эта работа была различна (писатель, редактор и т. д.). Другими словами, есть потребность

работы с текстом, и есть проблема работы с текстом. С появлением компьютера человек «углядел», что последний может быть полезен для удовлетворения этой потребности, и разработал вначале простейший текстовый редактор, затем текстовый процессор с огромными, если сравнивать с первыми версиями, возможностями. Отметим, что в данном случае (и в других) совершенство инструмента, а текстовый процессор является инструментом, оказывает обратное влияние на процесс работы с текстом (обратная связь). Но не это главное, а в продолжение нашего наивного вопроса – «где здесь информатика»? Итак, в самом общем виде (см. рисунок), есть действительность, есть проблемы этой действительности, есть отображение этих проблем в некие их решения, использующие компьютер, в результате которого получают некие продукты (так называемые новые информационные технологии – это лишь небольшая толика этих продуктов), удовлетворяющие потребностям этой действительности.

Примеры можно продолжить. Еще один для наглядности, связанный, в частности, и с обработкой текстов. В молекулярной биологии и в других областях знаний есть проблема поиска в больших объемах текстовых данных входящих в нее образцов. Другими словами, прообразом этой проблемы является элементарная задача поиска слова в тексте. Постоянное наращивание возможностей уровня *B* (см. рисунок) не решало запросы действительности. И в последние 25 лет сформировался новый подраздел информатики – «методы обработки строк».

Итак, вернемся к первоначальному вопросу. Информатика – это уровни *A*, *B*, *C* или *A*, *B* (см. рисунок)? Мы оставляем только самые логически разумные сочетания. Отметим, что при положительном ответе на вопрос о взаимосвязи информатики и компьютера («возможна ли информатика без компьютера») в нашей схеме, а она применима не только к информатике, появляется еще один, трудно определимый, «прямоугольник». Далее можно было бы рассуждать сле-



дующим образом. «Взять» образовательный стандарт школьного курса, соотнести его положения с приведенной схемой, а затем на основе некоего анализа с учетом привнесенных извне положений сделать определенные выводы и ответить на поставленный вопрос. Очевидно, что в зависимости от «привнесенных извне положений» могут получиться различные результаты, вплоть до того, что мы рассуждаем вовсе не о том, или до появления некоей новой информатики, например «асоциальной». Но попробуем пойти другим путем – спроецируем ситуацию на высшую школу, поймем, как в ней отвечают на поставленный вопрос, и сделаем обратную проекцию на общеобразовательную школу. Операции проецирования, как прямого, так и обратного, не приводят к искажению предмета анализа, ибо речь идет об одном и том же, просто в разных объемах.

«Лужайка» высшей школы также поражает своим многообразием направленности подготовки специалистов по информатике, но имеет более определенные границы. Вычленим три основополагающих российских образовательных стандарта.

Специальность 010200 Прикладная математика и информатика, квалификация – математик, системный программист. Естественно, что осуществляется фундаментальная подготовка по математике. В блоке общеобразовательных дисциплин информатика представлена курсами «Языки программирования и методы трансляции», «Системное и прикладное программное обеспечение» и «Базы данных и экспертные системы». Остальные составляющие подготовки зависят от выбранной в вузе специализации, но в основном, если последняя связана с информатикой, рассматриваются вопросы программирования и функционирования операционных систем разного назначения. Таким образом, согласно нашей схеме (см. рисунок), это уровни *A* и *B*. При этом уровень *A* трактуется чисто с математических позиций. Согласно логике стандарта подготовка по математике обеспечивает реализацию отображения *A*, с чем, конечно, трудно согласиться. Математическая составляющая в *A* безусловно есть, в некоторых проблемах она имеет принципиальное значение, но сведение методов решения проблем только к ней не соответствует реальному состоянию дел. И здесь в информатике имеет место парадоксальная ситуация, образно характеризующаяся словами – «сапожник без сапог». Специалисты по информатике создают системы обработки информации различного назначения, в частности автоматизирующие различные виды деятельности, но собственная деятельность по отображению проблемы в некий программный продукт (уровень *A*) соответствующей поддерж-

ки практически не имеет. Структурное, объектно-ориентированное проектирование, а также различные технологии типа CASE (Computer Aided Software Engineering) или RAD (Rapid Application Development) лишь частично решают эту проблему, причем последние относят скорее к уровню *C*, а не *A*. Естественно, что эта часть проблематики уровня *A* слабо представлена в образовательных стандартах или никоим образом не затрагивается.

Специальность 351400 Прикладная информатика (по областям), квалификация – информатик – (квалификация в области). Для определенности будем считать областью – экономику (самый распространенный вариант). Общеобразовательный блок представлен стандартным набором курсов: вычислительные системы, сети и телекоммуникации; информационные системы; базы данных; высокоуровневые методы информатики и программирования; операционные системы, среды и оболочки; информационные технологии. К специальным дисциплинам относятся: проектирование информационных систем; интеллектуальные информационные системы; информационная безопасность, а также информационные системы в бухгалтерском учете и аудите, в банковском деле и т. д. Итак, в стандарте четко представлен уровень *B* (его программная составляющая) и та часть уровня *C*, которая относится к выбранной отрасли. Уровень *A* дается скорее в технологическом плане, связанном с общими вопросами проектирования информационных систем.

Специальность 654600 Информатика и вычислительная техника, квалификация – инженер. Общеобразовательный блок с точностью до деталей совпадает с аналогичным блоком предыдущей специальности, а в специальных дисциплинах (в зависимости от специализации) более детально представлен уровень *B*, например, курсами: теория вычислительных процессов; архитектура вычислительных систем и т. д. Уровень *A* отождествляется с программированием.

С некоторыми оговорками можно считать, что подготовка специалистов по уровню *A* осуществляется в рамках специальности «прикладная математика и информатика» (квалификация выпускника – математик, системный программист); уровню *B* – «информатика и вычислительная техника» (квалификация выпускника – инженер); уровню *C* – «прикладная информатика (по отраслям)» (квалификация выпускника, например, информатик-экономист). Итак, если уровни *B* и *C* достаточно четко очерчиваются, то с уровнем *A* несколько сложнее. Так или иначе в нем присутствует то, что обозначают понятием «программирование». Но само по себе программирование, как знание систем программирования и умения что-то на них записывать, пусть даже в форма-

лизованном виде, не решает проблемы отображения A (см. рисунок), как не решает ее и чисто математическая составляющая подготовки. Приведем примеры в подкрепление точки зрения «за» и «против». Новый раздел информатики – методы обработки строк – имеет огромное значение для многочисленных приложений. Его математическая составляющая минимальна. Другой раздел – методы защиты информации. Без знания математики не достигается даже первичное понимание сути пионерской работы У. Диффи и М. Э. Хеллмана. Но в том и другом случае ключом является понятие «алгоритм» и нечто (вспомним ситуацию «сапожника без сапог»), что условно обозначим как искусство перевода решения проблемы (оно не сводится к алгоритму) на язык, понятный на уровне B .

Итак, подведем определенную черту под вышеизложенным. Уровни B и C прописываются, чего не скажешь об уровне A . Общепринятого понимания уровня A нет. Различные его трактовки опираются на не очень понятную аксиоматику. Использование понятия «искусство» вынужденное. Человек, когда его не удовлетворяет определенный язык как инструмент формализации, создает некий метаязык. Но в данном случае все попытки создания его в информатике трудно считать успешными. Популярное в настоящее время объектно-ориентированное проектирование вряд ли в полном объеме решает проблему.

А в школе, в школьном образовании, делается попытка интеграции уровней в единое целое в рамках ограниченного времени на изучение. И естественно, возникает вопрос – возможно ли решение данной проблемы? Собрать в малом (по времени) большое, и не просто большое, а постоянно растущее, особенно на уровне C , целое.

Но помимо теоретической возможности такой интеграции есть еще один вопрос – а надо ли это делать? Если следовать логике одного из государственных мужей, утверждающего, что основной задачей школы является «подготовка грамотного пользователя технологий», естественно с приставкой – «новых информационных», то, конечно, «да», и основной акцент должен быть сделан на уровне C . Изучение уровня C имеет еще одну особенность, которую просто необходимо отметить. Научно доказано [4], что существующие методики изучения уровня C , во всяком случае те, которые представлены в современных учебниках, не развивают или очень слабо развивают интеллектуальный потенциал школьника. Мы не говорим, что не учат, это несколько разные срезы явления. Сумма запоминаемого фактографического материала огромна, так же как и навыки манипулирования «мышкой». Простой пример. Текстовый процессор Word известен всем, и все могут, так или иначе, работать с ним. Однако на вопрос о

том, как он, текстовый процессор, выполняет с такой скоростью поиск как грамматических ошибок, так и многих стилистических погрешностей, ответят немногие.

В заключение необходимо сказать о той огромной ответственности современного общества перед детьми за их будущее. Любое достижение цивилизации помимо позитивной несет в себе и негативную составляющую. Компьютер – это хорошо или плохо для образования, для развития школьника? Современный школьник не знает таблицу умножения – зачем? есть калькулятор. Так в чем же заключается негативная составляющая компьютера? Любому педагогу, связанному с информатикой, приходилось и, вероятно, не один раз отвечать на вопрос родителей – «мы купили ребенку компьютер в надежде на то, что он повлияет на его развитие, а ребенок день и ночь играет – что делать»? Приведем почти абсурдный пример из другой области для того, чтобы пояснить эту мысль. Потребность человека убивать себе подобного известна. Человек придумал атомную бомбу, чтобы эффективно убивать, и она стала предметом общего пользования. Что произойдет – предсказать нетрудно. Так что убьет компьютер? Компьютер, при его бездумном использовании, убьет творческое начало в человеке и сделает из него нечто, как начинающееся, так и заканчивающееся примитивным потреблением кем-то созданных услуг. Пусть это несколько гипертрофированное утверждение, но это лучше, чем очередная эйфория, связанная с всеобщей компьютеризацией.

Примечания

1. Окулов, С. М. Информатика: развитие интеллекта школьника [Текст] : монография / С. М. Окулов. 2-е изд. М.: БИНОМ. Лаборатория знаний, 2008; Окулов, С. М. О фундаментальных основах информатики (полемические заметки) [Текст] / С. М. Окулов // Информатика и образование. 2005. № 1. С. 26–31.

2. Автор не играет в игру – «кто прав, кто виноват», «что такое хорошо, что такое плохо», а рассматривает ситуации как данность, сложившуюся в ходе исторического развития явления, естественно, в результате объективных и субъективных факторов.

3. У автора, в его педагогической деятельности, есть небольшой курс под условным названием «О тупости компьютера», в котором, в частности, показывается ограниченность его возможностей. Он, естественно, основан на универсальном методе решения задач – переборе, а его «краугольным камнем» является процесс поиска человеком эффективных способов (механизмов) отображения реальных проблем на уровень компьютера.

4. Имеется в виду исследование Т. Н. Суворовой (Совершенствование методики изучения информационных технологий в школьном курсе информатики: автореф. дис. ... канд. пед. наук. М., 2007), выполненное под руководством А. А. Кузнецова.

Р. А. Семенов

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ПРОБЛЕМЫ ГОЛЬДБАХА – ЭЙЛЕРА

В статье проанализирована одна из старейших нерешённых задач теории чисел – проблема Гольдбаха – Эйлера – и рассмотрены различные подходы к программной реализации её алгоритма.

Математика является одной из древнейших наук, которая дала начало своему развитию ещё в первобытные времена. В связи с этим довольно часто приходится слышать мнение, что в ней в основном всё уже известно, что времена открытий в этой науке давно прошли, а теперь остаётся только изучать теоремы, названные именами учёных прошлых веков. Но в действительности это далеко не так. Яркий пример тому – утверждение, называемое проблемой Гольдбаха – Эйлера. Несмотря на свою предельно простую формулировку, решение этой задачи до сих пор никем не было найдено. Её суть заключается в том, что каждое чётное число, большее двух, представимо в виде суммы двух простых чисел.

Пусть n – чётное число ($n > 2$), причём $n = p + q$, где p и q являются простыми ($p \leq q$); $r(n)$ – число возможных выражений, удовлетворяющих предыдущему условию. Тогда количество способов записи числа n суммой двух простых величин с учётом их порядка составит

$R(n) = 2 \cdot r(n)$, где $\frac{n}{2}$ не простое, и

$R(n) = 2 \cdot r(n) - 1$ в случае, когда $\frac{n}{2}$ является простым числом. Проблема Гольдбаха – Эйлера гласит, что $r(n) > 0$ или $R(n) > 0$ (эти неравенства обладают взаимно-однозначным соответствием) для любого чётного n , большего двух.

Первоначально Гольдбахом была выдвинута менее сильная гипотеза, состоящая в возможности представления любого нечётного числа, большего пяти, суммой трёх простых. В таком случае нечётному $n > 5$ будет соответствовать $r(n)$ выражений вида $n = p + q + s$, где p, q и s – простые числа. Тогда, согласно рассматриваемой проблеме, $r(n) > 0$ для каждого нечётного числа n , большего пяти.

Очевидно, что истинность первого утверждения влечёт за собой достоверность второго. Действительно, представив любое нечётное число, большее пяти, в виде $2n + 1$, можно записать его как $3 + 2 \cdot (n - 1)$, причём $2 \cdot (n - 1) > 2$. Тогда из возможности разложения $2 \cdot (n - 1) = p + q$

следует допустимость равенства $3 + 2 \cdot (n - 1) = p + q + s$, где p, q и s – простые, причём $s = 3$.

Ввиду того, что результат нахождения всех разложений натурального числа N согласно проблеме Гольдбаха – Эйлера является подмножеством множества простых чисел от 1 до N , одной из подзадач этого алгоритма станет реализация метода решета Эратосфена. После этого, если число N чётное, из полученного массива исключаются те числа, которые в сумме с любым из остальных элементов структуры не дают N . При нечётности N выполняется тот же самый алгоритм исключения лишних чисел, но для величины $(N - 3)$. В таком случае каждой паре найденных простых чисел будет соответствовать ещё одно простое число 3. Также стоит обратить внимание на то, что нахождение второго числа целесообразно реализовать с помощью бинарного поиска, поскольку заранее известно, что массив простых чисел отсортирован по возрастанию элементов.

Для начала приведём наиболее очевидный способ реализации основной процедуры поиска всех разложений числа N в соответствии с проблемой Гольдбаха – Эйлера, при котором простые числа и получаемый результат хранятся в одномерных целочисленных массивах.

```

type
  Tract = array[1..10000000] of
Integer;

<...>

procedure Goldbach(N: Integer; var A,
B: Tract; var num: Integer);
var
  i, cur, shift, opposite, it_num:
Integer;
begin
  if Odd(N) then {Если число N нечёт-
ное – поиск разложений для (N -
3)...}
    N := N - 3;
  Eratosthen(N, A, num); {Построение
решета Эратосфена для промежутка [1,
N] (запись результата в массив A).
Переменная num – количество найденных
простых чисел в этом промежутке.}
  shift := 1; {Инициализация перемен-
ной сдвига индекса в массиве A.}
  cur := 1; {Инициализация переменной
текущего индекса результирующего мас-
сива B.}
  if Odd(num) then {it_num – верхняя
граница цикла поиска пар соответству-
ющих чисел.}
    it_num := num shr 1 + 1
  else
    it_num := num shr 1;

```

```

for i := 1 to it_num do
begin
  opposite := N - A[cur shr 1 + 1 +
  shift];
  if not Exist(opposite, A, num)
then {Проверка наличия числа, которое
в сумме с уже простым A[cur shr 1 + 1
+ shift] даёт N, в массиве A (простое
ли оно).}
  Inc(shift) {Если число не най-
дено, пропускаем текущее (увеличиваем
сдвиг).}
  else
  begin
    V[cur] := A[cur shr 1 + 1 +
  shift]; {Если число найдено, записы-
ваем его пару и само это число в мас-
сив V.}
    V[cur + 1] := opposite;
    Inc(cur, 2);
  end;
end;
num := cur - 1;
end;

```

На рис. 1 приведён график зависимости времени, требуемого для разложения чётного (нечётного) числа на сумму двух (трёх) простых, от величины этого числа (в тестах использовался центральный процессор AMD Sempron 2500+ с тактовой частотой 1,75 ГГц; объём оперативной памяти – 1024 Мб). При этом подавляющую часть времени для каждого случая занимает алгоритм построения массива на основе метода решета Эратосфена (бинарный поиск второго числа пары и исключение лишних элементов из массива для любого числа из рассматриваемого промежутка

занимают не более 20 миллисекунд). Как видно из графика, эта зависимость нелинейная. Тогда как коэффициент пропорциональности между отношением 2000 к 1000 исходных величин и соответствующим временным отношением равен 1,76, такой же коэффициент для чисел 20000000 и 10000000 составляет 0,8.

Находя разложения всё больших чисел на сумму пар и троек простых, мы довольно быстро столкнёмся с ограничениями встроенных типов данных ($N < 2^{31}$). В случае же использования длинной арифметики возникнет проблема нехватки оперативной памяти для хранения массивов простых чисел. Так, при записи длинных чисел в массивы из 1000 элементов, оперативной памяти объёмом в 1024 мегабайта не будет хватать уже для попытки разложения числа $2^{27} + 1$ (без учёта использования файла подкачки операционной системы), поскольку в каждый момент времени выполнения основной процедуры нужно знать все простые числа, не превосходящие N . Следовательно, необходимы два массива из N длинных чисел. В один из них заносятся простые числа и отсеиваются ненужные, другой служит для записи результата. При этом каждое число в обоих массивах занимает 4000 байтов памяти.

Всё это наталкивает на мысль о необходимости использования временных файлов. Причём логичнее прибегнуть не к текстовым файлам, а к типизированным. Такие файлы содержат данные определённого типа (встроенного или пользовательского), а связанные с ними файловые переменные объявляются следующим образом: *var F: file of Integer* (переменная F соответствует файлу целочисленных величин). Главной особенностью этих файлов является возможность произвольного доступа к их компонентам с помощью

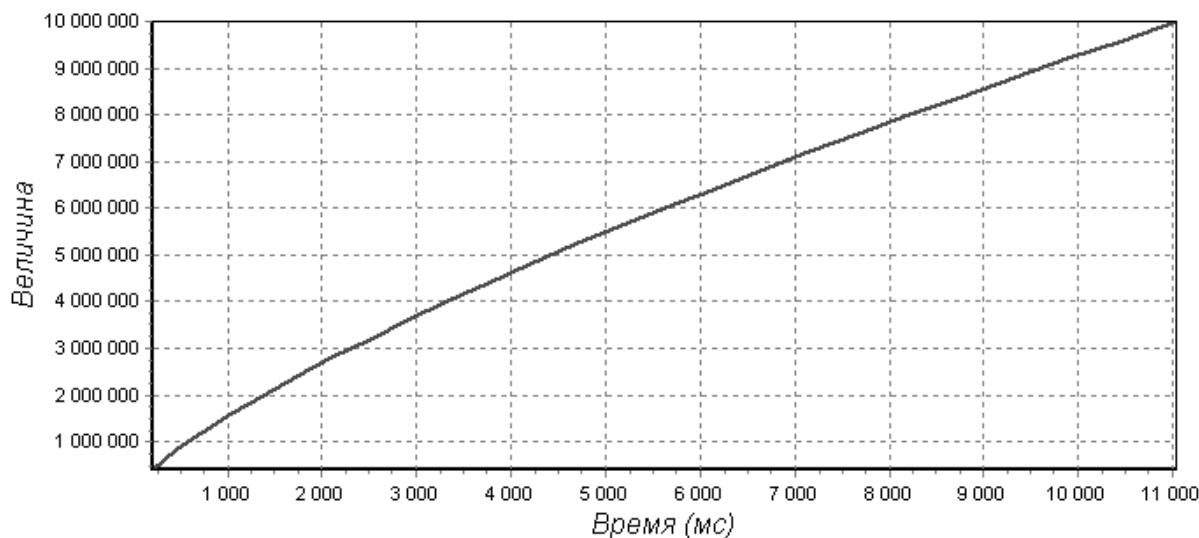


Рис. 1

процедуры Seek. При этом нумерация компонент начинается с нуля. В частности, чтобы обратиться к третьей компоненте типизированного файла F , необходимо выполнить оператор $Seek(F, 2)$, после чего можно осуществлять чтение из этой компоненты или запись в неё (в типизированных файлах, в отличие от текстовых, допускается чередование операций записи и чтения вне зависимости от оператора, используемого при открытии файла). Такие свойства типизированных файлов позволяют применять в них бинарный поиск.

Сначала попробуем воспользоваться файлами для 32-битовых переменных. В этом случае понадобится лишь два таких файла. Тогда подпрограмма поиска разложений будет выглядеть аналогично уже рассмотренной, только вместо действий над массивом будут выполняться операции чтения и записи в файл. Естественно, что в этом случае перед вызовом подпрограммы Goldbach файловые переменные должны быть связаны с внешними файлами и открыты процедурой Rewrite, причём лишь один раз.

Рис. 2 содержит графики, аналогичные первому. Сплошной линией показана временная зависимость при использовании файлов, пунктирной – при употреблении массива. Видно, что для записи и чтения из файла необходимо в десятки раз большее количество времени, чем для выполнения этих же операций над массивом. Это объясняется тем, что скорость передачи и время доступа у жёсткого диска гораздо более низкие, чем у оперативной памяти.

Для реализации этого алгоритма при помощи структуры типизированных файлов, содержащих одномерные массивы для хранения простых длинных чисел, необходимо создание нескольких по-

добных файлов. Их количество будет зависеть от величины числа N в соответствии с соотношением $(N - 1) \div MaxInt + 1$.

На рис. 3 пунктирная линия описывает время работы алгоритма, использующего типизированные файлы одномерных массивов, в которые записываются длинные числа, причём основание системы счисления в одном элементе структуры равно 1000000000, каждый массив состоит из 1000 элементов. График, обозначенный точками, отображает этот же алгоритм с той лишь разницей, что в нём используются массивы из 10 элементов. Сплошная линия соответствует алгоритму, реализованному через файлы целых чисел. Сокращение количества элементов в массивах для хранения длинных чисел, с одной стороны, ведёт к резкому уменьшению диапазона величин (с 9000 цифр в одном числе до 90), с другой – к росту быстродействия в несколько раз. Такая значительная разница в эффективности объясняется тем, что на инициализацию массивов с большим числом элементов требуется заметно большее количество времени.

Стоит отметить, что заносить длинные числа в типизированные файлы в виде линейных списков невозможно, поскольку такие файлы по определению содержат записи фиксированной длины и не могут указывать на внешние данные.

В XX в. был изобретён ещё один алгоритм нахождения простых чисел в определённом диапазоне. Он оптимизирует работу решета Эратосфена и в целом действует по сходному принципу. В честь своего создателя он получил название решета Аткина. Его суть заключается в первоначальном выборе произвольного натурального числа M и последующем поиске чисел n , удовлетворяющих двум условиям:

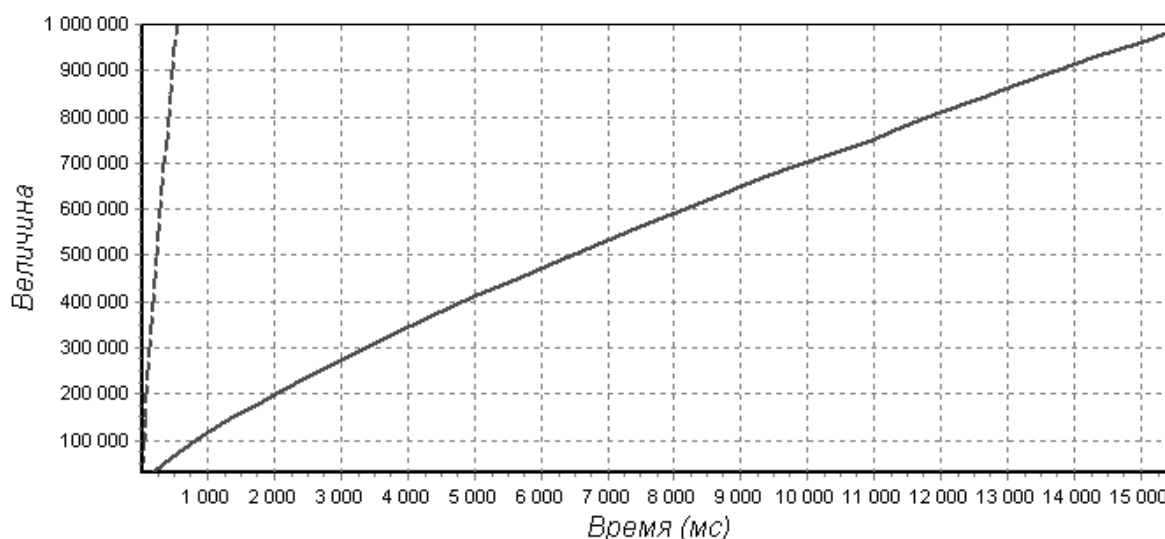


Рис. 2

• $n = a_i x^2 + b_j y^2$ для всех натуральных x, y из промежутка $[1, \sqrt{N}]$;

• $n \bmod M = r_j$, где r_j – определённые натуральные числа, зависящие от выбранного M .

При этом имеет место взаимно-однозначное соответствие между числами M, a_i, b_j, r_j .

Если количество найденных n нечётно, элемент результирующего логического массива с индексом n принимает значение *True* (изначально все элементы массива приравнены к *False*). Далее необходимо итеративно присвоить значение *False* элементам полученного массива, если их индексы кратны квадратам чисел, являющихся в свою очередь индексами элементов данного массива со значением *True*. В итоге ненулевые значения будут принимать только те элементы массива, индексы которых – простые числа.

Кроме того, что сам рассматриваемый алгоритм работает значительно быстрее, чем его предшественник (так как проход осуществляется не по самим числам, а по их квадратам), наличие логического результирующего массива освобождает от необходимости осуществления поиска второй пары простого числа. Можно просто обратиться с соответствующим индексом к построенному массиву. Тогда процедура Goldbach для реализации через целочисленный массив будет выглядеть следующим образом:

```
type
  Tract = array[1..10000000] of
    Integer;
var
  B: Tract;
  simple_nums: array[1..10000000] of
    Boolean; {Логический массив для хранения информации о простых числах.}
```

```
procedure Goldbach(N: Integer; var B:
  Tract; var num: Integer);
var
  i, cur, opposite, it_num: Integer;
begin
  if Odd(N) then
    N := N - 3;
    Atkin(N); {Построение решета Аткина
    (запись результата в логический
    массив simple_nums).}
    simple_nums[2] := True;
    simple_nums[3] := True;
    cur := 1;
    if Odd(num) then
      it_num := num shr 1 + 1
    else
      it_num := num shr 1;
    for i := 2 to it_num do
      if simple_nums[i] then
        begin
          opposite := N - i;
          if simple_nums[opposite]
          begin
            B[cur] := i;
            B[cur + 1] := opposite;
            Inc(cur, 2);
          end;
        end;
      num := cur - 1;
    end;
```

На рис. 4, 5 и 6 изображены временные зависимости алгоритмов поиска разложений через массив, типизированные файлы целых чисел и файлы массивов соответственно. Сплошная линия отображает график реализации при помощи решета Аткина, пунктирная – посредством решета Эратосфена.

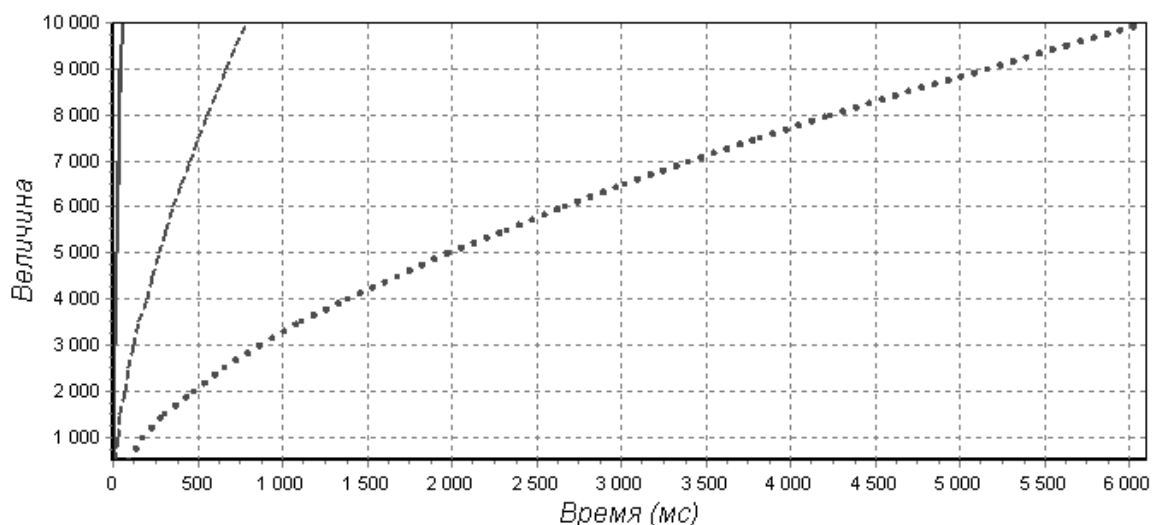


Рис. 3

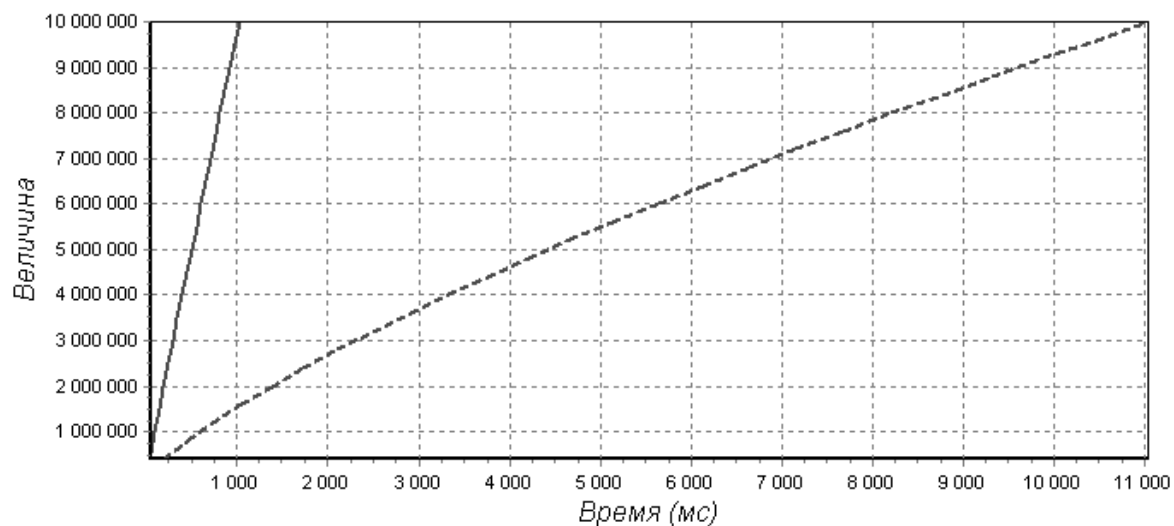


Рис. 4

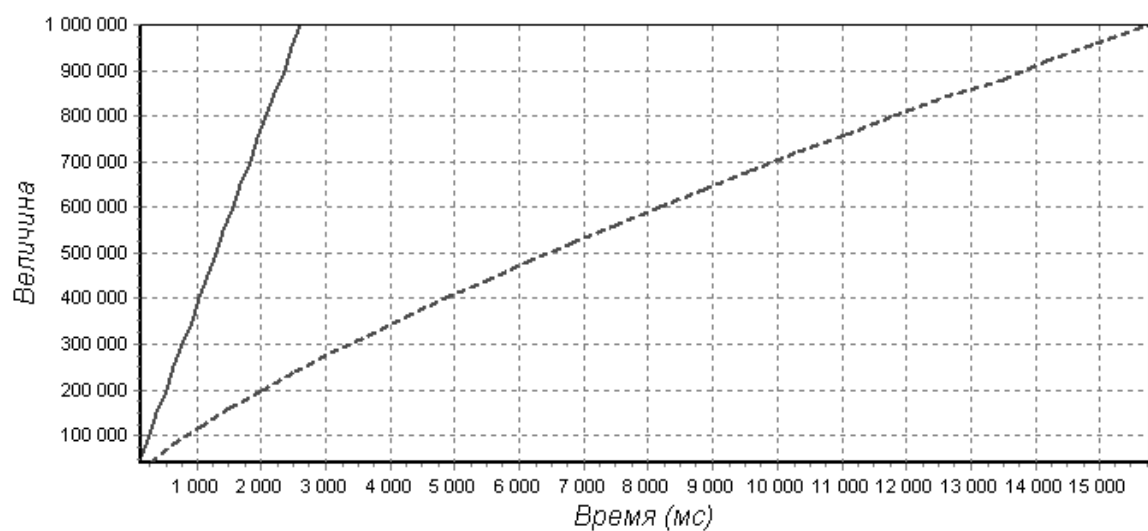


Рис. 5

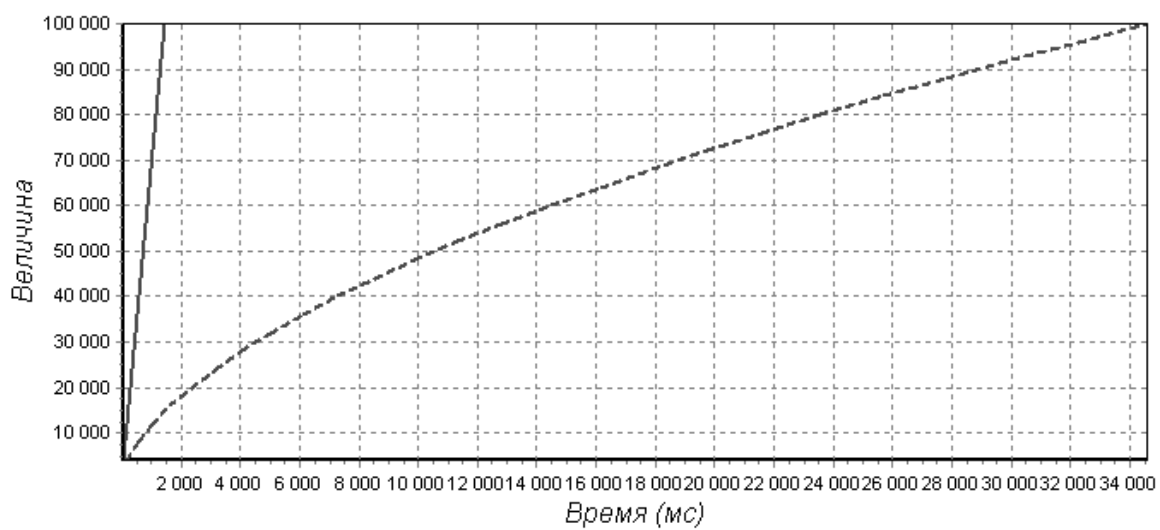


Рис. 6

Очевидно, что решето Аткина повышает эффективность алгоритма в несколько раз. Кроме того, для программы поиска чисел согласно проблеме Гольдбаха – Эйлера с использованием решета Аткина зависимость от времени для определённого количества начальных величин близка к линейной с коэффициентом 1, а сложность на

бесконечности стремится к $O\left(\frac{n}{\log(\log(n))}\right)$.

Ниже приведены результаты тестов для различных реализаций алгоритма поиска разложения случайно выбранного числа 516250 в соответствии с проблемой Гольдбаха–Эйлера.

В итоге все попытки улучшить исходный алгоритм приводят лишь к более или менее значительным результатам, но даже в самом оптимальном случае разложение уже 20-значного числа современный персональный компьютер будет находить в течение более чем 2000 лет.

	Массивы		Файлы целых чисел		Файлы массивов	
	Решето Эратосфена	Решето Аткина	Решето Эратосфена	Решето Аткина	Решето Эратосфена	Решето Аткина
Затраченное время (мс)	260	50	7140	1392	786231	7701
Выделенная оперативная память (б)	4665344	3596288	2592768	2592768	2596864	2596864
Суммарное место на диске для временных файлов (б)	—	—	199984	546914	2148300	853554

А. Е. Скобелев

РЕШЕТА ПРОСТЫХ ЧИСЕЛ

Рассматриваются некоторые задачи, связанные с простыми числами, и распространённые алгоритмы перечисления простых чисел на заданном интервале: решета Эратосфена, Аткина и Сундарама.

Простое число – число, не имеющее делителей, кроме единицы и самого себя. Основные связанные с ними задачи – проверка того, является ли данное число простым (тест простоты) и перечисление простых чисел до некоторого значения.

Проверка простоты числа

Самый простой способ проверки простоты данного числа – непосредственно по определению.

```
function IsPrime(x: integer):
boolean;
var
i: integer;
begin
Result:= true;
for i:= 2 to x - 1 do
Result:= Result and (x mod i <> 0);
end;
```

СКОБЕЛЕВ Артем Евгеньевич – студент II курса факультета информатики ВятГУ
© Скобелев А. Е., 2008

Происходит перебор всех делителей на отрезке $[2; a - 1]$.

Если число a составное, то оно имеет хотя бы один делитель в промежутке $[2; \lfloor \sqrt{a} \rfloor]$. Поэтому функцию можно оптимизировать:

```
function IsPrime(x: integer):
boolean;
var
i: integer;
r: integer;
begin
Result:= true;
r:= trunc(sqrt(x)) + 1;
for i:= 2 to r do
Result:= Result and (x mod i <> 0);
end;
```

Сложность такой функции $O(N)$ (первый вариант) или $O(N^{1/2})$ (второй вариант). Для больших значений N желателен более быстрый алгоритм. Такие алгоритмы для определения простоты чисел существуют, правда, многие из них являются вероятностными (алгоритм Миллера – Рабина, тест BPSW и т. п.).

Перечисление простых чисел

Следующая, более распространённая задача – перечисление простых чисел из данного интер-

вала (чаще всего от 1 до N). Очевидное решение – перебор:

```

procedure ListPrimes(const N: integer);
begin
  for i:= 1 to N do
    if IsPrime(i) then
      Write(i, ' ');
end;

```

Такой метод не требует больших объемов памяти, но потребляет много вычислительных ресурсов. В этом случае сложность алгоритма составляет $O(N^2)$ или $O(N^{3/2})$, что делает его неработоспособным при больших N. Более эффективны способы, основанные на использовании так называемых решет: решета Эратосфена, решета Аткина и решета Сундарама.

Решето Эратосфена

Решето Эратосфена – самый известный и самый простой способ нахождения списка простых чисел. Суть метода рассмотрена на примере.

Пример. Найти простые числа до 20.

N = 20.

Выписываются по порядку все числа от 2 до N.
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Вычеркиваются все числа, кратные 2.

2 3 5 7 9 11 13 15 17 19

Затем вычеркиваются числа, кратные 3.

2 3 5 7 11 13 17 19

Так делается до тех пор, пока мы не остановимся на последнем числе из списка. Необходимо вычеркнуть все числа $k * p$, где p – простое, а $k = 1, 2, \dots$ (в примере $p = 3$). После этого все составные числа из интервала будут вычеркнуты и останутся только простые числа. Для увеличения скорости может применяться разложение чисел на множители.

Алгоритм будет эффективнее, если просматривать числа не все числа p на интервале $[2; N]$, а только те числа, которые удовлетворяют условию $p^2 \leq n$.

Примечание 1. Во всех следующих программных блоках предполагается, что число N определено заранее.

Примечание 2. Поскольку в следующих алгоритмах рассматриваются натуральные числа (в рамках типа integer), программные блоки можно оптимизировать, заменив умножение на степени 2 операциями битового сдвига. Также можно более эффективно работать со структурой хранения данных (использовать не тип boolean, а совокупность битовых флагов), что потребует минимального изменения кода.

Вариант программной реализации решета Эратосфена:

```

var
  i, j: integer;
  prime: array of boolean;
  sqrN: integer;
begin
  sqrN:= Trunc(Sqrt(N));
  SetLength(prime, N + 2);
  for i:= 1 to N do
    prime[i]:= false;
  i:= 2;
  repeat
    while (i <= sqrN) and not
      (prime[i]) do
        inc(i);
    if (i <= sqrN) then
      begin
        j:= 2 * i;
        while (j <= n) do
          begin
            prime[j]:= false;
            j:= j + i;
          end;
        end;
        inc(i);
      until (i > sqrN);
    end;
  for i:= 2 to N do
    Write(i, " ");

```

Временная сложность – $O(N \log N)$.

Оценка времени для разных значений N (P-III, 850 МГц):

N	10^3	10^4	10^5	10^6	10^7
Время, с	0,0	0,0	0,0	0,15	1,782

Здесь и далее оценка времени производится с помощью демонстрационной программы PrimeSieves (<http://fmsoftware.narod.ru/Work/Sieves/PrimeSieves.zip>).

Решето Аткина

Решето Аткина – быстрый современный алгоритм для нахождения простых чисел до определенного значения. Создан А. О. Л. Аткином и Д. Бернштейном. Это оптимизированная версия решета Эратосфена. Алгоритм проделывает некоторую предварительную работу, а затем вычеркивает числа, кратные квадратам простых чисел.

Алгоритм вычисляет простые числа до N, используя $O(N/\log \log N)$ операций и $O(N^{1/2+o(1)})$ ячеек памяти. Для сравнения, решето Эратосфена использует $O(N)$ операций и $O(N^{1/2}(\log \log N)/\log N)$ ячеек памяти.

Алгоритм:

- Под остатками подразумеваются остатки – остатки по модулю 60 (от деления на 60).

- Все числа, включая x и y , натуральные.
- Под перебрасыванием значения понимается его смена на противоположное.

1. Создать список результатов, заполненный 2, 3 и 5.

2. Создать массив решета с ячейками для каждого натурального числа (до N). Все ячейки должны быть помечены как составные.

3. Для каждого элемента массива:

- если элемент – число с остатком 1, 13, 17, 29, 37, 41, 49 или 53, перебросить значение для каждого возможного решения $4x^2 + y^2 = \langle \text{индекс элемента} \rangle$;

- если элемент – число с остатком 7, 19, 31 или 43, перебросить значение для каждого возможного решения $3x^2 + y^2 = \langle \text{индекс элемента} \rangle$;

- если элемент – число с остатком 11, 23, 47 или 59, перебросить значение для каждого возможного решения $3x^2 - y^2 = \langle \text{индекс элемента} \rangle$.

4. Начать с наименьшего числа в решете.

5. Найти следующее за ним число, помеченное простым.

6. Добавить это число в список результатов.

7. Возвести число в квадрат и пометить все кратные ему как составные.

8. Повторять шаги 5–8 до достижения квадратного корня из N .

Вариант программной реализации:

```
var
  sqrN: integer;
  prime: array of boolean;
  x2, y2: integer;
  i, j, k: integer;

begin
  sqrN:= Trunc(Sqrt(N));
  SetLength(prime, N + 2);
  for i:= 1 to N do
    prime[i]:= false;

  x2:= 0;
  for i:= 1 to sqrN do
    begin
      x2:= x2 + 2 * i - 1;
      y2:= 0;
      for j:= 1 to sqrN do
        begin
          y2:= y2 + 2 * j - 1;

          k:= 4 * x2 + y2;
          if ((k <= N) and ((k mod 12 =
1) or (k mod 12 = 5))) then
            prime[k]:= not prime[k];

          k:= k - x2;
          if ((k <= N) and (k mod 12 =
7)) then
```

```
            prime[k]:= not prime[k];

          k:= k - 2 * y2;
          if ((i > j) and (k <= N) and
(n mod 12 = 11)) then
            prime[k]:= not prime[k];
          end;
        end;
      end;
    end;
  for i:= 5 to sqrN do
    begin
      if prime[i] then
        begin
          k:= i * i;
          for j:= 1 to N div k do
            prime[j * k]:= false;
          end;
        end;
      end;
    end;
  Write("2 3 ");
  for i:= 5 to n do
    if prime[i] then
      Write(i, " ");
  end;
```

Оценка времени для разных значений N :

N	10^3	10^4	10^5	10^6	10^7
Время, с	0,0	0,0	0,01	0,125	1,366

Решето Сундары

Решето Сундары – простой детерминированный алгоритм для нахождения простых чисел до некоторого целого числа. Разработан индийским математиком С. П. Сундары в 1934 г.

Из ряда натуральных чисел исключаются члены вида $z = i + j + 2ij$, где $i = 1, 2, 3, \dots, N$, $j = 1, 2, \dots, i$. Оставшиеся числа умножаются на 2, и к результату прибавляется 1. Полученная последовательность представляет собой ряд простых чисел p :

```
4 7 10 13 16 19 ...
7 12 17 22 27 32 ...
10 17 24 31 38 45 ...
13 22 31 40 49 58 ...
16 27 38 49 60 71 ...
```

Эта таблица состоит из бесконечных арифметических прогрессий, причем каждый член первой прогрессии начинает новую прогрессию. Разности прогрессий являются нечетные числа, начиная с 3. Если некоторое число p находится в этой таблице, то $2 * n + 1$ – составное число, иначе $2 * n + 1$ – простое. Заменяя n в формуле натуральными числами, которых нет в таблице, можно получить все простые числа, большие 2.

Временная сложность алгоритма – $O(N \log N)$, что хуже, чем у решета Эратосфена. Высокая

сложность отчасти компенсируется тем, что с помощью массива длины N это решето находит простые числа на интервале до $2 * N + 1$, хотя эта особенность может требовать дополнительных условных конструкций для записи в массив.

Отличие алгоритма – его компактность на любом языке программирования. Вариант программной реализации:

```
var
prime: array of boolean;
i, j: integer;
idx: integer;

begin
SetLength(prime, N + 2);
for i:= 1 to N do
  prime[i]:= true;

for i:= 1 to n do
  for j:= 1 to i do
    begin
      idx:= i + j + 2 * i * j;
      if (idx <= n) then
        prime[idx]:= false;
    end;

for i:= 1 to n do
  if prime[i] then
    Write(2 * i + 1, " ");
end;
```

Оценка времени для разных значений:

N	10^3	10^4	10^5
Время, с	0,04	0,455	42,552

Среднее время работы алгоритмов, с:

N	10^3	10^4	10^5	10^6	10^7
Решето Эратосфена	0,00	0,00	0,01	0,15	1,822
Решето Аткина	0,00	0,01	0,01	0,128	1,422
Решето Сундарамы	0,05	0,482	44,564	–	–
Перебор делителей	0,03	0,018	0,35	9,534	276,227

Примечание. Решето Сундарамы при $N = 10^6$ требует порядка 1 ч 20 мин.

Е. В. Соболева

К ВОПРОСУ О РАЗРАБОТКЕ ТРЕНИНГА ПЕДАГОГИЧЕСКОГО ОБЩЕНИЯ ДЛЯ УЧИТЕЛЕЙ ИНФОРМАТИКИ

В данной статье на примере конкретного занятия представлен фрагмент тренинга для учителей информатики, ориентированного на оказание помощи в разрешении различных проблем педагогического общения, возникающих на компьютерном уроке.

Проблеме общения в психологии и педагогике посвящены многочисленные работы (А. В. Петровский, Н. В. Кузьмина, В. А. Кан-Калик, С. В. Кондратьева, А. А. Леонтьев, Я. Л. Коломинский и др.). В них определена структура и особенности педагогической коммуникации, выделены её основные образующие компоненты, сделана попытка изучить их взаимосвязь с содержательными и методическими аспектами как педагогического процесса, так и деятельности самого педагога.

Также была выделена категория педагогического общения, которая является стержнем педагогической деятельности. Однако если содержательные и методические аспекты педагогической деятельности в настоящее время в значительной мере исследованы и исследуются, то «психологический срез обучения и воспитания, и деятельности педагога, и проблемы общения не получили в должной мере ни теоретического, ни прикладного осмысления» [1].

Исследование психологических феноменов педагогического процесса в целом и проблем педагогического общения в частности представляется на сегодняшний день весьма актуальным.

В научно-методической литературе существует несколько подходов к определению понятия и сущности педагогического общения. Отметим те из них, которые оказали значительное влияние на проводимую автором работу.

Так, например, в трудах одного из популярных отечественных психологов В. А. Кан-Калика, довольно чётко прослеживается идея о том, что «профессионально-педагогическое общение – это система приёмов и методов, обеспечивающих реализацию целей и задач педагогической деятельности и организующих, направляющих социально-психологическое взаимодействие педагога и воспитуемых; содержанием этого взаимодействия является обмен информацией, творческое и интеллектуальное развитие, межличностное познание, организация и регуляция взаимоотношений с помощью различных коммуникатив-

СОБОЛЕВА Елена Витальевна – ассистент кафедры информатики и методики обучения информатике ВятГГУ © Соболева Е. В., 2008

ных средств в целях оказания воспитательного воздействия» [2].

Данное определение подчёркивает не только системный характер педагогического общения, но и взаимосвязь его дидактических, воспитательных и развивающих функций. И это необходимо учитывать при подготовке молодых специалистов.

При осмыслении общения в системе педагогического творчества учитывались положения, разработанные Б. Г. Ананьевым [3], А. А. Бодалевым [4], Н. В. Мясищевым [5], их учениками и сотрудниками: общение занимает важное и самостоятельное место среди важнейших видов человеческой деятельности, выступает как фактор формирования личности, многофункционально, предлагает осуществление важнейшей внутренней стороны человеческого взаимодействия – познания участниками общения друг друга, социально-перцептивные процессы играют важную роль в осуществлении качественных преобразований личности.

Идею о взаимосвязи функций педагогического общения поддерживает и развивает ещё один классик отечественной психологии, А. А. Леонтьев. Он отмечает, что «оптимальное педагогическое общение – такое общение учителя со школьниками в процессе обучения, которое

1) создаёт наилучшие условия для развития мотивации учащихся и творческого характера учебной деятельности, для интеллектуального развития личности;

2) обеспечивает благоприятный эмоциональный климат обучению (препятствует возникновению психологического барьера);

3) позволяет максимально использовать в учебном процессе личностные особенности учителя» [6].

Таким образом, и Кан-Калик, и Леонтьев, говоря об общении, организуемом учителем на уроке, выделяют его в самоценный вид деятельности, который можно рассматривать как метод интеллектуального развития.

Использование компьютера, как это показано в работах [7], позволяет не только усилить традиционные информационные потоки (учитель – ученик, ученик – ученик), возникающие при общении участников учебного процесса, тем самым активизируя информационный обмен, но и добавить новые (между учеником и компьютером, между учителем и учеником посредством компьютера). Кроме того, компьютерный урок информатики обладает рядом особенностей, которые необходимо учитывать при организации педагогического общения, т. е. при выборе указанной системы приёмов и методов взаимодействия.

Анализ деятельности учителей информатики, особенно начинающих, позволяет сделать вывод,

что организация процесса общения как вида деятельности, в ходе которого происходит взаимодействие между учеником, учителем и компьютером, вызывает определённые трудности. Это связано прежде всего с тем, что учителя информатики не всегда знают структуру и правила педагогического общения, в большинстве случаев не используют проблемы педагогического общения в качестве инструмента для достижения образовательных и воспитательных целей, у них плохо развиты коммуникативные способности и коммуникативная культура в целом.

Один из возможных путей разрешения подобных проблем – это проведение с учителями (будущими и работающими) ролевых игр и тренингов по общению.

Несмотря на многочисленность подобных тренингов в психолого-педагогической практике (А. С. Прутченков, Н. А. Ольшевская, Н. А. Богомолова, Л. А. Петровская и др.), они теряют свою эффективность на уроке информатики, так как не учитывают возможности компьютера.

Для плодотворного теоретического, экспериментального и прикладного исследования педагогического общения необходима разработка такого тренинга, в структуре которого можно было бы объединить рассмотрение всех функций общения с учётом особенностей компьютерного урока информатики. Поэтому был поставлен вопрос о разработке специального тренинга по общению для учителей информатики [8], который может стать (при эмпирическом подтверждении) хорошим практическим подспорьем для проведения психодиагностической и воспитательной работы, для обеспечения условий, способствующих интеллектуальному развитию личности школьника.

В рамках этого тренинга планируется проведение работы, направленной на формирование у учителей информатики «психологических качеств диалогического общения, т. е. качеств, обеспечивающих оптимальный уровень взаимодействия с другими людьми (в первую очередь со своими учениками)» [9].

Напомним, что предлагаемый тренинг рассчитан на шесть занятий.

Первое занятие:

1) проведение теста «Психологический портрет учителя информатики», в ходе которого намечается круг проблемных вопросов, связанных с педагогическим общением на уроке информатики, и определяются цели дальнейшей работы;

2) деловая игра «Оценка педагогических программных средств по информатике», при которой учителя анализируют деятельность других учителей, связанную с применением компьютера на уроке.

Второе занятие полностью посвящено анализу реальных практических конфликтных ситуаций при выстраивании и организации педагогического общения (в которых компьютер используется как арбитр или помощник) и моделированию возможных решений. Значение этого занятия в том, что предложенные упражнения дают возможность участнику побыть и в роли учителя, и в роли ученика, и в роли стороннего наблюдателя, увидеть, как другие применяют компьютер в разных ситуациях и как это влияет на организацию взаимодействия.

Третье занятие – конференция, посвящённая вопросам и проблемам организации и проведения лабораторной работы по решению задач за компьютером.

Четвёртое занятие посвящено обсуждению ситуаций общения на уроке информатики и возможностей применения компьютера в качестве активного участника общения.

На **пятом занятии** планируется организовать работу, направленную на формирование положительного отношения к привлечению игрового элемента при организации работы с компьютером, оценки его влияния на организацию учебного процесса и общения.

На **шестом занятии** предлагается проведение упражнения, в ходе которого участники обмениваются мнениями о методах и приёмах, используемых при организации процесса общения на уроке информатики и при использовании компьютера для взаимодействия в различных ситуациях.

Завершаться тренинг будет тем же тестом, с которого всё начиналось.

Более подробное описание структуры тренинга, предлагаемых средств и форм проведения представлено в предыдущей работе [10].

Рассмотрим фрагмент данного тренинга на примере первого занятия.

На нём предлагается провести тест «Психологический портрет учителя информатики», в котором присутствуют вопросы по следующим направлениям:

- 1) приоритетные ценности учителя;
- 2) психоэмоциональное состояние;
- 3) самооценка;
- 4) стиль преподавания.

В каждом направлении основная часть вопросов относится к организации общения на уроках информатики и ориентирована на выявление конкретных проблем в общении для конкретного учителя.

Кроме того, тест содержит блок вопросов, направленных на определение степени достоверности работы участника для оценки его открытости и готовности к взаимодействию с группой. Эти вопросы могут использоваться и на

стадии рефлексии, но при проведении остальных занятий тренинга их результаты учитываться не будут.

После того как участники ответят на вопросы, им, вопреки законам тестирования, интерпретация теста не выдаётся. Но лист с ответами сохраняется до конца тренинга. Дело в том, что первое занятие направлено на определение круга проблемных вопросов, на ориентацию к работе, а не на мгновенную «раздачу слонов». Участник, отвечая на вопросы, задумывается над проблемами, которые в них охватываются, и на следующих занятиях при разрешении разных ситуаций будет поневоле их вспоминать, корректировать свою точку зрения или, наоборот, убеждаться в собственной правоте. Таким образом, учитель сам определяет, что для него важно.

Как отмечает Л. А. Петровская [11], такая активная позиция в виде самодиагностики является важной предпосылкой эффективности последующих усилий, направленных на коррекцию тех аспектов, которые были выявлены в качестве вероятных источников затруднений в общении.

I. Тест «Психологический портрет учителя информатики»

1. На уроках информатики основная цель:
 - a) развивать логическое мышление;
 - b) сформировать основные навыки и умения работы с компьютером;
 - c) поддерживать благоприятную «рабочую» атмосферу.
2. Если в ходе урока ученики обмениваются мнениями, разбирают решения друг друга, то я испытываю:
 - a) интерес;
 - b) каждый должен решать сам, не нарушая при этом дисциплины;
 - c) мне это безразлично, был бы результат.
3. Когда ученик при решении задачи использует не изученные ранее конструкции или методы, то я выражаю:
 - a) восхищение и одобрение;
 - b) одобрение, но пожелание решить ещё и другим, уже разобранным способом;
 - c) недовольство.
4. Когда я допускаю на уроке ошибку:
 - a) мне трудно признать это;
 - b) я могу признать её и похвалить учеников, обнаруживших её;
 - c) таких случаев не было.
5. При организации работы на уроке по решению задач с помощью компьютера я много внимания уделяю:
 - a) общению с учениками и консультированию;
 - b) самостоятельности учеников;

- с) дисциплине.
6. Если ученик не может быстро понять идею решения, то это вызывает у меня раздражение и недовольство, которое бывает трудно скрыть:
- никогда;
 - иногда;
 - часто.
7. Я считаю, что ученикам в кабинете нельзя пользоваться компьютерными игрушками даже на переменах:
- напротив, компьютерные игры способствуют снятию психоэмоционального напряжения учащегося;
 - верно;
 - зависит от ситуации и компьютерной игры.
8. Я считаю, что как учитель по информатике во многом превосхожу других знакомых учителей по этому предмету:
- нет;
 - не уверен;
 - да.
9. На перемене я предпочитаю общаться:
- с учениками, у которых возникли на занятии проблемы;
 - с учениками, которые проявляют дополнительный интерес к предмету;
 - лучше побыть одному.
10. При организации работы по решению задач с помощью компьютера я предпочитаю:
- давать алгоритмы и чёткие инструкции;
 - не вмешиваться в работу, для меня важен результат;
 - предлагать простор для творчества, но оказывать необходимые консультации.
11. На уроках мне случалось рассказывать материал, в котором я плохо разбираюсь:
- да;
 - не помню;
 - нет.
12. После нахождения с учениками удачного решения, ошибки в программе я чувствую себя счастливым:
- часто;
 - редко;
 - никогда не испытывал подобных эмоций в данной ситуации. Я считаю, что ученик должен самостоятельно работать над задачей. У него уже есть помощник – компьютер.
13. Замечания других учителей информатики по форме и организации работы и общения:
- являются стимулом для изменения стиля работы;
 - задевают меня;
 - меня мало волнуют.
14. Во время урока я чётко придерживаюсь конспекта и подобранной системы задач:
- всегда;
 - предпочитаю импровизацию и спонтанность;
 - зависит от ситуации.
15. Когда я вижу, что ученик при решении задачи игнорирует мой совет:
- плачу ему той же монетой;
 - пытаюсь разобраться в ситуации;
 - не придаю этому значения.
16. Я стараюсь следить за развитием науки и по каждой теме подбираю новый интересный материал:
- всегда;
 - обычно;
 - иногда.
17. Самым главным для учителя является:
- самореализация и успешность учеников;
 - одобрение коллег;
 - чувство собственной необходимости.
18. Атмосфера на уроке зависит от моего душевного и физического состояния:
- иногда;
 - в большинстве случаев;
 - не зависит, учитель всегда должен быть «на коне».
19. Если я обращаюсь к коллеге с просьбой, то мне всегда помогут:
- уверенности нет;
 - затрудняюсь ответить;
 - конечно.
20. Если ученик предлагает решение, которое ошибочно:
- в таких случаях внимание всех остальных учеников также надо обратить на допущенную ошибку (и того, кто её сделал) – чтобы они не повторяли её в своих решениях;
 - говорю ученику, что он не прав, а ошибку пусть ищет и исправляет сам;
 - объясняю ученику его ошибку, используя компьютер в качестве арбитра.
21. По-моему, на уроке информатике важнее всего:
- возможность работать творчески;
 - отсутствие конфликтов;
 - трудовая дисциплина.
22. Выскажите свое отношение к высказыванию «Я ничему не могу научить этого ученика, потому что у него нет способностей к выполнению самостоятельной работы при решении задач»:
- что-то в этом есть;
 - согласен полностью;
 - не согласен.
23. В своём профессионализме:
- есть сомнения;
 - не приходилось задумываться;
 - у меня нет сомнений.
24. Я почти всегда иду на урок в приподнятом настроении:
- да;
 - не всегда;

- с) нет.
25. Я считаю, что учитель на уроке не должен переходить на выяснение отношений, лучше всего переносить конфликт «в плоскость решения задач»:
- с некоторыми учениками это не получается;
 - никогда не пользовался таким приёмом;
 - да, это помогает.
26. Общение с учениками дается мне ценой большого напряжения:
- редко;
 - иногда;
 - всегда.
27. На конференциях и методических советах я делюсь опытом, выступаю по актуальным проблемам:
- часто;
 - иногда;
 - мне там чаще всего не интересно.
28. Я считаю, что совместная работа ученика и учителя (особенно в действиях «против» компьютера) способствует установлению атмосферы доброжелательности и пониманию общности целей:
- бесспорно;
 - возможно, но этим не стоит злоупотреблять;
 - это напрасная потеря времени.
29. Если учащийся не прав:
- надо сразу показать ученику на ошибку;
 - пусть разбирается сам;
 - в любой работе ученика надо сначала похвалить, а потом объяснить в чём он не прав. И желательно не показывать ошибку сразу, а постараться, чтобы он сам нашёл её, используя для этого компьютер и справочные материалы.
30. Мои ученики относятся ко мне с симпатией:
- не знаю;
 - не все;
 - да.
31. Мысли о проблемах во взаимоотношениях с учениками мешают мне:
- часто;
 - редко;
 - никогда не мешают.
32. Непредвиденные ситуации на уроках:
- только мешают учебному процессу;
 - лучше игнорировать;
 - можно эффективно использовать.
33. Я нахожу в себе достаточно сил, чтобы справиться с трудностями в общении и взаимодействии:
- редко;
 - обычно;
 - всегда.
34. На уроке я испытываю потребность в общении с учениками:
- да;
 - если мысли учеников интересные, почему бы не пообщаться;
 - нет.

35. Мне всегда удаётся сделать так, чтобы ученик считал идею решения своей собственной, даже если это не совсем так:
- редко;
 - обычно;
 - всегда.
36. Я считаю, что мои коммуникативные умения находятся на достаточно высоком уровне:
- не задумывался над этим вопросом;
 - нет;
 - да.
37. Правильный выход из конфликтной ситуации я нахожу:
- не всегда;
 - не задумываюсь о правильности принятых мной решений;
 - всегда.
38. Я опаздываю на работу:
- регулярно;
 - изредка;
 - никогда не опаздываю.
39. Я откладываю на завтра то, что необходимо сделать сегодня:
- часто;
 - иногда;
 - никогда.
40. Ответы на некоторые вопросы давались мне с трудом:
- скорее даже на все;
 - да, приходилось делать нелёгкий выбор;
 - нет.

Участники отвечают на вопросы теста, а затем заполняют предложенную ниже таблицу. Однако (повторимся) интерпретацию теста методист (руководитель тренинга) не даёт, а листы с результатами забирает до последнего занятия.

Приоритетные ценности учителя	№	1	5	9	16	17	21	27	34
	a								
	b								
Психоэмоциональное состояние	№	2	3	6	12	18	24	26	28
	a								
	b								
Самооценка	№	4	8	11	15	19	23	30	31
	a								
	b								
Стиль преподавания	№	7	10	14	20	22	25	29	32
	a								
	b								
Достоверность ответов	№	13	33	35	36	37	38	39	40
	a								
	b								
	c								

Интерпретация теста

1. Приоритетные ценности учителя информатики.

1.1. Если в ваших ответах преобладает первый вариант, то, несмотря на хорошие с большинством учеников отношения, Вы не боитесь задавать провокационные вопросы и умеете использовать возникающие конфликтные ситуации в учебных целях.

Вы понимаете значение программирования для развития интеллекта и соответственно организуете свою работу, однако не считаете свою точку зрения единственно верной.

Вы предпочитаете взаимодействие по стилю «ученик и учитель – компьютер» или «ученик – ученик».

1.2. Если в ваших ответах преобладает второй вариант, то Вы не жалуете времени для общения с учениками и коллегами, иногда даже в ущерб себе. На уроках активно используете компьютер в качестве «участника диалога», значение информатики видите в формировании базовых ЗУН по работе с компьютером.

При организации взаимодействия предпочитаете пользоваться связью «учитель через компьютер – ученик». Однако излишне много внимания уделяете поддержанию дисциплины, стараетесь избегать конфликтных ситуаций, что мешает творческому подходу к решению задач. Стабильный эмоциональный фон не всегда залог успешной работы.

1.3. Вам в тягость общение с учениками и коллегами, мало интересуют достижения науки и успехи учеников, а больше беспокоит сохранение авторитета. На уроке для Вас главное – порядок, поэтому взаимодействие с учениками и учеников между собой стараетесь свести к минимуму.

Вы предпочитаете воздействие (*а не взаимодействие!*) по традиционной схеме – «учитель – ученик». И возможности компьютера как участника диалога используете только на стадии проверки результата, как способ продемонстрировать свои знания.

2. Психоэмоциональное состояние.

2.1. Если в Ваших ответах преобладает первый вариант, то, очевидно, что общение с коллегами и учениками доставляет Вам радость. Вы не теряете самообладание в экстремальных ситуациях, активно используете компьютер в качестве участника диалога, для разрешения конфликтов.

Ваше благополучное психоэмоциональное состояние определяет эффективность вашей работы.

2.2. Если в Ваших ответах преобладает второй вариант, то Ваше настроение зависит от внешних факторов. На Ваше эмоциональное благополучие нередко оказывает влияние психоэмоциональное состояние в коллективе, поэтому Вы

стараетесь избегать длительного диалога с учеником. Для Вас легче составить подробную инструкцию, перевести закрепление нового материала в «плоскость самостоятельного решения задач».

2.3. Если в Ваших ответах преобладает третий вариант, то очевидно, что поддерживать общение с коллегами и учениками Вам трудно. И это связано не с тем, что Вы способны остро воспринимать состояние другого человека, сопереживать ему, помогать. Дело в том, что Вы не знаете, как организовать диалог с учениками и коллегами, а может быть, даже не задумываетесь, что у Вас есть такие коммуникативные проблемы. Но помните, что живое общение учителя с учеником не заменит даже самая умная машина.

3. Самооценка.

3.1. Если в Ваших ответах преобладает первый вариант, то Вам не всегда легко вступать в общение с другими. Вам свойственно принижать значение своей личности. Это мешает действовать спонтанно, уверенно, создавать благоприятную творческую атмосферу на уроке. Вы часто думаете о том, как бы не допустить ошибку или не сразу найти её у ученика в программе. Поэтому и общение у Вас на уроке часто сводится к контрольно-оценочной схеме, вы считаете, что любая помощь ученику принесёт больше вреда, чем пользы, так как развивает мышление только самостоятельное решение задачи.

3.2. Если в Ваших ответах преобладает второй вариант, то Вам свойственно адекватное самовосприятие. Вы отличаетесь творческим не догматическим мышлением, способны создавать на уроке атмосферу живого общения, выступая с учеником в единой связке против компьютера, что даёт Вам и учащимся возможность ощущать себя настоящими союзниками и даже друзьями. Благодаря этому становится возможным эмоциональное развитие школьников, а их представления о себе приобретают реалистический характер.

3.3. Если в Ваших ответах преобладает третий вариант, то Вам свойственно излишне позитивное самовосприятие. Вы редко прислушиваетесь к мнению учеников и других учителей, считая свой стиль преподавания единственно верным. Вы редко «скатываетесь на выяснение отношений», но не потому, что используете компьютер для подтверждения своей правоты, а по причине ощущения собственной непогрешимости. Деятельность учеников по решению задач направляется жёстко, но при этом мало внимание уделяете творчеству – Вам важен результат. Вы не признаёте ограниченность Ваших способностей и, как следствие, ограниченность возможностей и способностей учеников.

4. Стиль преподавания.

4.1. Если в Ваших ответах преобладает первый вариант, то Вам близка позиция миротворца. Как правило, Вы стараетесь избегать душевного напряжения, сопутствующего конфликтам. Вы редко испытываете чувство вины и не взваливаете на себя чужую работу. Организацию и контроль деятельности учащихся осуществляете без системы, проявляя нерешительность и колебания. Выбранный Вами либеральный стиль не способствует ни интеллектуальному развитию ученика, ни продуктивности в работе.

4.2. Если в Ваших ответах преобладает второй вариант, то, возможно, в учительском коллективе Вы имеете репутацию «сильного педагога». Ученик для Вас – объект воздействия, а не равноправный партнёр. Естественно, учащиеся это чувствуют, и в результате возникают барьеры взаимонепонимания. В случаях совершения учеником ошибок при решении задач Вы не спешите приходить на помощь и не пытаетесь прикладывать усилия для превращения ошибки в инструмент обучения. Возможности компьютера для установления взаимоотношений субъект-субъектного типа используются минимально.

Никакого совместного поиска решения задач, никакой общей работы над исправлением ошибки. В большинстве случаев Вы просто даёте задание и предоставляете инструкцию. Но нет ни обучения работе с инструкцией, ни возможности увидеть ход мысли ученика.

Наличие авторитарных тенденций в Вашем стиле преподавания не способствует взаимопониманию и творческой атмосфере урока.

4.3. Если в Ваших ответах преобладает третий вариант, то для Вас ученик – равноправный партнёр, вы привлекаете учащихся к совместному поиску решения, прислушиваетесь к их мнению, поощряете самостоятельность в выборе метода и алгоритма решения. При организации взаимодействия с учеником стараетесь учитывать не только успеваемость, уровень обученности, но и психологические качества личности.

5. Достоверность ответов.

Преобладание 1–2 вариантов говорит о высокой степени Вашей искренности и способности объективно оценивать себя.

Преобладание же ответов третьего варианта свидетельствует о том, что Вы ещё не чувствуете себя свободным, не готовы к открытому диалогу (иными словами, «не лезьте своими грубыми руками в мою нежную душу»).

Как уже было сказано, участники не получают результаты тестирования на этом же заня-

тии, а завершаться тренинг будет тем же тестом, с которого всё начиналось. На последнем занятии учителя не только получают его интерпретацию, но и листы с результатами первого тестирования. Таким образом, учителю предоставляется возможность самому сравнить полученные результаты до и после тренинга и сделать соответствующие выводы, которые он может обсуждать или не обсуждать с другими участниками.

Педагогическое общение на уроке информатики – это достаточно большая тема, требующая дальнейшего исследования, в связи с чем планируется не только проработать остальные занятия тренинга в соответствии с указанными целями [12], но и выполнить формальное описание схемы непосредственного взаимодействия учителя и ученика на уроке информатики, на примере которой показать применение представленных в тренинге правил и приёмов общения.

Примечания

1. Кан-Калик, В. А. Педагогическое общение как предмет теоретического и прикладного исследования [Текст] / В. А. Кан-Калик, Г. А. Ковалёв // Вопросы психологии. 1985. № 4. С. 9–16.

2. Там же.

3. Ананьев, Б. Г. Избранные психологические труды [Текст] : в 2 т. / Б. Г. Ананьев. М.: Педагогика, 1980. Т. 1. 232 с.; Ананьев, Б. Г. Избранные психологические труды [Текст] : в 2 т. / Б. Г. Ананьев. М.: Педагогика, 1980. Т. 2. 288 с.

4. Бодалёв, А. А. Психология общения. Избранные психологические труды [Текст] / А. А. Бодалёв. М.: Изд-во Моск. психол.-соц. ин-та; Воронеж: Изд-во НПО «МОДЭК» 2002. 320 с.

5. Мясичев, В. Н. Психология отношений [Текст] / В. Н. Мясичев. Воронеж: НПО «МОДЕК», 1995. 340 с.

6. Леонтьев, А. А. Педагогическое общение [Текст] / А. А. Леонтьев; под ред. М. К. Кабардова. 2-е изд. М.; Нальчик: Изд-во «Эль-Фа», 1996. 96 с.

7. Окулов, С. М. Информатика: Развитие интеллекта школьников [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2005. 212 с.; Васенина, Е. А. Общение на уроке информатики [Текст] / Е. А. Васенина, С. М. Окулов // Информатика и образование. 2004. № 8.

8. Соболева, Е. В. Тренинг педагогического общения для учителей информатики [Текст] / Е. В. Соболева // Вестник ВятГГУ. Информатика. Математика. Язык. Вып. 4, 2007. С. 83–86.

9. Кан-Калик, В. А. Педагогическое общение как предмет теоретического и прикладного исследования...

10. Соболева, Е. В. Указ. соч.

11. Петровская, Л. А. Компетентность в общении. Социально-психологический тренинг [Текст] / Л. А. Петровская. М.: Изд-во МГУ, 1989. 216 с.

12. Соболева, Е. В. Указ. соч.

ПОСТРОЕНИЕ ФОРМАЛЬНОЙ МОДЕЛИ ОПИСАНИЯ ВЗАИМОДЕЙСТВИЯ УЧИТЕЛЯ И УЧЕНИКА НА УРОКЕ ИНФОРМАТИКИ

В данной статье предлагается вариант формального описания схемы взаимодействия учителя и ученика на компьютерном уроке, сопровождающего самостоятельную деятельность последнего в целях его интеллектуального развития.

Одним из важнейших направлений личностно-ориентированного образования является развитие интеллекта школьников. При этом общение как вид деятельности, организуемый учителем на уроке, может являться одним из методов интеллектуального воспитания.

Ранее в литературе [1] уже затрагивалась тема обеспечения условий для эффективного интеллектуального развития учащихся в процессе обучения информатике и отмечалось, что общение

а) позволяет индивидуализировать учебный процесс, дает учителю возможность регулярной диагностики уровня интеллектуального развития ребенка;

б) позволяет не только констатировать какие-то факты относительно уровня образовательных достижений, но и обеспечивает возможность активного индивидуального «точечного» воздействия на процесс обучения с целью управления им для получения максимального эффекта.

По какой же схеме наиболее рационально выстроить общение между учителем и учеником? Не будет ли такой процесс случайным? И как можно в таком случае управлять случайными процессами и в целом случайностью?

Да, оказывается, можно, и подобные процессы в теории вероятности так и называются – *управляемыми* [2].

Все мы играли в детстве в игру «горячо – холодно». Заключается она в том, что водящему завязывают глаза, выводят на середину комнаты; несколько раз вращают до потери ориентировки и потом заставляют найти кого-нибудь из играющих, стоящих у стен комнаты. После каждого шага водящего присутствующие оживлённо комментируют правильность его действий и корректируют направление, подавая подсказки «горячо – холодно». Каждый шаг водящий делает в случайном направлении, но случайность здесь какая-то неполная (частично управляемая).

Весь процесс игры необычайно напоминает процесс познания (деятельность по решению задачи), где в качестве водящего выступает уче-

ник, а учитель говорит «правильно – неправильно». Каждый шаг, каждое изменение состояния в таком управляемом процессе определяется не только настоящим состоянием, но и качеством подсказки.

Таким образом, определяя соотношение процессов случайного и управляемого, можно выделить:

- а) случайные процессы;
- б) управляемые процессы;
- с) случайность + управление.

Последний вид процессов получил широкое распространение в науке, что определило необходимость выбора инструмента для описания формальной схемы взаимодействия элементов соответствующих систем.

Подобная необходимость существует и при описании схемы педагогического общения, которое можно рассматривать как процесс взаимодействия систем «Учитель» и «Ученик». Применение математического инструмента для моделирования этого процесса взаимодействия позволяет отбросить несущественные, случайные элементы и сконцентрировать внимание на основных, значимых свойствах и связях.

Конечно, каждый учитель по-своему планирует и организует общение с учеником. Кроме того, не будем забывать и о том, что общение – это всегда творческий процесс, который, по мнению В. А. Кан-Калика [3], предполагает:

1. Творчество учителя в процессе подготовки к деятельности и общению с учениками.
2. Творчество в ходе познания учителем учащихся, в системе взаимодействия с ними.
3. Творчество в организации непосредственного воздействия на ребёнка (система регуляции его поведения, реализация процедур вербальных и невербальных форм взаимодействия).
4. Творчество в управлении собственным поведением (саморегуляция общения).
5. Творчество при организации взаимоотношений.

Воплощённое в педагогической деятельности, общение, таким образом, и как процесс решения учителем бесчисленного множества коммуникативных задач, и как его результат является профессиональной категорией.

Если предложить некоторую общую схему взаимодействия и инструмент для построения соответствующей формальной модели, то это:

– с одной стороны, оказывало бы существенную помощь в планировании и организации общения с учеником тем учителям, у которых эта сторона образовательного процесса вызывает затруднения;

– с другой стороны, учитель мог бы творчески преобразовывать такую схему в зависимости от особенностей управляемой системы (класса)

и от собственных коммуникативных умений, которая бы способствовала качественным изменениям мыслительной деятельности ученика.

Речь идёт о том, чтобы описать деятельность учителя по созданию условий для максимально-го приближения обучения к реальному процессу познания через усиление самостоятельности мысли, к которой приводила бы самостоятельность действий.

Попытки описания схемы взаимодействия учителя и ученика на уроке, сопровождающего самостоятельную деятельность последнего, в целях его интеллектуального развития уже предпринимались Е. А. Васениной в статье «Компьютер на уроке информатики как инструмент индивидуализации интеллектуального развития школьников» [4]. Автором введена в рассмотрение система «Ученик», находящаяся в некотором состоянии, определяемом уровнем его учебных достижений. Из множества всевозможных состояний такой системы можно выбрать наиболее вероятные (прогнозируемые). И далее учитель подбирает и осуществляет систему воздействий, которые могут перевести систему «Ученик» в желаемое состояние. В заключение приводится общая схема перехода, основанная на постоянной диагностике состояния системы и анализе получившихся отклонений. Кроме того, выделяются некоторые способы непосредственного взаимодействия учителя с учеником, которые могут помочь в проведении диагностики.

Попытаемся развить предложенное Е. А. Васениной формализованное описание, придав ему больше строгости, для чего воспользуемся математическим инструментом моделирования состояний систем. Обратим особое внимание на то, что в предлагаемой далее схеме (постановка задачи, обсуждение, исследование проблемы и исправление ошибок) эти и другие способы взаимодействия вплетаются в ткань общения на уроке и осуществляются практически всё занятие, но не одинаково со всеми учащимися, а с каждым учеником индивидуально.

Вначале определим основные понятия, которые будем использовать при моделировании. Через $P = \{P_1, P_2, \dots, P_n\}$ зададим конечное множество состояний для системы «Ученик». А соответствующие воздействия учителя будем обозначать через конечное множество $T = \{T_1, T_2, \dots, T_m\}$. Кружок является позицией, а планка – переходом. Ориентированные дуги соединяют позиции и переходы. Дуга, направленная от позиции P_1 к переходу T_1 , определяет позицию, которая называется входом перехода. Выходная позиция указывается дугой от перехода к позиции [5].

В самом начале урока ученик находится в состоянии ожидания до тех пор – начальное состояние P_1 , пока ему учитель не объяснит теоретичес-

кий материал и не предложит задачу для самостоятельного решения – состояние размышления P_2 . После этого ученик выполняет задачу и предоставляет её учителю на проверку – состояние P_3 .

В качестве примера рассмотрим схему взаимодействия, моделирующую возможные события и состояния для системы «Ученик» [6] с конкретного фрагмента урока при изучении темы «Конструкция ветвления». Итак, учителем была сформулирована следующая задача: составить программу, запрашивающую у пользователя число x , по которому затем вычисляет значение выражения $y = 2 \cdot x^2 + 3 \cdot \sqrt{x}$ и полученный результат выводит на экран.

Задача при этом представляется в традиционной формулировке, которая для учеников привычна, но не всегда способствует дальнейшей мыслительной деятельности.

Как вариант – учитель разрабатывает диалоговую программу, которая представляет собой уже полученное решение. Для организации общения в этом случае предлагается воспользоваться методом демонстрации результата будущей работы. Такую деятельность можно рассматривать как способ реконструкции задачи, когда ученику требуется по результату определить план, составить алгоритм и подобрать средства для решения задачи, а затем получить похожий результат.

Интересно, что запускается программа пока только для положительных чисел, чтобы скрыть от учеников использование ветвления. Это позволит, с одной стороны, ученикам с более высокими способностями к анализу самим определить проблему для отрицательных чисел (маленькое «открытие», но своё). С другой стороны, тестирование диалоговой программы для отрицательных чисел уже на стадии проверки собственной программы ученика является действенным способом воздействия.

После этого ученики получают распечатку протокола диалога и работают самостоятельно. К тексту программы они доступа не имеют.

Если рассматривать самый благоприятный исход (т. е. когда ученик всё понял в теории, у него не возникает вопросов по решению задачи), то модель деятельности системы «Ученик» можно проиллюстрировать следующим образом (рис. 1):

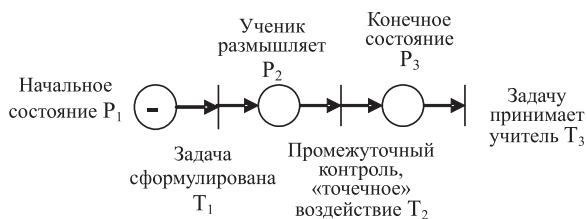


Рис. 1

Однако ситуации, подобные данной, встречаются нечасто. Они моделируют действия тех учеников, чей уровень образовательных достижений и уровень интеллектуального развития достаточно высокий (таких учащихся, согласно классификации, предложенной ранее [7], отнесём к группе Advanced).

Рассмотрим общий случай. Пусть для системы «Ученик» определено некоторое начальное состояние $P_1 \in P$, которое соответствует состоянию ученика на момент начала урока (рис. 2).

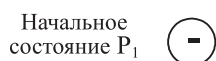


Рис. 2

Учитель объясняет материал, формулирует задачу для самостоятельного решения, т. е. запускает процесс (соответствующий переходу $T_1 \in T$).

Система «Ученик» переходит в состояние ожидания (размышления, озарения) – $P_2 \in P$ (рис. 3).

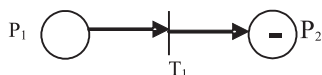


Рис. 3

Для некоторых это состояние может и не вызвать значительных трудностей. Но для тех учащихся, у которых индивидуальный темп работы, уровень самостоятельности в поиске решения не высок (т. е. средние и слабые ученики, которых дальше будем относить к группам Minimal и General [8]), это состояние, если не оказать необходимое воздействие, может стать завершающим. Выделим наиболее частые ошибки и затруднения, которые возникают у них на данном этапе работы, и предложим возможные пути решения.

Первые проблемы возникают на этапе описания переменных: либо обе переменные x и y объявляют целочисленными, либо (обращая внимание на корень) переменную y как вещественную, а x как целую

Некоторые из ребят по прошествии какого-то времени могут поднять руку и попросить объяснить, например: «Как это “запрашивающую число”? Что за оператор надо использовать? Сколько переменных понадобится? И вообще, я не понимаю это ваше программирование!». Или могут молчать, сидеть и с упорством смотреть в компьютер, что-то набирать, но в результате максимум, что имеют:

```
Program My;
  Var x,y:Integer;
Begin
End.
```

Далее, кто-то из учащихся начнет листать тетради или учебник, смотреть ранее решённые задачи. Их вопросы более существенные: какой оператор использовать – read или readln, как организовать вывод результата на экран?

На этом этапе работы, когда собственной программы у учеников ещё нет, неоценимую помощь и оказывает разработанная учителем диалоговая программа. На ней достаточно просто можно показать:

- как это – «запросить число»;
- что работает она не только, например, для набора $x = 3$; $x = 5$; $x = 10$, но и для числа $x = 3.5$ (прямой выход на вещественный характер переменной x);
- в какой форме результат должен быть выведен на экране.

Определить же тип переменной y можно по результату действия программы (например, для $x = 5$ получается $y = 27,24$) или отложить до запуска собственной программы, которая и выдает соответствующее сообщение.

Таким образом, учителем запускается один из соответствующих процессов – переходы $T_{21} \in T$ (для ученика группы Minimal) и $T_{22} \in T$ (для ученика группы General) к состоянию $P_3 \in P$. Проиллюстрируем соответствующие изменения на рис. 4.

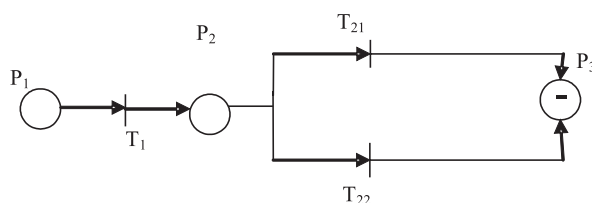


Рис. 4

В состоянии $P_3 \in P$ (составление алгоритма, задание команд ввода-вывода, запись условия) учащиеся вновь могут испытывать трудности:

1. Неправильно применяют команды ввода-вывода.
2. Делают синтаксические ошибки.
3. Путают функции $\text{sqr}(x)$ и $\text{sqrt}(x)$.
4. Результат выводят в виде записи с плавающей точкой (например, 9E-6).
5. Использование условия «если... то».

На ошибки первого и второго типа ученикам указывает компьютер при запуске программы.

Ошибки третьего типа компьютер не распознает, так как ему всё равно, что вычислять – квадрат или корень квадратный. В этом случае учителю опять пригодится подготовленный набор исходных данных, который и укажет на ошибку.

Для исправления ошибок четвёртого типа не лишней будет помощь, подсказка учителя, так как большинство учеников ту часть теоретического материала, которая касается вывода вещественных чисел на экран, не помнит или не до конца понял. А здесь – прекрасная возможность и для повторения, и для закрепления.

Но самая распространённая ошибка – это непонимание того, где надо использовать ветвление (более того, чаще всего ученики даже и не связывают тему урока с конкретной задачей). Остановимся на последнем виде ошибки. Здесь можно рассмотреть два варианта воздействия (моделей общения) со стороны педагога.

Самым неудачным выбором воздействия ($T_3 \in T$) будет схема общения, по которой учитель, не объясняя почему, просто скажет, что задача выполнена неправильно (для этого запустит программу, например, при $x = -2$), и добавит необходимое условие в программу. Если у учащегося способность к анализу выражена слабо или уровень самостоятельности в переносе теоретических знаний на практику невысокий, то система «Ученик» попадает в тупиковое состояние (рис. 5).

Если учащийся завершает работу в состоянии $P_3 \in P$ (нет качественного роста ни в образовательных достижениях, ни в интеллектуальном развитии), то это показатель того, что учителю необходимо пересмотреть методы работы с данным учеником, внести коррективы в свой «банк информации» об особенностях протекания его интеллектуальной деятельности, может быть, составить для него индивидуальный план работы.

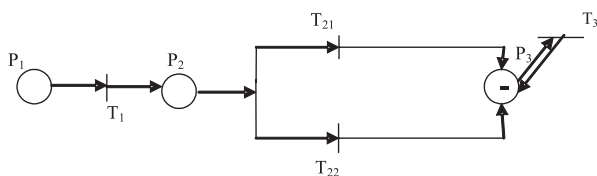


Рис. 5

Зная индивидуальные качества личности ученика и особенности его стиля мышления, можно выбрать другой вариант общения ($T_4 \in T$). Конечно, надо показать, что программа не будет работать для отрицательных чисел – подобрать набор соответствующих значений. Но не стоит и забывать про одно из правил общения – «не тыкайте ученика лицом в грязь». Ученик уже проделал большую работу, и в любом случае его надо похвалить, поощрить, и это требуется сделать обязательно.

Исправление ошибки можно начать с реконструкции темы и теоретического материала уро-

ка, а затем вместе посмотреть, используется ли где-нибудь у него в программе конструкция ветвления. Если ученик не сразу понимает, где надо применять изученную конструкцию, то следует обратить его внимание на вычисляемое выражение и обсудить ещё раз составные части – x^2 и \sqrt{x} . При вычислении какого из этих двух выражений может возникнуть ошибка, если вводим отрицательное значение x ? Полезно будет также вслух проговорить и выделить интонацией условие – «если».

Достоинством указанного диалога является то, что, с одной стороны, ученик проявил (пусть и небольшие) способности к анализу, обобщению, поиску информации, осуществил перенос теоретических знаний на практику, активно участвовал в поиске решения задачи; а с другой стороны, воздействие учителя не вызвало отрицательных эмоций и чувств. Как правило, после такого успеха ученик «рвётся в бой» и просит следующее задание. Таким образом, система «Ученик» переходит в качественно новое состояние (рис. 6).

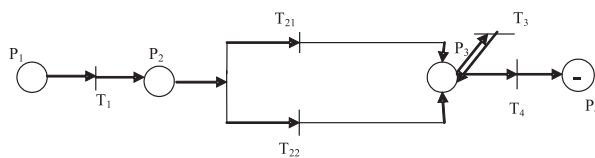


Рис. 6

В результате для системы «Ученик» (рис. 6) удалось построить управляемую схему общения, которая учитывает индивидуальные особенности протекания познавательных процессов и интеллектуальные характеристики личности каждого школьника и способствует его дальнейшему интеллектуальному развитию (в данном примере произошёл переход из состояния $P_1 \in P$ в качественно новое состояние $P_4 \in P$). Определяющими условиями для моделирования такой схемы общения являлись знания об уровне, достигнутом учеником, об особенностях его познавательной деятельности, о «слабых» и «сильных» качествах его ума, которые непосредственно используются во взаимодействии учителя и ученика.

В психологии для описания индивидуальных познавательных особенностей личности используется понятие познавательного стиля. М. А. Холодная в своём исследовании, посвящённом когнитивным стилям [9], определяет, что познавательные стили «характеризуют индивидуально-своеобразные способы изучения реальности». Ею выделяются следующие виды стилей: стили кодирования информации, стили переработки информации, стили постановки и решения проблем,

стили познавательного отношения к миру. Ссылаясь на представителей нейролингвистического программирования, которые считают, «что неуспеваемость школьников во многом объясняется несовпадением метода обучения с присущим тому или иному ребёнку стилем кодирования информации... и стоит только привести в соответствие поведение учителя и стиля ученика, как тот начинает демонстрировать растущие успехи в учёбе» [10], автор формулирует тезис о том, что наличие индивидуально-своеобразных способов кодирования информации играют важную роль в интеллектуальном развитии личности.

Таким образом, можно сделать вывод, что учитель для максимальной индивидуализации интеллектуального развития школьника должен обладать информацией об особенностях его познавательного стиля.

В этом случае важной составляющей для организации процесса общения (и обучения в целом) будет являться способность учителя быть «на одной волне» с учеником. Это значит, что несмотря на то, что индивидуальные способы кодирования информации учителя и ученика могут не совпадать (как часто и бывает на практике), для учителя важно научиться не только понимать, с помощью каких субъективных средств учащийся в своём ментальном опыте воспроизводит окружающий мир, но и подстраиваться под него и выбирать необходимое управляющее воздействие. Эти субъективные средства и являются стилями кодирования информации.

Другими словами, знание и понимание особенностей организации ментального опыта (а значит, и особенностей стилей кодирования информации) может быть использовано учителем для анализа индивидуальных познавательных способностей ученика, а также и для осуществления педагогических воздействий в процессе непосредственного общения.

И в соответствии с различными модальностями ментального опыта выделяются следующие стили кодирования информации [11]:

- словесно-речевой способ кодирования информации (с помощью знаков);
- визуальный способ кодирования информации (с помощью зрительных образов);
- предметно-практический способ кодирования информации (в основе принятия и переработки информации лежат предметные действия);

– сенсорно-эмоциональный способ кодирования информации (с помощью сенсорно-эмоциональных впечатлений).

Таким образом, для визуала типичная познавательная позиция – смотреть, представлять; для аудиала – слушать, говорить; для кинестетика – действовать, чувствовать. Поэтому при построении схемы взаимодействия с каждым учеником необходимо учитывать ту модальность ментального опыта, которая лежит в основе его индивидуального стиля кодирования информации и определяет спектр управляющих воздействий.

В дальнейшем планируется продолжить работу по построению формальной модели описания взаимодействия учителя и ученика на уроке информатики, при этом уже будут учитываться указанные стили кодирования информации, способы постановки и решения учениками проблем познавательного отношения к миру.

Примечания

1. *Васенина, Е. А.* Компьютер на уроке информатики как инструмент индивидуализации интеллектуального развития школьников [Текст] / Е. А. Васенина // Вестник ВятГГУ. Информатика. Математика. Язык. Вып. 4. 2007. С. 15–22; *Окулов, С. М.* Информатика: Развитие интеллекта школьников [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2005; *Кротова, Н. А.* Индивидуализация обучения на уроках информатики [Текст] / Н. А. Кротова // Вестник ВятГГУ. Информатика. Математика. Язык. Вып. 4. 2007. С. 29–32; *Соболева, Е. В.* Тренинг педагогического общения для учителей информатики [Текст] / Е. В. Соболева // Вестник ВятГГУ. Информатика. Математика. Язык. Вып. 4. 2007. С. 83–86.
2. *Андреев, А. Н.* Эти замечательные цепи [Текст] / А. Н. Андреев, А. Я. Иоффе. М.: Знание, 1987.
3. *Кан-Калик, В. А.* Педагогическое общение как предмет теоретического и прикладного исследования [Текст] / В. А. Кан-Калик, Г. А. Ковалёв // Вопросы психологии. 1985. № 4. С. 9–16.
4. *Васенина, Е. А.* Указ. соч.
5. *Питерсон, Дж. Л.* Теория сетей Петри и моделирование систем [Текст] / Дж. Л. Питерсон; под ред. В. А. Горбатова; пер. с англ. М. В. Горбатовой. М.: Изд-во «Мир», 1984.
6. *Васенина, Е. А.* Указ. соч.
7. *Кротова, Н. А.* Указ. соч.
8. Там же.
9. *Холодная, М. А.* Когнитивные стили: о природе индивидуального ума [Текст] : учебное пособие / М. А. Холодная. М.: ПЕРСЕ, 2002.
10. Там же.
11. Там же.

О. В. Токмакова

ПРОБЛЕМЫ УПРАВЛЕНИЯ ПРОЦЕССОМ ОБУЧЕНИЯ НА УРОКАХ ИНФОРМАТИКИ

В статье раскрыты особенности реализации основных управленческих функций учителем информатики на учебных занятиях, показаны перспективные направления совершенствования урока информатики.

Многие ученые-дидакты считают, что основными способами улучшения образовательного процесса в современных условиях модернизации образования являются: комплексное планирование задач обучения, воспитания и развития детей; выделение главного, генерального в содержании учебного материала, его структурирование и концентрация на нем внимания обучаемых; конструирование оптимальной структуры учебного занятия в соответствии с дидактической целью, логикой конкретного типа учебного занятия и с учетом дидактических требований к составу и последовательности этапов занятия; обоснованный выбор методов, форм и средств обучения; выявление и учет индивидуальных особенностей обучаемых, создание условий для развития индивидуального потенциала каждого ребенка; определение оптимального объема и сложности домашнего задания и т. д. Собственно, этими вопросами занимается теория управления образовательными системами, которая и может подсказать путь приведения процесса обучения информатике к эффективным результатам.

Как отмечает Н. Ф. Талызина, для управления познавательной деятельностью учащихся преподавателю необходимо указать цели управления; установить исходное состояние управляемого процесса; определить программу воздействий, предусматривающую основные переходные состояния процесса; обеспечить систематический контроль за управляемым процессом, то есть систематическую обратную связь; обеспечить переработку информации, полученной по каналу обратной связи, выработать корректирующие воздействия и их реализовать.

Процесс обучения на учебном занятии по определению должен быть управляемым. Мы рассматриваем учителя информатики как управленца первого уровня, задача которого – реализация управленческих функций в полном объеме на учебном занятии.

В современной теории управления выделяются следующие управленческие функции: информационно-аналитическая, мотивационно-целевая,

планово-прогностическая, организационно-исполнительская, контрольно-диагностическая и коррекционно-регулятивная (П. И. Третьяков, Т. И. Шамова и др.). Раскроем сущность каждой управленческой функции по отношению к учебному занятию и проблемы их реализации на уроках информатики.

Начнем с *мотивационно-целевой*. Она направлена на формирование целей учащихся на основе мотива, возникшего из потребности в данном виде деятельности. Ее неполноценная реализация приводит к тому, что учащиеся не понимают значимости приобретаемых знаний, умений и навыков, учеба для них превращается в «натаскивание», и говорить при этом о качественном обучении не приходится, ведь если нет цели, то нет и результатов. Однако не секрет, что далеко не у всех учащихся на необходимом уровне сформированы познавательные мотивы, а соответствующие потребности у многих учащихся просто-напросто не являются приоритетными. Подобную ситуацию мы довольно часто наблюдали при посещении уроков информатики, в том числе и у опытных педагогов. Создавалось впечатление, что учитель уверен в том, что дети, придя на урок, уже тем самым готовы принять те цели, которые он сформулировал для них на основе программы учебной дисциплины. В анкетах учителя отмечали также, что учащиеся «должны, обязаны изучать то, что для них приготовил учитель». Такие установки прослеживались даже у педагогов, придерживающихся современной компетентностной парадигмы. В своих конспектах к уроку при формулировках триединой дидактической цели они использовали следующие словосочетания: «ученики должны изучить... продемонстрировать... суметь использовать...». Получается, что цели формулируются за учеников, они под нажимом авторитета учителя эти цели вынуждены принимать, несмотря на несформированные мотивы и прочее, а педагоги при этом утверждают, что на уроке идет формирование свободных личностей.

Информационно-аналитическая функция учителя в управлении процессом на уроке является основным инструментом управления. Информационные потоки на учебном занятии необходимо запускать, поддерживать, изучать, регулировать, контролировать, анализировать и пр. На уроках информатики полноценно реализовать эти действия у обычного учителя не всегда получается. Информатика как молодая наука и как учебная дисциплина находится в развитии. Ее основные категории, понятия, причинно-следственные связи, законы и закономерности, часто еще не успев сформироваться, подвергаются пересмотру и обновлению. Кроме того, учитель, в отличие от учеников, не всегда владеет современной

информацией (особенно практического характера). Существенное отставание наблюдается и в содержании школьных учебников по информатике: они просто не успевают за столь бурно развивающейся отраслью знаний. Все это наносит отпечаток на систему преподавания информатики в школе.

Следующая функция – *планово-прогностическая*. Это организационная основа управления. Суть организации любого процесса заключается в построении иерархической системы целей-задач-результатов. По нашим наблюдениям, проектирование результатов обучения является одной из самых значимых проблем в деятельности учителя информатики. Дело в том, что уровень знаний, умений и навыков у учащихся одного и того же класса, группы значительно отличается. В каждом классе есть дети, которые и в теории, и в практике знают намного больше учителя и есть учащиеся, которые с трудом представляют себе, что такое компьютер. Эту разницу практически невозможно устранить в течение школьного обучения информатике, хотя учителя информатики часто на уроках делают невозможное для массовой школы – процесс обучения информатике индивидуализируют почти на сто процентов. При таких различиях в изначальном уровне знаний, умений и навыков спрогнозировать конечные результаты обучения очень трудно. А еще труднее продумать путь получения спланированных результатов, то есть структуру урока. Она, скорее всего, будет не линейной, а разветвленной и в некоторых ситуациях даже циклической, а потому достаточно сложной для реализации.

Организационно-исполнительская функция определяется как вид деятельности по формированию и регулированию определенной системы организационных взаимодействий. Специфика урока информатики заключается в том, что большой процент времени учебного занятия ученик работает за компьютером индивидуально, теоретический материал учителя информатики обычно предпочитают объяснять фронтально. Таким образом, на традиционных уроках информатики преобладают индивидуальная и общеклассная формы организации познавательной деятельности учащихся, благодаря чему диалоговое пространство урока сужено, взаимообмен информацией между учащимися неэффективен. Выбирая метод обучения, учителя информатики не придают особого значения построению системы приемов на проблемной основе, если и используют проблемные задания, то чаще всего это единственный случай. Как правило, объясняя свой выбор, они апеллируют к тому, что необходимо всех научить хотя бы на «три», а проблематизация учебного материала по времени очень затратна.

Контрольно-диагностическая функция предполагает наличие обратной связи между учителем и учащимися на учебном занятии. Это необходимо для того, чтобы определить, решены ли конкретные задачи, достигнуты ли поставленные цели урока и учтены ли при этом индивидуальные особенности обучаемых. Урок информатики настолько проблематизирует реализацию учителем этой управленческой функции, что встает вопрос об эффективности контроля на уроке. Изначально на практической части учебного занятия по информатике учащиеся занимались по 2–3 человека за компьютером. Поэтому окончательную индивидуальную оценку за выполненную практическую работу невозможно было выставить. До сих пор эта проблема имеется во многих школах. Домашние контрольные работы в противовес мнению многих учителей информатики не являются выходом из этой ситуации, так как не у всех учащихся дома имеются компьютеры и соответствующие компьютерные программы. Да и учащиеся могут значительно облегчить выполнение работы дома, воспользовавшись интернетом или обратившись за помощью к друзьям. Письменные контрольные работы эффективны тогда, когда имеется качественная система заданий разного уровня. Из-за ее отсутствия по ряду тем учителя стремятся создать ее самостоятельно. Основным затруднением, возникающим при этом, является правильность определения уровня сложности задания. Устные опросы на уроках информатики также не достаточно эффективный способ контроля, так как вызов к доске сопряжен с необходимостью удержания внимания класса и одновременным контролем ответа вызванного ученика. «Сильный» ученик вызывает вполне естественную реакцию его одноклассников – «можно не слушать и готовиться к следующему вопросу учителя или вообще заняться своим делом». «Слабый» же ученик может отвлечь класс от выполнения целей урока. Итак, реализация и этой управленческой функции имеет достаточно высокие резервы совершенствования.

Функция регулятивно-коррекционная направлена на оперативные изменения в связи с выявленными в процессе контроля и диагностики проблемами. Особенности расположения индивидуальных рабочих мест в компьютерном кабинете не предполагают взаимодействия «учитель-класс» во время выполнения практической работы, что значительно затрудняет взаимообмен информацией и последующее регулирование процесса обучения. Специфика проблем, возникающих у учащихся во время выполнения задания за компьютером, имеет очень широкий спектр и требует от учителя достаточно быстрого ответа. Учитель не всегда может быстро определить, к ка-

кому кругу проблем относится заданный ребенком вопрос, и адекватно отреагировать. В принципе, вопрос может совершенно не касаться выполняемого задания, а без ответа на него задание выполнить невозможно. Например, задание связано с программированием какого-либо процесса, а выполнить его ученик не может из-за проблемы, связанной с работой операционной системы компьютера. Учителю информатики для реализации целей учебного занятия необходимо разбираться в самом широком круге проблем обучаемых.

Таким образом, в реализации учителем информатики ключевых управленческих функций на уроках имеются серьезные препятствия в виде недостаточно сформированных профессиональных умений и навыков, что, в свою очередь, требует коррекции системы обучения преподавателей информатики.

О. В. Токмакова

ОРГАНИЗАЦИЯ СИСТЕМЫ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ В ВУЗЕ

В статье проанализированы особенности организации системы дистанционного образования в ВятГГУ.

Образовательные потребности наших современников сопровождаются особыми требованиями к условиям обучения, такими, как доступность, открытость, экономичность, гибкость, демократичность, индивидуализация, наравне с потребностями в высоком уровне квалификации преподавателей, качественном образовании и пр. Такие требования трудно обеспечить только с помощью традиционных образовательных технологий. Нужны новые способы и средства работы с обучаемыми. В настоящее время в образовании имеет место интеграция образовательных и информационно-коммуникационных технологий, на основе которой и формируется новая группа – дистанционные образовательные технологии. Дистанционное обучение востребовано населением многих стран мира. Это свершившийся факт не только за рубежом, но и в нашей стране. Для демократического развития любой страны в эпоху информационного общества необходимо, чтобы каждый гражданин имел возможность выбора в разные периоды своей жизни в зависимости от складывающихся обстоятельств очной, очно-заочной, заочной форм обучения, экстер-

ната и различных их сочетаний с дистанционными образовательными технологиями.

Традиционно под дистанционными образовательными технологиями понимаются средства, методы и способы обучения, предполагающие постоянное, воспроизводимое формализованное представление, передачу и контроль знаний в открытой информационной среде. Информационно-коммуникационные технологии являются средством реализации дистанционных образовательных технологий.

В России внедрение новой группы образовательных технологий в учебный процесс, несмотря на безусловное успешное продвижение в этом направлении таких лидеров образования, как МЭСИ, УГТУ-УПИ и других, все-таки идет чрезвычайно медленно и не столь эффективно, как это могло бы быть. И дело здесь не только в инертности педагогического мышления или в непроработанности специальной группы нормативно-правовых актов на соответствующих уровнях.

В регионах, значительно отдаленных от центра, потребности в обучении с использованием дистанционных образовательных технологий в основном возникают у граждан, большой процент которых проживает в сельской местности. Как правило, информационно-коммуникационная инфраструктура в таких местах развита недостаточно: интернет-связь или вообще отсутствует, или качество ее оставляет желать лучшего. Как следствие большинство потенциальных студентов имеют очень небольшой опыт работы в открытых информационных пространствах. Кроме этого следует учесть и проблемы, связанные с подбором педагогических кадров для системы дистанционного образования. Тьютором может быть только преподаватель, владеющий информационно-коммуникационными технологиями, являющийся профессионалом высокого класса по своему учебному направлению и обладающий педагогическим даром организации разных форм взаимодействия и общения, умеющий заражать своими идеями на расстоянии. Согласитесь, такой уникальный педагог не столь уж и частое явление в нашей системе высшего образования, а приглашать преподавателей столичных вузов подчас слишком дорогое удовольствие, не приносящее вузу никаких экономических дивидендов, кроме рекламы. Всё это является серьезным препятствием для единовременного внедрения дистанционного образования в вузе сразу по всем направлениям и формам обучения. Следовательно, системная организация дистанционного образования в региональном вузе должна происходить на моделях, принципиально отличающихся от подобных моделей, осваиваемых центральными вузами страны. И ей скорее будет свойственна пошаговость и постепенность, чем крупномасштабность в реализации.

Дистанционное образование как система предусматривает различные модели реализации на базе очной, заочной, очно-заочной форм обучения и экстерната.

Очная форма обучения предполагает, например, использование дистанционных образовательных технологий в самообразовательной деятельности студентов. Примером могут послужить ситуации взаимодействия и общения студента с преподавателем во время работы над курсовым или дипломным проектом посредством электронной почты, различных сервисов общения (QIP или ICQ), общения по телефону. Большинство преподавателей, стремящихся не отставать от своих студентов в овладении современной техникой, с удовольствием осваивают новые информационно-коммуникационные технологии и находят наиболее оптимальные способы использования их в процессе обучения. Кроме того, прогрессивно мыслящие преподаватели активно разрабатывают разные варианты учебных пособий на электронной основе по своим учебным дисциплинам (курс лекций-презентаций, система компьютерных тестов, электронные учебные пособия с интерактивными модулями и видеофрагментами лекций, дистанционные учебные курсы и др.), рекомендуя студентам использовать эти материалы в процессе самостоятельной работы при подготовке к коллоквиумам, зачетам, экзаменам и прочим контрольным мероприятиям.

Очно-заочная форма – это вариант вечернего обучения. Процент самостоятельной работы студентов при ее использовании по сравнению с очной формой обучения значительно возрастает. Следовательно, усиливается потребность в помощи и поддержке студентов в этой деятельности. Электронные учебные пособия могут выступать при этом в качестве практического руководства самостоятельной работой студентов. Более того, систематизация учебно-методического материала преподавателем обязательно выведет на его оформление в виде кейса, за которым и последует использование в процессе обучения кейс-технологии.

Заочная форма предполагает, что студенты большую часть учебного материала изучают самостоятельно, во время сессий им «начитывается» определенное содержание, дается установка на самостоятельное его развитие и проработку, однако сам процесс самостоятельной работы, так же, как и в остальных формах, остается бесконтрольным, и преподаватель не является соучастником самостоятельного творчества студента. Не является секретом и низкий уровень качественной подготовки специалистов при использовании этой формы обучения. Дистанционные технологии позволяют вывести на совершенно иной уровень качество образования студентов заочной

формы обучения. Как правило, здесь хорошо работают и сетевые технологии, и TV-технологии, и различные их сочетания друг с другом и с кейс-технологией.

Особое место в этом ряду занимает *экстернат*. Нет сомнения в том, что дистанционные образовательные технологии в полной мере позволяют поднять результативность и качество образования студентов, решивших выбрать такую сложную форму получения образования. Можно с уверенностью утверждать, что реализация экстерната как более экономичной и прогрессивной формы обучения, чем, например, традиционная заочная, возможна с использованием всех групп дистанционных образовательных технологий и самых различных их сочетаний.

Итак, закономерно встают следующие вопросы: надо ли начинать внедрение дистанционных образовательных технологий одновременно во всех формах; и если нет, то с какой формы лучше всего начинать. Ответы можно получить, рассмотрев внедрение дистанционных образовательных технологий как инновационный процесс. Исходя из анализа практических вариантов реализации дистанционного образования в зарубежной системе высшего образования и у нас в стране, можно четко выделить три стратегии его становления и развития: незначительные локальные изменения; блочно-модульные и системные изменения.

Незначительные локальные изменения потому и являются незначительными, поскольку существенных прорывов в развитии здесь ожидать не приходится. Как правило, инновационный процесс идет фрагментарно, новаторы друг о друге почти ничего не знают, опытом не обмениваются. Эта стратегия практически не требует от вуза серьезных капиталовложений, ведь преподаватели-энтузиасты настолько увлечены новыми технологиями обучения, что способны поддерживать их реализацию самостоятельно. При этом если количество преподавателей, работающих в новом направлении, превысит критическую массу и у них появятся последователи, зараженные интересными идеями, то есть шанс перейти к более прогрессивной модели внедрения – блочно-модульным изменениям.

Вторая стратегия является промежуточной по «затратности» для вуза. Однако она позволяет создать довольно комфортные условия для постепенного развертывания дистанционной системы во всех формах обучения. Здесь предполагаются следующие основные этапы:

1. Создание минимальной ресурсной базы реализации системы дистанционного образования на одном или двух факультетах.

2. Реализация дистанционных образовательных технологий на нескольких специальностях.

3. Разработка и апробация моделей внедрения дистанционных технологий в образовательную систему вуза.

4. Теоретическое обобщение сформированного опыта, установление точек роста и развития построенных моделей.

Системное внедрение требует настолько значительных капиталовложений и огромной подготовительной работы по созданию ресурсной базы: нормативно-правовой, материально-технической, информационной, мотивационной, кадровой и прочей, – что подобное освоение инновации по силам вузам, финансовое состояние которых устойчиво к всевозможным инновационным рискам.

Таким образом, пренебрежение и невнимательность в выборе или построении модели реализации системы дистанционного образования на основе региональных условий, как правило, приводит или к полному «замиранию» инновационного процесса, или к его деградации и профанации самой идеи. Практические исследования моделей реализации дистанционного образования вузами, работающими в Кировской области, выполненные на этапе разработки программы внедрения системы дистанционного образования в ВятГГУ, полностью подтверждают наши выводы.

Д. А. Хомякова

ОБ ОПЫТЕ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА ПРИМЕРЕ РАЗРАБОТКИ ЭЛЕКТРОННОГО УЧЕБНОГО СРЕДСТВА ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Данная статья посвящена описанию процесса разработки электронного учебного средства для организации самостоятельной работы по решению задач на ЭВМ группой студентов-исследователей.

На сегодняшний день особую актуальность в обучении информатике приобретают такие прогрессивные подходы, как проблемно-деятельностный, личностно-ориентированный, задачный. Они требуют от преподавателя смещения акцента с передачи учащемуся готовых знаний, умений и навыков на интенсивное развитие его интеллекта посредством предъявления системы задач, среди которых достаточно большую часть составляют проблемные задачи.

Применение задачного подхода является наиболее эффективным в обучении программирова-

нию, его суть заключается в следующем. Новые знания приобретаются учеником в процессе выполняемой совместно с учителем работы по решению проблем, которые задаются системой специально подобранных учебных задач. Он органично соединяется с проблемным подходом к обучению. Преподаватель предъявляет учащемуся некоторую проблему (задачу) и вовлекает в деятельность по ее решению. При этом наличие некоторого противоречия мотивирует эту деятельность. Учащийся осознает недостаток имеющегося инструментария для решения задачи, анализирует существующие знания и генерирует новый способ действия в процессе активной мыслительной деятельности, тем самым самостоятельно развиваясь.

Реализация задачного подхода сопровождается рядом сложностей, в частности трудностью подбора задач. И не просто задач, а системы, обеспечивающей усвоение учащимся перечня знаний по соответствующей теме, формирование необходимых навыков и умений. На сегодняшний день существует большое количество различных печатных и электронных источников задач по программированию. В них задачи, как правило, разбиты на тематические блоки. Но нет четко определенных принципов классификации, не указано, какие именно умения формирует та или иная задача, и нет рекомендаций к тому, какому ученику ее следует предложить для решения.

Каждому преподавателю приходится самостоятельно отбирать задачи из всего существующего многообразия и формировать из них некоторую систему, опираясь при этом на субъективные представления. Осложняет ситуацию необходимость систематического составления индивидуальных заданий для студентов на каждое занятие (по каждой теме). Очевидно, что данные действия могут и должны быть инструментированы. Достижению этой цели в полной мере может служить специально созданное электронное учебное средство.

Каким должен быть программный продукт, призванный поддерживать процесс обучения программированию на основе задачного подхода? Во-первых, не стоит перекладывать на систему функцию сообщения новых знаний. К решению задач учащийся приступает с уже имеющимся базовым набором теоретических сведений по теме, которые будут расширяться и углубляться в процессе его деятельности. Безусловно, учебное средство должно содержать минимальный объем материала для обеспечения справочной поддержки. Но основой наполнения является банк задач, разделенных по темам и классифицированных таким образом, чтобы преподавателю было максимально удобно подбирать задачи и составлять

индивидуальные задания для каждого студента или микрогруппы. Учебное средство должно обеспечивать удобство выбора задач по определенным критериям и возможность пополнения базы. Студент, работая с системой, просматривает подобранные для него задачи, решает их, тестирует на различных входных данных созданные алгоритмы, может фиксировать результаты своей работы в отчете.

Разработка подобного электронного учебного средства позволила бы оптимизировать временные затраты преподавателей на подготовку к занятиям, а главное – значительно повысить продуктивность работы. Однако, несмотря на разнообразие функциональных возможностей системы, она должна исполнять роль помощника преподавателя, но не заменять его.

Реальная потребность в описанном электронном учебном средстве существовала на факультете информатики ВГГУ, ввиду чего преподавателями факультета был организован проект, в рамках которого данное средство и было разработано. Для этого была создана исследовательская группа из трех студентов: двух специалистов предметной области «Информатика и методика обучения информатике» и программиста-разработчика. Объем и направление работы каждого члена группы были четко определены. Так, задание специалистов предметной области – создание модели информационно-методического наполнения электронного учебного средства и ее реализация для тематических разделов курса. Программист-разработчик отвечал за электронную реализацию данной модели в виде учебной среды.

Процесс групповой разработки учебного средства представляет большой интерес как с точки зрения методики информатики, так и с позиции теории информационных систем. В данной статье остановимся на втором аспекте, касающемся разработки информационного продукта.

Так как электронное учебное средство представляет собой информационную систему, стадии его разработки соответствуют этапам жизненного цикла ИС. Таким образом, работа всех членов группы проводилась в несколько этапов: формирование концепции, подготовка технического задания, собственно проектирование, разработка и ввод в эксплуатацию. Теоретические основы проектирования ИС были адаптированы к реальному процессу создания электронного учебного средства, который на практике выглядит следующим образом.

1. Фаза формирования концепции

• Заказчиком проекта выступил его руководитель (преподаватель факультета информатики). Им была выдвинута идея и поставлена цель исследования.

• Согласно цели сформирована «ключевая команда проекта» – группа из 3 студентов V курса факультета информатики (А. Кавардаков, Д. Хомякова, И. Шаклеина).

• В ходе первоначальных консультаций заказчиком была четко обозначена проблема (отсутствие электронного обучающего средства для реализации самостоятельной работы студентов по решению задач на ЭВМ и потребность в нем), описаны основные требования, которым должна соответствовать информационная система.

• Сбор исходных данных и анализ существующего состояния проводился членами творческой группы совместно, в результате чего выяснено:

о На факультете информатики ВГГУ на I–II курсах специальностей ИНФ, ПМ и ПИ ведется преподавание практических курсов, направленных на обучение основам программирования, таких, как «Практикум по решению задач на ЭВМ», «Информатика и программирование» и других.

о Основу содержания обозначенного курса составляют задачи, предъявляемые студентам для решения.

о Большинство преподавателей, ведущих данный предмет, обладает собственной базой задач (в письменном или электронном виде), из которой они формируют содержание курса для каждой отдельной группы студентов.

о Используются на факультете подходы и идеи к обучению информатике и программированию: деятельностный, личностно-ориентированный, задачный подходы, идея развития интеллекта через обучение программированию.

о Для успешной реализации обозначенных подходов существует потребность в создании единого электронного банка задач, дополненного справочными теоретическими сведениями. Основная функция информационной системы – хранение банка классифицированных задач с возможностью составления индивидуальных электронных карточек с заданиями для отдельных студентов (микрогрупп студентов) и хранения информации об их деятельности по решению задач.

Анализ собранной информации позволил более четко сформулировать цели и задачи работы.

• Заказчиком были обозначены основные требования и ограничения по выполнению работы. Был разрешен доступ к компьютерному оборудованию и ПО, имеющемуся на факультете информатики.

• Программистом-разработчиком был проведен анализ существующих электронных учебных средств (оценка альтернатив), с позиций достигаемых с их помощью целей и решаемых задач, а

также содержания и вида представления информации, требований к пользователю. В результате выявлены достоинства и недостатки систем, сформулированы требования к разрабатываемому продукту.

- На завершающем этапе членами исследовательской группы разработан макет системы, в который заказчиком внесены коррективы и дополнения.

2. Подготовка технического задания

- Разработка базовой структуры проекта и основного содержания произведена специалистами в предметной области. Она включает в себя общую структуру, описание режимов работы программы и основных модулей электронной среды: модуль «Банк задач», модуль «Теория», модуль «Карточка для студента».

- Разработанный проект работ согласован с заказчиком.

- В ходе построения модели электронного учебного средства были выделены две основные подсистемы:

- о информационно-методическое наполнение ИС (блок теории, блок задач, блок тестов);

- о компьютерная реализация ИС (разработка профилей «Студент» и «Преподаватель» и отдельных модулей в них).

- Программистом-разработчиком были определены потребности в ресурсах. Задействован информационный ресурс Internet. Для реализации выбраны следующие программные продукты: Microsoft SQL Server 2005, Microsoft Visual Studio 2005.

- Совместно с заказчиком определен план консультаций и объем работ для выполнения к каждой консультации (календарное планирование).

3. Проектирование

- Выполнение базовых проектных работ осуществлялось параллельно всеми членами группы и заключалось в разработке моделей отдельных подсистем ИС.

- В ходе анализа и разработки моделей специалистами предметной области внесен ряд замечаний и предложений в первоначальный план работы. Например, был уточнен способ хранения в базе данных банка теории и задач. Программист-разработчик принял к сведению замечания от специалистов предметной области и внес исправления.

В свою очередь он сообщил о невозможности компьютерной реализации некоторых первоначально заложенных в модель функций ИС. Так, например, оказалось сложным создавать отдельные «карточки» для каждого студента. Наиболее удобным разработчики сочли способ отображения списка назначенных студенту задач в виде таблицы. Принадлежность той или иной задачи

к конкретной «карточке» (то есть к конкретному занятию или теме) отражается посредством даты, на которую задача назначена. Таким образом, при выборе метода сортировки задач по дате студент может легко определить, какие задачи ему требуется прорешать на текущем занятии.

На этапе разработки частных технических заданий некоторые функции ИС были переформулированы, от несущественных моментов пришлось отказаться.

- Концептуальное проектирование состояло в уточнении модели информационной среды и ее детальном описании.

- Составление технических спецификаций и инструкций – совместная работа членов творческой группы.

- Разработанный окончательный проект работ был согласован с заказчиком, и исследовательская группа перешла к этапу разработки.

4. Разработка

- Программистом-разработчиком выполнена разработка программного обеспечения.

- Подготовка системы к внедрению заключалась в добавлении информационно-методического наполнения к разработанной программной оболочке.

- Тестирование и отладка системы производилась программистом-разработчиком на месте разработки. Неплоадки и неточности в работе устранены.

- 5. Ввод системы в эксплуатацию включает следующие этапы:

- Комплексное испытание.

Предполагает установку информационной среды на компьютеры факультета информатики и проведение серии занятий с системой для студентов I–II курсов (на протяжении минимум одного учебного семестра).

- Подготовка кадров для эксплуатации системы.

Предполагает обучение преподавателей факультета информатики работе с информационной системой.

- Ввод в эксплуатацию с последующим сопровождением, поддержкой и обслуживанием.

Предполагает постоянную поддержку работоспособности системы в течение периода эксплуатации.

- Оценка результатов проекта и подготовка итоговых документов.

В полном объеме может быть осуществлена не ранее окончания курса (нескольких курсов), полностью проведенного с использованием электронного обучающего средства.

- Разрешение конфликтных ситуаций и закрытие работ по проекту.

- Накопление опытных данных для последующих проектов.

В ходе разработки электронного учебного средства творческой группой специалистов были успешно пройдены 4 фазы проектирования информационной системы: формирование концепции, подготовка технического задания, собственно проектирование и разработка. Работа была проведена в соответствии с планом, и промежуточные цели всех четырех фаз достигнуты. Безусловно, разработчики столкнулись с рядом проблем. Но благодаря грамотному поэтапному выполнению работы их возникновение также было предусмотрено планом. Поэтому решение трудностей не потребовало возврата к пройденным этапам.

Во временных рамках, отведенных на выполнение проекта, оказалось невозможным реализовать фазу ввода продукта в эксплуатацию, поскольку последняя включает в себя комплексное испытание, подготовку кадров для эксплуатации системы, ввод в эксплуатацию с последующим систематическим сопровождением, поддержкой и обслуживанием. Данная фаза занимает столько времени, сколько разработанная информационная система будет эксплуатироваться. Поэтому внедрение предоставлено самим заказчикам проекта, то есть преподавателям.

Разработанное электронное учебное средство представляет собой сетевое приложение и устанавливается на компьютеры, объединенные локальной сетью. Реализованы два профиля: «Преподаватель» и «Студент» – с определенным набором доступных функций. Каждый учащийся заходит в систему под собственным именем и паролем, имеет возможность решать задачи и сохранять результаты своей работы в отчете. Преподавателю предоставляется возможность подготавливать для каждого пользователя набор задач и следить за ходом их решения, просматривая отчеты.

В соответствии с целью создания данное учебное средство будет применяться на факультете информатики ВГГУ для обучения студентов первых курсов основам программирования.

Р. В. Шарьгин

ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ

В статье рассматриваются такие характеристики производительности параллельных программ, как ускорение, эффективность, стоимость вычислений. Вводятся понятия масштабируемости параллельных программ, функции изоэффективности. Описания конкретных характеристик содержат иллюстративные примеры.

Последовательные алгоритмы часто оцениваются с точки зрения времени их выполнения, которое выражается функцией от размера решаемой задачи. Время выполнения параллельных алгоритмов зависит не только от размера решаемой задачи, а еще от числа используемых процессоров, времени взаимодействия между различными частями задачи и скорости такого взаимодействия. Поэтому параллельные алгоритмы не могут оцениваться в полной изоляции от параллельной архитектуры без потери точности в оценке. В этой статье мы рассмотрим различные характеристики производительности параллельных программ.

Количество характеристик производительности очень разнообразно. Самой простой из характеристик можно назвать время выполнения данной задачи на данной параллельной архитектуре. Однако такую оценку очень трудно экстраполировать на другую задачу или параллельную архитектуру. Другая характеристика измеряет выигрыш от введенного параллелизма, т. е. насколько быстрее исполняется параллельная программа по отношению к последовательной программе. Однако и у такой оценки есть недостаток, ее также сложно экстраполировать на другую параллельную архитектуру или задачу. Какой выигрыш мы получим от использования не очень быстрого последовательного алгоритма, который очень легко распараллелить? Для того чтобы ответить на этот вопрос, необходимо ввести более сложные характеристики оценки производительности параллельных систем.

1. Накладные расходы в параллельных программах

Увеличив вдвое доступные ресурсы для решения задачи, мы ожидаем, что время выполнения нашей программы будет вдвое уменьшено. Однако это происходит довольно редко ввиду различных накладных расходов (рис. 1). Правиль-

ШАРЬГИН Роман Владимирович – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Шарьгин Р. В., 2008

ная оценка накладных расходов имеет немало-важное значение для оценки производительности параллельных программ.



Рис. 1. Профиль выполнения некоторой параллельной программы на четырех процессорах

При решении задачи наша параллельная программа выполняет не только основные вычисления, т. е. те вычисления, которые бы выполняла аналогичная последовательная программа. Дополнительное время параллельная программа тратит на межпроцессное взаимодействие (interprocess interaction), ожидание (idling), избыточные вычисления (excess computation), т. е. вычисления, которые отсутствуют в аналогичной последовательной программе [1].

Межпроцессное взаимодействие. Большинству параллельных программ необходимо обмениваться данными между отдельными ветвями вычислений, будь то потоки или отдельные процессы. Время, которое проводит программа, выполняя подобное взаимодействие, составляет значительную часть накладных расходов.

Ожидание (или время простоя). Отдельные ветви вычислений могут простаивать в процессе выполнения программы по многим причинам, например дисбаланса загрузки отдельных ветвей или подзадач программы, в результате необходимых синхронизаций, присутствия последовательных вычислений в программе и многих других причин. Например, во многих параллельных программах, где процесс генерации подзадач является динамическим, невозможно или крайне сложно заранее предугадать размер выделяемой подзадачи. Если ветви вычислений имеют различный объем работ, то некоторые из них могут простаивать, пока другие работают над решением задачи. В некоторых параллельных программах ветви должны выполнять синхронизацию на определенных этапах вычислений, если некоторые ветви не готовы к синхронизации в определенный момент времени, то те ветви, которые закончили вычисления раньше, будут простаивать, ожидая готовности остальных веток или подзадач

программы. Часто бывает так, что часть вычислений не может быть распараллелена, поэтому ветви вычислений должны ожидать завершения той ветви или подзадачи, которая производит последовательные вычисления.

Избыточные вычисления. Порой самый быстрый из последовательных алгоритмов трудно или невозможно распараллелить, поэтому приходится использовать не столь быстрый, но легко распараллеливаемый последовательный алгоритм (алгоритм с более высокой степенью параллельности) для решения задачи. Разница в количестве производимых вычислений параллельной программой и лучшей последовательной программой называется избыточными вычислениями, вносимыми параллельной программой.

Так как параллельные алгоритмы для решения одной и той же задачи могут обладать разным количеством накладных расходов, то очень важно иметь оценку накладных расходов для каждого алгоритма.

2. Характеристики производительности параллельных программ

Характеристики производительности параллельных программ очень важны, так как они позволяют определять наилучший алгоритм, оценивать параллельную платформу и преимущества от введения параллелизма.

2.1. Время исполнения

Время выполнения последовательной программы T_s и параллельной T_p определяется разницей во времени между началом и концом исполнения последовательного решения задачи и параллельного соответственно. Следует сказать, что мы не включаем в T_s и T_p время, затраченное на подготовку данных для обработки (ввод и вывод, например), конечно, если подготовка данных не является частью алгоритма решения задачи.

2.2. Накладные расходы

Определим накладные расходы (overhead) параллельных вычислений T_o как разницу между суммарным временем работы параллельного решения на каждом вычислительном элементе и временем самого быстрого последовательного решения на одном вычислительном элементе. Суммарное время, затраченное на решение проблемы на всех вычислительных элементах, есть $p \cdot T_p$, поэтому накладные расходы можно выразить таким образом:

$$T_o = p \cdot T_p - T_s, \quad (1)$$

2.3. Ускорение

При оценке параллельных вычислений мы часто заинтересованы в том, чтобы знать, какой прирост производительности мы получим, если распараллелим наше последовательное решение. Ускорение (speedup) S есть величина, которая отражает пользу от внесения параллелизма в решение задачи. Эта величина выражается как отношение времени решения на одном вычислительном элементе ко времени решения на p идентичных вычислительных элементах

$$S = \frac{T_s}{T_p}. \quad (2)$$

Пример 1. Сложение n чисел с использованием p вычислительных элементов.

Допустим, что мы решаем задачу сложения n чисел на p вычислительных элементах. Перед началом вычислений все числа распределены по вычислительным элементам, к концу вычислений сумма элементов размещается на каком-то одном вычислительном элементе. В предположении, что n есть степень двойки, мы можем вычислить сумму за $\log n$ шагов, передавая частичные суммы по узлам бинарного дерева вверх, к корню (рис. 2).

Каждый шаг вычислений состоит из одного сложения и одной операции пересылки данных. Время выполнения данной операции есть константа. Поэтому

$$T_p = \Theta(\log n).$$

Сложение n чисел на одном вычислительном элементе может быть выполнено за время $I(n)$, поэтому ускорение равно

$$S = \Theta\left(\frac{n}{\log n}\right).$$

Для данной проблемы может существовать более одного последовательного алгоритма её решения, однако не каждое из них может быть успешно распараллелено. Когда нам для решения задачи доступен только один вычислительный элемент, то естественно использовать последовательный алгоритм, который решает эту задачу за наименьшее время. Для параллельного алгоритма будет справедливым оценивать его производительность в сравнении как раз с таким наибо́льшим последовательным алгоритмом, который решает ту же самую проблему на од-

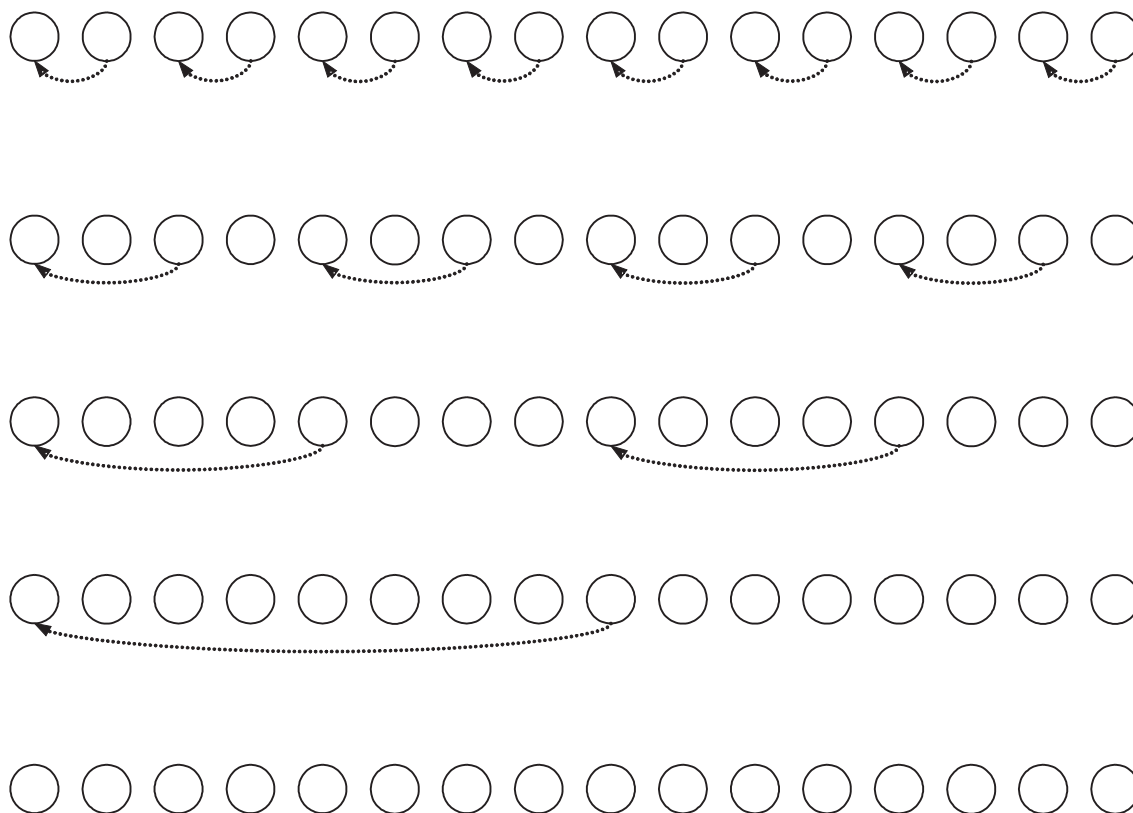


Рис. 2

ном вычислительном элементе. Иногда асимптотически самый быстрый последовательный алгоритм не известен или оценка времени его исполнения содержит большие константы, которые на практике делают этот алгоритм неприменимым для решения задачи. Тогда для оценки производительности параллельного алгоритма мы берем самый быстрый известный нам последовательный алгоритм, который является таковым и на практике. Формально ускорение S – это отношение времени исполнения лучшего последовательного алгоритма для решения задачи ко времени исполнения параллельного алгоритма для решения той же самой задачи на p вычислительных элементах, идентичных вычислительному элементу, который используется в последовательном варианте решения [2].

Пример 2. Вычисление ускорений параллельной программы сортировки чисел.

Возьмем «пузырьковую» сортировку (bubble sort). Предположим, что последовательная сортировка некоторого множества чисел занимает 200 единиц времени, а последовательная «быстрая» сортировка (quick sort) тратит на упорядочивание этого набора чисел 40 единиц времени. Параллельная версия «пузырьковой» сортировки, еще называемая сортировкой четно-нечетными перестановками, занимает 60 единиц времени на четырех вычислительных элементах. Кажется, что мы достигли ускорения в $3,33 = 200/60$. Однако в реальности ускорение равно $0,67 = 40/60$.

Теоретически ускорение никогда не превзойдет количества вычислительных элементов p . Если время решения данной задачи для лучшего последовательного алгоритма T_s , то ускорение p на p вычислительных элементах может быть достигнуто лишь при условии, что для решения задачи каждый вычислительный элемент тратит время

не превосходящее $\frac{T_s}{p}$. Ускорение свыше p достижимо лишь в том случае, когда каждый вычислительный элемент для решения данной задачи

тратит время меньше, чем $\frac{T_s}{p}$. В этом случае изменим последовательный алгоритм решения задачи таким образом, чтобы он эмулировал решение задачи на p вычислительных элементах, что позволит нам решить задачу за время меньше T_s . Получили противоречие с тем, что T_s есть время решения данной задачи лучшим последовательным алгоритмом.

На практике, однако, можно наблюдать ускорение, которое превосходит количество вычислительных элементов p . Такое явление называют сверхлинейным ускорением (superlinear speedup) [3]. Обычно это случается, когда последователь-

ный алгоритм выполняет больший объем работ в сравнении с параллельным алгоритмом или в результате преимуществ, которые получает параллельный алгоритм при исполнении на параллельной платформе. Например, при решении задачи на одном процессоре оказывается недостаточно объема памяти кэша первого уровня (L1 cache), тогда появляется необходимость обращения к более медленной внешней памяти. В случае же использования нескольких процессоров объема памяти кэша первого уровня может оказаться достаточным для обработки данных за счет разделения данных между процессорами. Источником сверхлинейного ускорения также может быть и различие схем последовательного и параллельного методов решения задачи.

Пример 3. Сверхлинейное ускорение.

Возьмем дерево, узлы которого помечены некоторыми значениями. Будем искать узел с меткой «X». Для поиска заданного узла используем обход в глубину. Последовательное решение обойдет все 14 узлов дерева, перед тем как найти заданный узел (рис. 3). Пусть посещение узла дерева занимает некоторое константное время t_c , тогда общее время будет равно $14 \cdot t_c$. Сейчас пусть нам доступны два вычислительных элемента. Разделим дерево на два поддерева и выполним параллельный поиск по ним на двух вычислительных элементах. Теперь, перед тем как найти заданный узел, мы обойдем только 9 узлов дерева, а если учесть параллельное выполнение поиска, то общее время поиска равно $5 \cdot t_c$. Ускорение, которое мы достигли на данном дереве,

$$\text{равно } \frac{14 \cdot t_c}{5 \cdot t_c} = 2,8 .$$

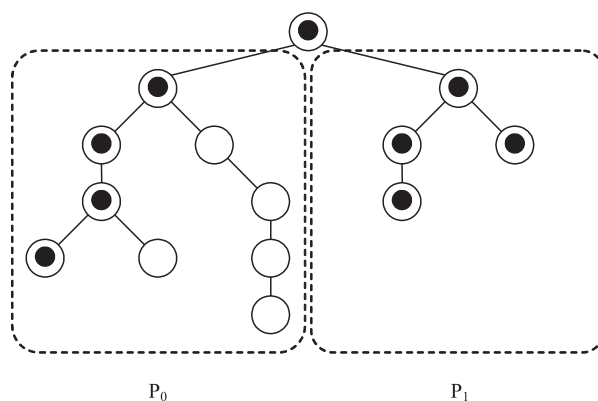


Рис. 3. Поиск заданного узла в дереве. Последовательное и параллельное решение с использованием двух вычислительных элементов

2.4. Эффективность

Только идеальные параллельные вычисления могут достигать ускорения p , используя p вычислительных элементов. В реальности такая ситуация редко достижима, потому что вычислительные элементы не могут выполнять одни лишь вычисления, какая-то часть времени тратится на синхронизацию, ожидание и взаимодействие с другими вычислительными элементами. Эффективность (efficiency) отражает долю времени выполнения алгоритма, в течение которой вычислительные элементы были реально задействованы для решения задачи

$$E = \frac{S}{p}. \quad (3)$$

В идеальном случае $S = p$ и $E = 1$. На практике $S < p$, а $0 < E < 1$. Попытки повышения качества параллельных вычислений по одному из показателей могут привести к ухудшению другого показателя. Так, например, увеличение ускорения может быть обеспечено за счет использования большего количества вычислительных элементов, что приводит на практике к падению эффективности. Увеличение же эффективности достигается порой за счет уменьшения количества вычислительных элементов. Поэтому на практике выбор параллельного алгоритма предполагает выбор некоторого компромиссного варианта с учетом желаемых показателей ускорения и эффективности.

Пример 4. Эффективность сложения n чисел с использованием n вычислительных элементов.

Вспомним пример 1, в котором мы вычислили ускорение. Эффективность алгоритма равна

$$E = \frac{\Theta\left(\frac{n}{\log n}\right)}{n} = \Theta\left(\frac{1}{\log n}\right).$$

2.5 Стоимость вычислений

Определим стоимость параллельных вычислений для решения данной задачи как произведение времени выполнения параллельного решения и количества вычислительных элементов использованных для этого:

$$C_p = T_p \cdot p. \quad (4)$$

Таким образом, эффективность может быть выражена отношением времени выполнения луч-

шего последовательного алгоритма к стоимости этих вычислений:

$$E = \frac{T_s}{C_p}. \quad (5)$$

Стоимость решения на одном вычислительном элементе равна времени выполнения лучшего последовательного алгоритма:

$$C_s = T_s. \quad (6)$$

Назовем параллельный алгоритм стоимостно-оптимальным (cost-optimal), если его стоимость пропорциональна времени выполнения наилучшего последовательного алгоритма:

$$C_p = k \cdot T_s. \quad (7)$$

Собирая вместе выражения (5), (6) и (7), получаем, что эффективность стоимостно-оптимального параллельного решения есть $\Theta(1)$

$$E = \frac{T_s}{C_p}, \text{ где } C_s = T_s \text{ и } C_p = k \cdot T_s,$$

преобразуем $E = \frac{1}{k}$, получаем $E = \Theta(1)$. (8)

Пример 5. Стоимость сложения n чисел с использованием n вычислительных элементов.

Рассмотрим алгоритм сложения n чисел из примера 1. $C_p = \Theta(n \cdot \log n)$ и $C_s = \Theta(n)$, по-

этому $E = \Theta\left(\frac{1}{\log n}\right)$, т. е. алгоритм не является

стоимостно-оптимальным.

Стоимость – очень важная практическая оценка, которая часто представляется в асимптотическом виде [4].

Пример 6. Производительность стоимостно-неоптимальных алгоритмов.

Пусть алгоритм сортировки n чисел, который использует n вычислительных элементов, выполняет сортировку за время $T_p = (\log n)^2$. Последовательный вариант сортировки имеет время выполнения $T_s = n \cdot \log n$. Ускорение этого ал-

горитма равно $S = \frac{n}{\log n}$, а эффективность

$E = \frac{1}{n}$, следовательно, этот алгоритм не является

стоимостно-оптимальным.

На практике количество вычислительных элементов много меньше числа сортируемых чисел

$p \ll n$, поэтому $T_p = \frac{n}{p} \cdot (\log n)^2$. Получаем

$$S = \frac{p}{\log n}, \text{ а } E = \frac{1}{\log n} - \text{ алгоритм так и остался}$$

стоимостно-неоптимальным. Возьмем $n = 1024$ и $p = 32$, получим $S = 3,2$ и $E = 0,1$. С ростом числа сортируемых элементов мы получаем убывающее значение ускорения и эффективности; так, уже при $n = 10^6$ ускорение и эффективность равны $S = 1,6$ и $E = 0,05$.

3. Влияние гранулярности вычислений на производительность параллельного алгоритма

Алгоритм из примера 1 есть стоимостно-неоптимальный (пример 5), этот алгоритм использует для своих вычислений избыточное количество вычислительных элементов. На практике мы имеем меньшее количество доступных нам вычислительных элементов для решения задачи, поэтому на каждый вычислительный элемент в отдельности приходится большее количество вычислений, т. е. больший объем входных данных, а это есть ни что иное, как увеличение гранулярности по данным. Использование меньшего количества вычислительных элементов, чем максимально доступных нам для решения задачи, называется уменьшением масштабируемости (scaling down) в терминах количества вычислительных элементов. Самый простой способ применить уменьшение масштабируемости на параллельный алгоритм – это спроектировать параллельные вычисления таким образом, чтобы на каждый вычислительный элемент приходилось по одному обрабатываемому значению входных данных. Затем использовать для вычислений меньшее количество вычислительных элементов, для того чтобы смоделировать больший объем входных данных.

Так как $p < n$, то объем вычислений для каждого вычислительного элемента вырос в n / p раз, или, другими словами, один вычислительный элемент выполняет вычисления за n / p вычислительных элементов (виртуальные вычислительные элементы). Если несколько таких виртуальных вычислительных элементов соотносятся с одним физическим вычислительным элементом, то суммарное время обмена данными между ними не увеличится более чем в n / p раз. Время T_p увеличится не более чем в n / p раз, а значение C_p не изменится. Поэтому если параллельный алгоритм стоимостно-оптимальный на n вычислительных элементах, то при условии моделирования

этих n вычислительных элементов на p ($p < n$) вычислительных элементах, алгоритм также будет стоимостно-оптимальным.

Слабая сторона такого подхода (увеличение гранулярности вычислений параллельных алгоритмов за счет использования уменьшения масштабируемости) заключается в том, что увеличение гранулярности для стоимостно-неоптимального алгоритма не гарантирует получение стоимостно-оптимального алгоритма.

Пример 7. Сложение n чисел на p вычислительных элементах ($p < n$).

Модифицируем алгоритм из примера 1. Ничто не мешает нам считать n и p степенями двойки, $n = 16$ и $p = 4$ (рис. 4). i -й виртуальный вычислительный элемент моделируется j -м физическим вычислительным элементом, где $j = i \bmod p$. Суммируемые значения распределяются по тому же правилу среди физических вычислительных элементов. $\log n$ шагов алгоритма

из примера 1 моделируются $\frac{n}{p} \cdot \log p$ шагами

модифицированного алгоритма на p физических вычислительных элементах. Таким образом, время вычислений занимает порядка

$$T_p = \Theta\left(\frac{n}{p} \cdot \log p\right), \text{ а } C_p = \Theta(n \cdot \log p), \text{ что пока}$$

зывает стоимостную неоптимальность модифи-

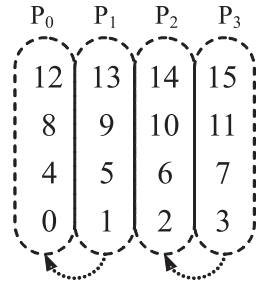
$$\text{цированного алгоритма } (E = \Theta\left(\frac{1}{\log p}\right)).$$

Пример 1 показал, что сложение n чисел на n вычислительных элементах может быть выполнено за $T_p = \Theta(\log n)$, модифицированный алгоритм из примера 7, моделируя n вычислительных элементов на p физически доступных элементах,

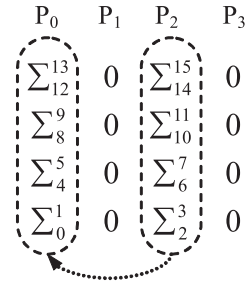
выполнил сложение за $T_p = \Theta\left(\frac{n}{p} \cdot \log p\right)$. Причи-

на тому – отсутствие коммуникации на последнем шаге модифицированного алгоритма (шаг 3), так как физически это был один вычислительный элемент. Этого оказалось недостаточно для того, чтобы сделать алгоритм стоимостно-оптимальным. Пример 8 демонстрирует, как мы можем получить стоимостно-оптимальный алгоритм за счет более продуманного распределения обрабатываемых элементов по физически доступным вычислительным элементам.

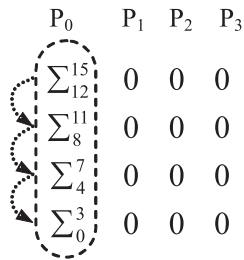
Пример 8. Стоимостно-оптимальный алгоритм сложения n чисел на p вычислительных элементах ($p < n$).



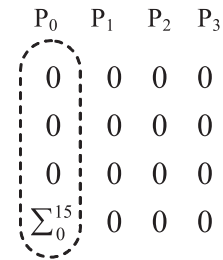
(1) Первый шаг алгоритма



(2) Второй шаг алгоритма

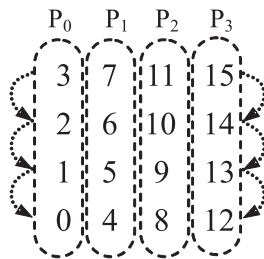


(3) Третий шаг алгоритма
(отсутствует межпроцессное взаимодействие)

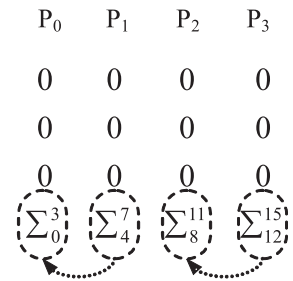


(4) Сумма на нулевом
вычислительном элементе

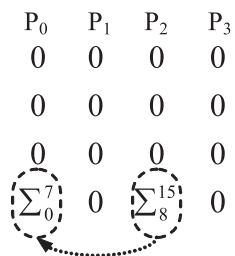
Рис. 4. $p = 4$ вычислительных элемента моделируют вычисления на $n = 16$ вычислительных элементах



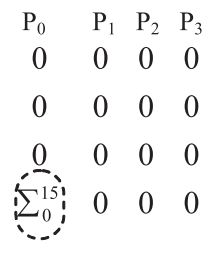
(1) Первый шаг алгоритма
(отсутствует межпроцессное взаимодействие)



(2) Второй шаг алгоритма



(3) Третий шаг алгоритма



(4) Сумма на нулевом
вычислительном элементе

Рис. 5. Стойменно-оптимальный алгоритм сложения $n = 16$ чисел с использованием $p = 4$ вычислительных элементов

Возьмем все те же $n = 16$ и $p = 4$. Распределим $\frac{n}{p}$ чисел локально для каждого из p вычислительных элементов и выполним сложение локально, всё это займет время пропорциональное $\Theta\left(\frac{n}{p}\right)$. Теперь сложим p частичных сумм на p вычислительных элементах – это занимает время пропорциональное $\Theta(\log p)$ (рис. 5). Таким образом, получаем алгоритм, время выполнения которого $T_p = \Theta\left(\frac{n}{p} + \log p\right)$. Его стоимость $C_p = \Theta(n + p \cdot \log p)$, так как $n = \Omega(p \cdot \log p)$, то стоимость есть $C_p = \Theta(n)$, а это есть ни что иное, как время выполнения последовательного алгоритма $T_s = \Theta(n)$. Получили стоимостно-оптимальный алгоритм.

Как мы уже замечали ранее, не все стоимостно-неоптимальные алгоритмы могут быть преобразованы к стоимостно-оптимальным за счет уменьшения масштабируемости.

4. Масштабируемость параллельных алгоритмов

Обычно параллельные программы создаются и отлаживаются на меньшей размерности задачи и на меньшем количестве вычислительных элементов, а в реальности же такие программы бывают рассчитаны на решение больших по объемам задач и с использованием большого количества вычислительных элементов. Если разработка программного кода может быть упрощена с помощью применения уменьшения масштабирования в терминах вычислительных элементов и размерности задачи, то оценку производительности и корректности написанного алгоритма уже не произвести с применением этого метода. А предсказание будущей производительности алгоритма на основании ограниченного набора наблюдаемых данных достаточно сложная задача, поэтому для этого используют аналитический аппарат.

Эффективность параллельных алгоритмов равна

$$E = \frac{S}{p} = \frac{T_s}{p \cdot T_p} . \quad (8)$$

Используя выражения для накладных расходов (1), перепишем выражение (8) следующим образом:

$$E = \frac{1}{1 + \frac{T_o}{T_s}} . \quad (9)$$

Величина T_o есть неубывающая функция от p (в большинстве случаев – возрастающая). Каждая параллельная программа содержит некоторое количество последовательного кода, пусть эта величина t_s . Получаем

$$\begin{aligned} T_o(p) &= p \cdot T_p - T_s = p \cdot (T_{p'} + t_s) - T_s = \\ &= p \cdot T_{p'} + p \cdot t_s - T_s = \\ &= p \cdot T_{p'} + p \cdot t_s - (T_{s'} + t_s) = \\ &= p \cdot T_{p'} - T_{s'} + (p - 1) \cdot t_s, \end{aligned}$$

где величина $(p - 1) \cdot t_s$ обеспечивает, по крайней мере, линейный рост функции. Однако в результате межпроцессного взаимодействия, синхронизации и избыточных вычислений рост функции может становиться суперлинейным. Изучая выражение (9), можно открыть несколько интересных фактов о масштабируемости параллельных алгоритмов. Если для данной размерности задачи (фиксируем значение T_s) увеличивать количество вычислительных элементов p , то накладные расходы T_o будут возрастать, а эффективность параллельного алгоритма будет снижаться. Снижение эффективности с ростом числа вычислительных элементов для данной размерности задачи – обычное явление для всех параллельных алгоритмов.

Пример 9. Ускорение и эффективность как функции от количества вычислительных элементов.

Рассмотрим все ту же проблему сложения n чисел с использованием p вычислительных элементов. Возьмем для анализа алгоритм из примера 8. Для наглядности будем оперировать неасимптотическими выражениями и положим, что время на коммуникации совпадает с временем сложения элементов и занимает одну единицу времени, тогда получим

$$T_p = \frac{n}{p} + 2 \cdot \log p , \quad S = \frac{n}{p} + 2 \cdot \log p ,$$

$$E = \frac{1}{1 + \frac{2 \cdot p \cdot \log p}{n}} .$$

Используем данные выражения для того, чтобы посчитать значения E и S для различных пар n и p (рис. 6 и 7).

Очевидно следующее: с увеличением размерности задачи при постоянном количестве вычислительных элементов эффективность и ускорение возрастают. При фиксированной же размерности задачи, но с ростом числа вычислительных элементов эффективность и ускорение уменьшаются.

Для некоторых параллельных алгоритмов можно удерживать эффективность на одном уровне за счет совместного изменения количества вычислительных элементов и размерности задачи. Для примера 9 очевидно, что эффективность 0,8 достижима в нескольких случаях: $(n = 64; p = 4)$, $(n = 192; p = 16)$, $(n = 512; p = 32)$. Способность сохранять эффективность на одном и том же уровне при одновременном увеличении количества вычислительных элементов и размерности задачи присутствует у многих параллельных алгоритмов. Такие параллельные алгоритмы называются масштабируемыми (scalable). Масштабируемость (scalability) есть величина, которая показывает возможность параллельного алгоритма демонстрировать большее ускорение по от-

ношению к возрастающему количеству вычислительных элементов [5]. Также эта величина показывает, насколько эффективно параллельный алгоритм может использовать вновь добавленные вычислительные элементы.

Масштабируемость и стоимостная оптимальность связаны между собой. Масштабируемый параллельный алгоритм всегда может быть сделан стоимостно-оптимальным, если количество вычислительных элементов и размерность задачи будут подбираться подходящим образом. Так, в примере 8 мы видели, что алгоритм будет стоимостно-оптимальным при условии $n = \Omega(p \cdot \log p)$, а пример 9 показал, что параллельный алгоритм будет к тому же и масштабируемым, если n увеличивается по отношению к p как $p \cdot \log p$.

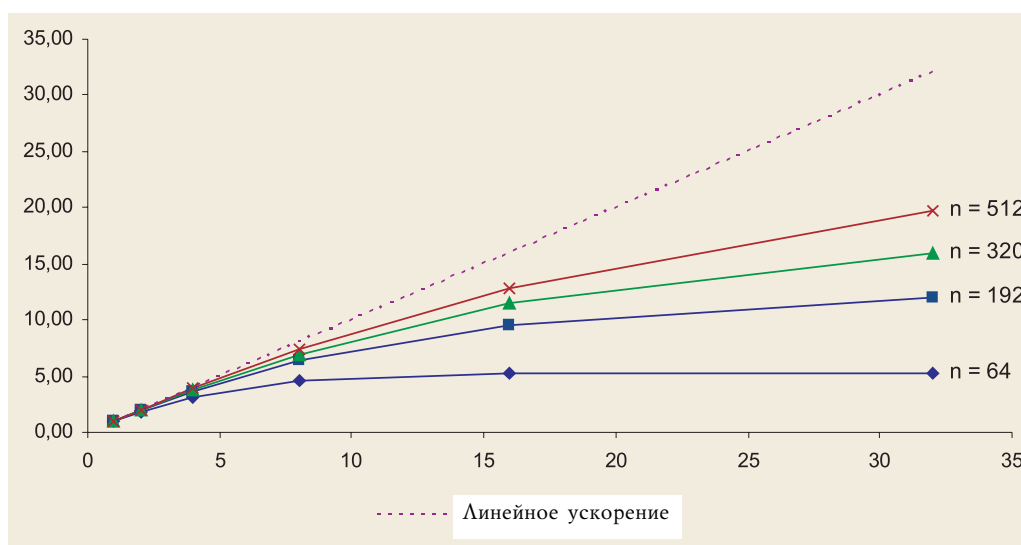


Рис. 6. Графики ускорения для различных пар n и p

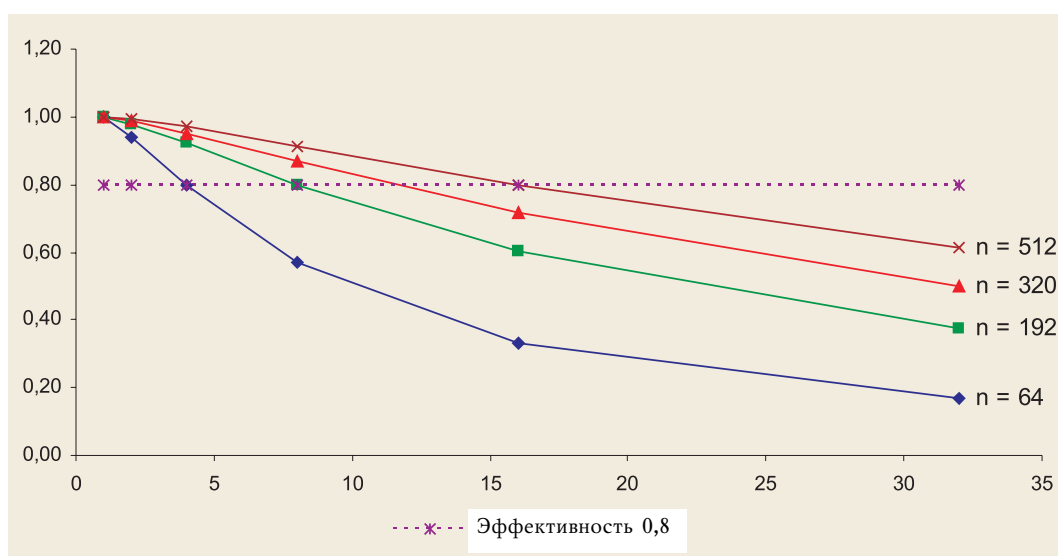


Рис. 7. Графики эффективности для различных пар n и p

5. Изоэффективность как мера масштабируемости

Зафиксируем два наблюдения, которые мы могли видеть в предыдущем разделе:

- для данной размерности задачи с увеличением количества вычислительных элементов эффективность параллельного алгоритма падает;
- в большинстве случаев эффективность параллельного алгоритма возрастает, если увеличивается размерность задачи, при этом количество вычислительных элементов остается постоянным.

Согласно этим двум наблюдениям мы можем определить масштабируемым алгоритмом такой алгоритм, эффективность которого остается постоянной при определенном соотношении в увеличении количества вычислительных элементов и размерности задачи. Полезно знать, в каком соотношении должны увеличиваться количество вычислительных элементов и размерность задачи, чтобы сохранять эффективность алгоритма постоянной. Такое соотношение определяет степень параллельности алгоритма. Перед тем как мы определим такое соотношение, необходимо определить понятие размерности задачи.

Когда мы анализируем параллельные алгоритмы, то очень часто используем понятие размерности задачи без точного его определения. Самый простой способ определения размерности — это размер входных данных для решения конкретной задачи. Недостаток такого определения заключается в том, что разные алгоритмы по-разному ведут себя на одном и том же количестве входных данных. Непротиворечивое определение размерности задачи не должно зависеть от конкретной задачи, то есть увеличение вдвое размерности задачи должно означать такое же увеличение количества вычислений. Так, для алгоритма перемножения матриц $n \times n$ размерность задачи будет равна $1(n^3)$. Для вполне конкретной задачи её размерность определим количеством основных вычислительных шагов в наилучшем последовательном алгоритме на одном вычислительном элементе. Так как для определения размерности задачи мы использовали сложность последовательного алгоритма, то размерность задачи есть функция W от размера входных данных. Далее будем полагать, что выполнение одного простого или основного вычислительного шага требует одной единицы времени. Имея в виду такое предположение, считаем, что $W = T_s$.

Выразим T_p , используя W , T_o и p . Используем выражение (1)

$$T_p = \frac{T_s + T_o}{p} = \frac{W + T_o(W, p)}{p}. \quad (10)$$

Выражение для ускорения переписывается в следующем виде:

$$S = \frac{W}{T_p} = \frac{W \cdot p}{W + T_o(W, p)}. \quad (11)$$

Перепишем выражение для эффективности:

$$E = \frac{S}{p} = \frac{W}{W + T_o(W, p)} = \frac{1}{1 + \frac{T_o(W, p)}{W}}. \quad (12)$$

Для различных параллельных алгоритмов размерность задачи W должна увеличиваться по отношению к количеству вычислительных элементов p с разной скоростью, чтобы сохранять эффективность постоянной. Например, W должна расти как экспоненциальная функция от p , чтобы сохранять эффективность на постоянном уровне. Масштабируемость такого параллельного алгоритма очень невелика. Причина кроется в том, что очень трудно получить хорошее ускорение на большом количестве вычислительных элементов. С другой стороны, если W должна линейно увеличиваться по отношению к увеличению количества вычислительных элементов p , то параллельный алгоритм обладает высоким уровнем масштабируемости, так как мы легко можем получить ускорение пропорциональное увеличиваемому количеству вычислительных элементов.

Перепишем выражение (12)

$$E = \frac{1}{1 + \frac{T_o(W, p)}{W}}, \quad W = \frac{E}{1 - E} \cdot T_o(W, p). \quad (13)$$

Предположим, что $K = \frac{E}{1 - E}$ и это константа, которая изменяется в зависимости от того, с какой эффективностью мы имеем дело.

$$W = K \cdot T_o(W, p). \quad (14)$$

Эта функция задает степень роста W по отношению к росту p с сохранением постоянной эффективности. Эта функция и называется функцией изоэффективности параллельного алгоритма. Функция изоэффективности показывает, насколько легко параллельный алгоритм сохраняет эффективность постоянной и достигает ускорения, пропорционального количеству вычислительных элементов. Чем меньше значение изоэффективности, тем меньше изменение в размерности задачи необходимо для эффективного использования добавленных вычислительных элементов, что, в свою очередь, означает хорошую

масштабируемость параллельного алгоритма. Для немасштабируемых параллельных алгоритмов не существует функции изоэффективности, потому что в таких параллельных алгоритмах нельзя обеспечить постоянную эффективность с ростом числа вычислительных элементов (не важно, как изменяется размерность задачи).

Пример 10. Функция изоэффективности сложения n элементов.

Вспомним пример 9, где

$$T_o = 2 \cdot p \cdot \log p.$$

Используем выражение для функции изоэффективности (14). Получаем:

$$W = K \cdot 2 \cdot p \cdot \log p.$$

Это означает, что при увеличении числа вычислительных элементов с p на p' , размерность задачи должна быть увеличена на коэффициент

$\frac{p' \cdot \log p'}{p \cdot \log p}$, чтобы получить ту же самую эффективность, что и на p вычислительных элементах.

После проведения анализа изоэффективности параллельного алгоритма мы можем испытать алгоритм на нескольких вычислительных элементах, а затем предсказать его производительность на большем количестве вычислительных элементов.

Подведем итог. Характеристики, приведенные в данной статье, позволят оценить способы решения конкретной задачи до момента их реализации, сделать прогноз о производительности решения при условии изменения количества вычислительных элементов или размерности задачи. Используя всё те же характеристики для анализа конкретного способа решения задачи, можно добиться эффективной утилизации вычислительных элементов.

Примечания

1. Grama, A. Introduction to parallel computing [Text] / A. Grama, A. Gupta, G. Karypis, V. Kumar. The Benjamin/Cummings Publishing Company, Inc., 1994.

2. Ibid.; Гергель, В. П. Теория и практика параллельных вычислений [Текст] / В. П. Гергель. М.: БИНОМ. Лаборатория знаний, 2007.

3. Гергель, В. П. Указ. соч.

4. Grama, A. Op. cit.

5. Ibid.

М. М. Шихов, Р. С. Петров

ПАРАЛЛЕЛЬНЫЙ МЕТОД ДЕДУКТИВНОГО ЛОГИЧЕСКОГО ВЫВОДА В ИСЧИСЛЕНИИ ПРЕДИКАТОВ

Статья содержит описание метода двунаправленного дедуктивного вывода в исчислении предикатов первого порядка. В статье приводятся постановка задачи логического вывода, краткое описание используемых в логическом выводе вспомогательных процедур, описание метода вывода и его оценка.

Введение

Моделирование рассуждений средствами логики является одним из важнейших направлений науки, изучающей методы и средства искусственного интеллекта [1]. Формальное описание предметной области в логике предикатов дает возможность обобщать сложные зависимости данных достаточно естественным для человека образом. Обычно выделяют следующие основные виды логического вывода: дедуктивный, индуктивный и абдуктивный. В данной работе приводится описание подхода к реализации параллельного метода двунаправленного дедуктивного вывода в исчислении предикатов первого порядка. Суть задачи дедуктивного вывода – доказать истинность или ложность некоторого высказывания на основании множества посылок.

Двунаправленный логический вывод

В реализации метода двунаправленного логического вывода используется представление базы знаний в виде дизъюнктов. За основу взят метод логического вывода делением дизъюнктов: см. [2]. Подробно используемые в данной работе алгоритмы частичного и полного деления дизъюнктов описаны в работе [2]. Далее приводится лишь краткое их описание.

Алгоритм частичного деления дизъюнктов предназначен для того, чтобы получить множество остатков. Делимое – это дизъюнкт (дизъюнкция литералов) базы знаний, на основании истинности которого можно делать выводы об истинности делителя. Истинность делителя необходимо доказать. Остаток – это дизъюнкт, в случае доказательства ложности которого будет доказана истинность делителя. Выделение общих частей есть результат унификации, то есть применения к обоим дизъюнктам делимого и делителя некоторой подстановки. Результатами час-

ШИХОВ Михаил Михайлович – Вятский государственный университет

ПЕТРОВ Родион Сергеевич – Федеральное государственное унитарное предприятие «Научно-исследовательский институт средств вычислительной техники» (НИИ СВТ)

© Шихов М. М., Петров Р. С., 2008

тичного деления дизъюнктов может быть множество решений и/или множество остатков. Также эти множества могут остаться пустыми, что будет означать, что делитель из делимого вывести невозможно. Возможных вариантов унификации литералов дизъюнктов в общем случае множество, поэтому в результате перебора вариантов будут получаться множества решений и остатков.

Введем следующие обозначения. Дизъюнкт будем обозначать как $D=(L_0 \mid L_1 \mid \dots \mid L_{N-1})$, где N – количество литералов в дизъюнкте. L_i – i -й литерал дизъюнкта ($0 \leq i < N$). Литерал – это предикат или отрицание предиката, например $P(x, y)$ или $\neg P(x, y)$, где символ « \neg » – символ отрицания. Подстановки будем обозначать символом λ , например, $\lambda = \{x/y, c/z, f(c)/k, f(?)/_g(?)\}$, где запись x/y обозначает замену переменной y на переменную x ; x, y, z, k – символы переменных; f – функтор; « c » – обозначение константы; $_g$ – символ переменной для унификации функторов. Пустую подстановку будем обозначать $\{0\}$, невозможную символом $\{fail\}$. База знаний представляет собой набор дизъюнктов $D_0 \& D_1, D_2 \& D_3 \& \dots \& D_{N-1}$, где символ « $\&$ » связывает одноименные переменные в литералах дизъюнктах, а символ « $,$ » начинает новую область видимости переменных. Например, переменная с именем x в

дизъюнктах D_0, D_1 – общая, а в дизъюнктах D_1, D_3 – нет. Аналогично выглядит заключение (цель).

Полное деление дизъюнктов – это результат применения процедуры частичного деления дизъюнктов ко всей базе знаний в процессе доказательства истинности какого-либо делителя. В результате полного деления дизъюнктов, возможно, будут получены непустые множества решений и остатков. В зависимости от этого будет выполняться очередной шаг логического вывода, будет сделано заключение о невозможности дальнейшего вывода на данном шаге или будут получены решения текущего шага, которые нужно согласовать с решениями предыдущих шагов.

Для сокращения объемов приводимых формул опишем основные объекты логического вывода и их взаимодействие:

- узел дерева логического вывода;
- агент логического вывода;
- менеджер агентов логического вывода.

Узел дерева логического вывода. Данный объект – элемент, составляющий дерево логического вывода. Структура дерева и узла дерева логического вывода приведена на рис. 1.

Подготовительный шаг логического вывода – создание начальной структуры дерева логического вывода для заключения $D_0 \& D_1 \& \dots \& D_{N-1}$.

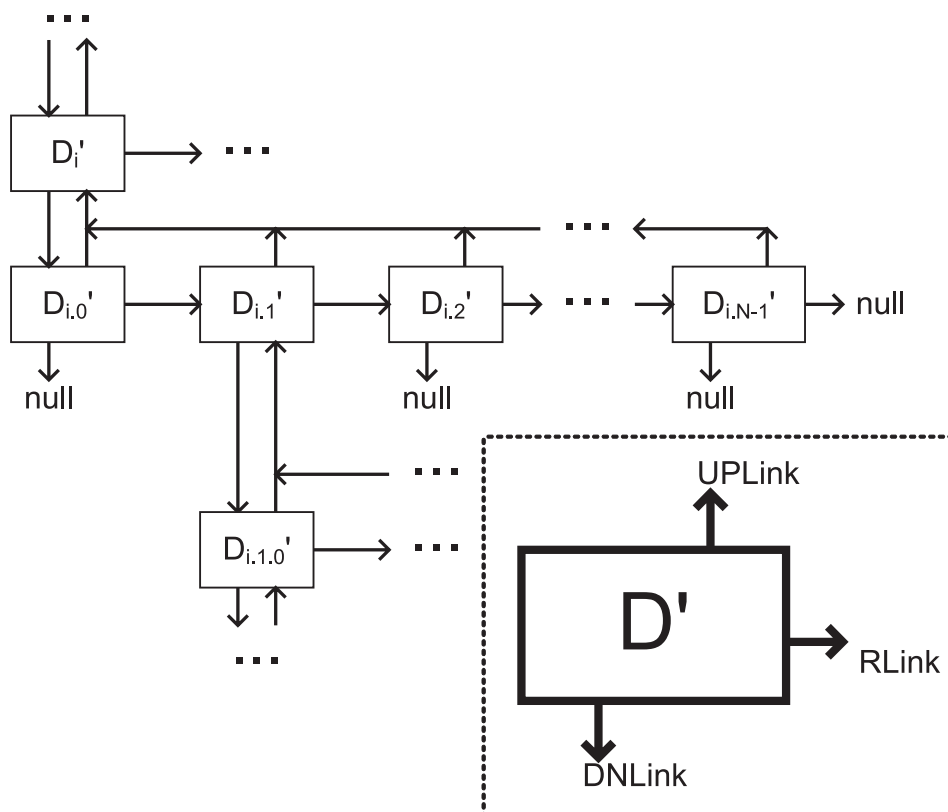


Рис. 1. Дерево и узел дерева логического вывода

Здесь приняты следующие обозначения:

$D_i' = (< L_{i,0}, \{ \} > | < L_{i,1}, \{ \} > | \dots | < L_{i,M_i-1}, \{ \} >)$ есть дизъюнкт-делитель, в котором $L_{i,j} \in D_i'$, а $D_i' = (L_{i,1} | L_{i,2} | \dots | L_{i,M_i})$ M_i – количество литералов $L_{i,j}$ в i -м литерале $0 \leq j < M_i$, N – количество дизъюнктов $0 \leq i < N$. Угловые скобки формируют отдельную структуру – пара литерал – подстановка (изначально пустая). Класс узла в нотации ОМТ (Object Modeling Technique) представлен на рис. 3.

Node
i; D; uplink; dnlink; rlink; F; R; O; M; RChild; Goals; NPSubNodes;
Node() ~Node() Divide() GetGoals() GetRFromGoals() MakeSubNodes() Reconciliation()

Рис. 3. ОМТ-диаграмма для описания узла дерева логического вывода

На рис. 3: i – индекс узла (индексная функция, см. определение в [2]); D – цель данного узла (выводимый данным узлом дизъюнкт, вместе с i соответствующий $D_i' = (< L_{i,0}, \lambda_{i,0} > | < L_{i,1}, \lambda_{i,1} > | \dots | < L_{i,M_i-1}, \lambda_{i,M_i-1} >)$); **uplink** – ссылка на родительский узел; **dnlink** – ссылка на первый дочерний узел; **rlink** – ссылка на соседний узел («братский» узел, т. е. порожденный тем же родительским узлом); **F** – флаг, указывающий на статус узла («ожидание доказательства» (false) / «в процессе доказательства» (true)); **R** – множе-

ство полученных решений; **O** – множество полученных остатков; **M** – ссылка на множество посылок (базу знаний $D_0 \& D_1, D_2 \& D_3 \& \dots$); **RChild** – множество решений, полученных в результате согласования решений из дочерних узлов; **Goals** – множество новых выводимых подцелей, полученное в результате преобразования множества остатков; **NPSubNodes** – количество недоказанных подузлов.

Методы узла рассмотрим подробнее.

Node(l_i, l_D, l_uplink, l_M) – конструктор узла. Вызывается в момент его создания. Инициализирует данные $i := l_i, D := l_D, uplink=l_uplink, M := l_M$. Статус узла $F := false$. Остальные ссылки устанавливаются в **null**. Все множества инициализируются пустым множеством.

~Node() – деструктор узла, вызывается при уничтожении узла, уничтожает сформированные множества узла, освобождая память.

Divide() – процедура полного деления аргумента **M** на выводимый **D**. В процессе деления порождаются копии дизъюнктов посылок, все переменные которых индексируются аргументом i (таким образом узел создает уникальное множество переменных – область видимости узла). Результат выполнения данной процедуры – множество остатков $O (O_{i,0} \& O_{i,1} \& \dots \& O_{i,N_i-1} = 0$, где N_i – количество остатков i узла, $O_{i,j} = < D_{i,j}, \lambda_{i,j} >$ – остаток, где $0 \leq j < N_i$), а также множество полученных решений, заносимое во множество **R**. Также возможно, что множества **O** и **R** останутся пустыми.

GetGoals() – данная процедура выполняет преобразование множества остатков во множество выводимых дизъюнктов. Выполняется после **Divide()**. Суть преобразований в следующем: $O_{i,0} \& O_{i,1} \& \dots \& O_{i,N_i-1} = 0$ преобразуется в $\overline{O_{i,0}} | \overline{O_{i,1}} | \dots | \overline{O_{i,N_i-1}} = 1$, а затем в $O'_{i,0} \& O'_{i,1} \& \dots \& O'_{i,M_i-1} = 0$.

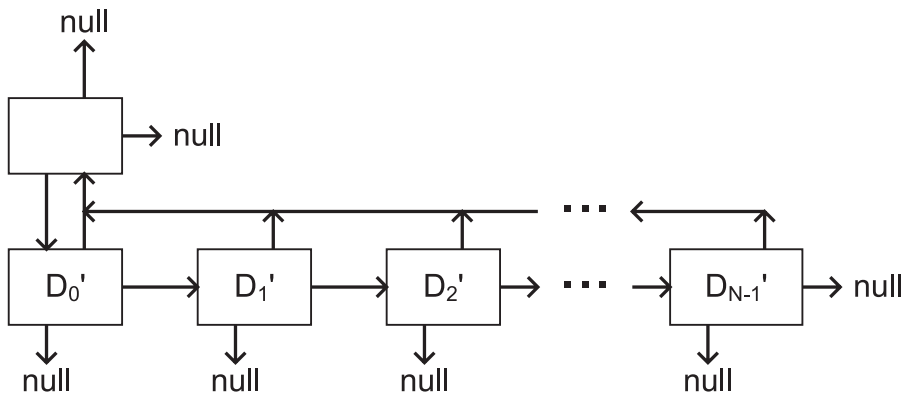


Рис. 2. Подготовительный этап логического вывода

$$O'_{i,l} = (\langle L_{i,0,k_{i,0}}, \lambda_{i,0} \rangle \langle L_{i,1,k_{i,1}}, \lambda_{i,1} \rangle \dots \langle L_{i,N_i,k_{i,N_i}}, \lambda_{i,N_i} \rangle) \quad (1)$$

где L_{i,j,k_j} – k_j -й литерал в составе дизъюнкта $D_{i,j}$ ($0 \leq k_j < \text{Power}(D_{i,j})$). Обозначение Power – количество литералов в дизъюнкте. Также справедливо ($0 \leq l < M_i$), где M_i определяется в соответствии с формулой 2. Каждому l соответствует число в системе счисления с переменными весами разрядов $l = (k_{i,0}, k_{i,1}, \dots, k_{i,N_i})$, каждому $k_{i,j}$ соответствует вес разряда $\text{Power}(D_{i,j})$ и $k_{i,j}$ изменяется в пределах $0 \leq k_{i,j} < \text{Power}(D_{i,j})$. Для каждого l , найдя разложение в системе счисления $(k_{i,0}, k_{i,1}, \dots, k_{i,N_i})$ в соответствии с формулой 3, получим все варианты новых целей $O'_{i,l}$.

$$M_i = \prod_{j=0}^{N_i-1} \text{Power}(D_{i,j}), \quad (2)$$

$$k_{i,0} + \sum_{j=1}^{N_i-1} k_{i,j} \prod_{m=0}^{j-1} \text{Power}(D_{i,m}) = l. \quad (3)$$

GetRFromGoals() – получить решения из подцелей. Возможно два варианта дальнейшего доказательства. Либо доказать истинность формулы $O'_{i,0} \& O'_{i,1} \& \dots \& O'_{i,M_i-1} = 1$, либо доказать, что каждый $O'_{i,l}$ истинен, выводя его из множества посылок. Данная процедура реализует первый вариант. В составе каждого дизъюнкта $O'_{i,l}$ находятся контрарные (взаимноинверсные) пары литералов L_{i,x,k_j} и L_{i,y,k_j} , один из которых (неважно, какой) инвертируется, и унифицируются. Решение для $O'_{i,l}$ будет представлять собой согласованное объединение полученной в результате унификации подстановки и подстановок $\lambda_{i,x}, \lambda_{i,y}$. Решения находятся для всех $O'_{i,l}$ в составе множества целей и согласуются. Если хотя бы для одного $O'_{i,l}$ решение не найдено или согласование полученных решений прошло неудачно – решений узла нет, иначе множество решений заносится в **R**.

MakeSubNodes() – сформировать подузлы. Для каждого элемента множества **O** – $O'_{i,l}$ конструируется отдельный узел: **Node**($i.l, O'_{i,l}, \text{this}, \text{M}$). Настраиваются ссылки в соответствии с рис. 1. **this** – ссылка на порождающий узел (ссылка на активный объект). Для текущего узла **NPSubNodes** := M_i .

Reconciliation() – процедура согласования решений. Когда в узле получено множество решений (в результате успешного выполнения **GetRFromGoals()** или успешного доказательства в подузлах), решения данного узла нужно перенести в родительский узел, причем переносимое решение должно согласоваться с решениями, полученными родительским узлом от других его подузлов. Происходит запись в родительский узел, поэтому данная процедура должна быть атомарной для узла **uplink**. В начале согласо-

вания узел **uplink** блокируется для доступа всех остальных процессов (агентов) операцией **LockRChild(uplink)** менеджера агентов. Происходит согласование решений **R** и **uplink.RChild**. Узел **uplink** освобождается вызовом операции **UnLockRChild(uplink)** менеджера агентов. Блокировки выполняются агентом.

Агент логического вывода. Структура класса агента в нотации ОМТ приведена на рис. 4.

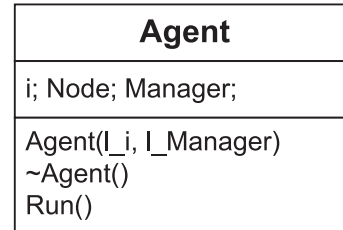


Рис. 4. ОМТ-диаграмма для описания агента логического вывода

Агент выполняется в отдельном вычислительном потоке, создаваемом в менеджере агентов. Параллельно может работать множество агентов. Количество одновременно работающих агентов определяется менеджером агентов.

На рис. 4: **i** – индекс узла дерева логического вывода, для доказательства которого был создан агент; **Node** – ссылка на текущий узел дерева логического вывода (у агента имеется возможность вызывать методы узла в своем вычислительном контексте и переходить от узла к узлу по ссылкам **Node.uplink**, **Node.dnlink**, **Node.rlink**); **Manager** – ссылка на объект – менеджер агентов.

Рассмотрим методы агента.

Agent(l_i, l_Node, l_Manager) – конструктор агента. Устанавливает **i** := **l_i**, **Manager** := **l_Manager**.

~Agent() – деструктор агента.

Run() – основной метод агента. Реализует методику логического вывода.

Алгоритм работы метода **Run()**

1) Если **Node.rlink** <> **null**, то попытаться создать агента для узла **Node.rlink**, обратившись к менеджеру **Manager.Create(Node.rlink)**. Перейти к шагу 2.

2) Последовательно выполнить методы узла: **Node.Divide()**, **Node.GetGoals()**, **Node.GetRFromGoals()**. Если не превышена глубина логического вывода, выполним **Node.MakeSubNodes()**. Если порождено множество подузлов (**Node.DnLink** <> **null**), то перейти по ссылке **Node** := **Node.dnlink** и перейти к шагу 1 (продолжается доказательство в направлении вниз). Иначе перейти к шагу 3.

3) Выполнить метод `Node.Reconciliation()` с блокировкой узла `Node.uplink`:

```

Manager.LockRChild(Node.uplink);
Node.Reconciliation();
Node.uplink.NPSubNodes :=
Node.uplink.NPSubNodes - 1;
Manager.UnlockRChild(Node.uplink);

```

Если $i = \text{Node.i}$, то агент выполнил свою работу по доказательству узла i и освобождает вычислительный ресурс, вызывая `M.Terminate(this)` (на этом алгоритм агента заканчивается, эквивалент перехода к шагу 6). Если в узле `Node.uplink` выполняется условие `Node.uplink.RChild = {fail}`, то дальнейшее доказательство подузлов выполнять не имеет смысла (получить согласованные решения уже невозможно), выполняем переход `Node := Node.uplink` и переходим к шагу 3 алгоритма. Иначе проверяется возможность попытки доказать узел по ссылке `Node.rlink`. Переходим к шагу 4.

4) Проверяется ссылка `rlink`. Если `rlink = null`, то на данный момент либо все подузлы узла `Node.uplink` доказаны, либо часть находится в процессе доказательства. Выполняется переход `Node := Node.uplink`. Агент должен перейти в состояние ожидания до тех пор, пока в узле не установится состояние в узле `Node`, такое, что либо `Node.NPSubNodes = 0`, либо `Node.uplink.RChild = {fail}`. Вызывается метод `Manager.Wait(this, Node)`. После выхода из ожидания объединяются множества `R` и `RChild` и происходит переход к шагу 3. Иначе, если `rlink <> null`, – к шагу 5.

5) Попытка захватить узел справа для доказательства. `Node := Node.rlink`. Выполняется операция `Manager.LockF(Node)`. Проверяется флаг `F`. Если `F = true`, то узел доказывается (или уже доказан) другим агентом, в этом случае выполняется `Manager.UnlockF(Node)` и выполняется переход к шагу 4. Иначе захватывается узел `F := true` и выполняется `Manager.UnlockF(Node)`. Переход к шагу 2.

6) Конец алгоритма `Run()`. Уничтожение агента логического вывода менеджером.

Итак, в соответствии с изложенным алгоритмом работы агента возможно два основных направления доказательства: вниз, порождая подузлы дерева логического вывода и получая решения, и вверх, согласуя полученные решения. В конечном итоге агент возвращается в ту вершину дерева, в которой был создан. Также исключена возможность входа в один и тот же узел двух агентов одновременно.

Менеджер агентов логического вывода. Структура класса агента в нотации ОМТ приведена на рис. 5.

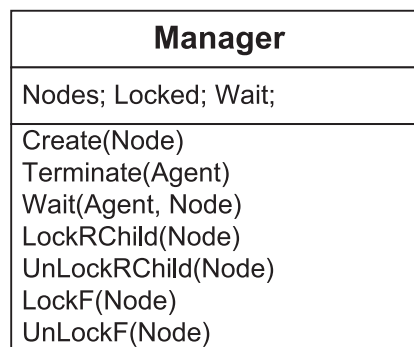


Рис. 5. ОМТ-диаграмма для описания менеджера агентов логического вывода

Функции менеджера агентов заключаются в рациональном использовании вычислительных ресурсов (диспетчеризации), создании агентов логического вывода, организации пассивного ожидания агентов, блокировке одновременного доступа к некоторым параметрам узла, уничтожении агентов. Реализация данного элемента может в значительной степени варьироваться, поэтому он описан более абстрактно, чем предыдущие объекты.

Nodes – список узлов дерева логического вывода, требующих доказательства. Алгоритмы создания и ведения списка в данной работе не рассматриваются. Количество создаваемых менеджером агентов порой ограничено, если для некоторого узла был запрос на создание агента для его доказательства, то данный узел фиксируется в множестве **Nodes**. В дальнейшем, когда некоторые агенты завершают свое существование, освободившиеся вычислительные ресурсы используются для создания агентов в данных узлах.

Locked – заблокированные узлы по тем или иным параметрам. Функции семейства `LockXXX(Node)` помещают узел в данный список. Агент, вызывающий функции семейства `LockXXX(Node)`, если узел заблокирован (находится в списке), попадает в пассивное ожидание до возникновения определенного события (тип события и агент фиксируются во множестве **Wait**).

Wait – множество, содержащее список соответствия агентов, находящихся в пассивном ожидании, событиям, в результате возникновения которых агент выводится из режима ожидания и продолжает работу. В ряде случаев ожидание может быть длительным (см. пункт 4 алгоритма `Agent::Run()`) и возможно перераспределение вычислительных ресурсов с целью ускорения логического вывода (например, агент ждет завершения доказательства подузлов в некотором узле. Его освободившийся вычислительный ресурс менеджер может направить на ускорение доказательства в один из подузлов в виде нового агента).

Методы объекта Manager:

Create(Node) – запрос на создание агента в узле Node. Если вычислительных ресурсов не будет хватать для создания агента, то данный узел будет в списке Nodes и при первой возможности менеджер создаст агента для доказательства данного узла (также должны выполняться проверки на то, что данный узел уже доказывается (доказан) одним из ранее созданных агентов). Данный метод создает агента, передавая следующие аргументы в его конструктор: **new Agent(Node.uplink.i, Node, this)**.

Terminate(Agent) – уничтожение агента (освобождение вычислительного ресурса).

Wait(Agent, Node) – установить в пассивное ожидание агента Agent до наступления события в узле Node такого, что либо **Node.NPSubNodes = 0**, либо **Node.uplink.RChild = {fail}**.

LockRChild(Node) – заблокировать узел Node (поместив в множество Nodes) по доступу к **Node.RChild** до вызова **UnLockRChild(Node)**.

UnLockRChild(Node) – разблокировать узел Node, исключив его из множества Nodes.

LockF(), **UnLockF()** – действие этих функций аналогично функциям **LockRChild(Node)** и **UnLockRChild(Node)** но в отношении доступа к **Node.F**.

Пример логического вывода

Задача поиска производных. Структура базы знаний приведена под рис. 6.

В результате было получено решение **{mul(exp(sin(«x»)), mul(cos(«x»), «1»))/r ...}**. В

случае достаточного количества ресурсов у менеджера агентов узлы с индексами 0.0.0 и 0.0.1 (а также 0.0.1.0 и 0.0.1.1) могли быть доказаны параллельно.

```

01> P(«x», «1») //x'=1
01> P1(neg(x), neg(x)) //(-x)'=-x
02> P1(sin(x), cos(x)) //sin(x)'=cos(x)
03> P1(exp(x), exp(x)) //exp(x)'=exp(x)
04> (P_f(s), mul(dfds, dsdx)) | //f(s(x))=f'(s)s'(x)
05> !P1_f(s), dfds |
06> !P(s, dsdx)
07>
08> ->
09> P(exp(sin(«x»)), r).
    
```

В целях упрощения восприятия в узлах не приводятся подстановки, делитель в узле также представим обычным дизъюнктом, а не в виде $O_{i,l}$. В узле на рис. 6 приводится индексная функция и делитель.

Заключение

В заключение дадим оценку быстродействия данного метода. Для дерева в N узлов в худшем случае (линейное дерево или ограниченность ресурсов менеджера 1 агент) потребуется $O(N)$ шагов в прямом направлении (деления, поиск решений и т. д. см. выше) и столько же шагов в обратном направлении (согласование решений). В случае неограниченного ресурса требуется $O(L)$ шагов логического вывода в прямом направлении и столько же в обратном, где L – количе-

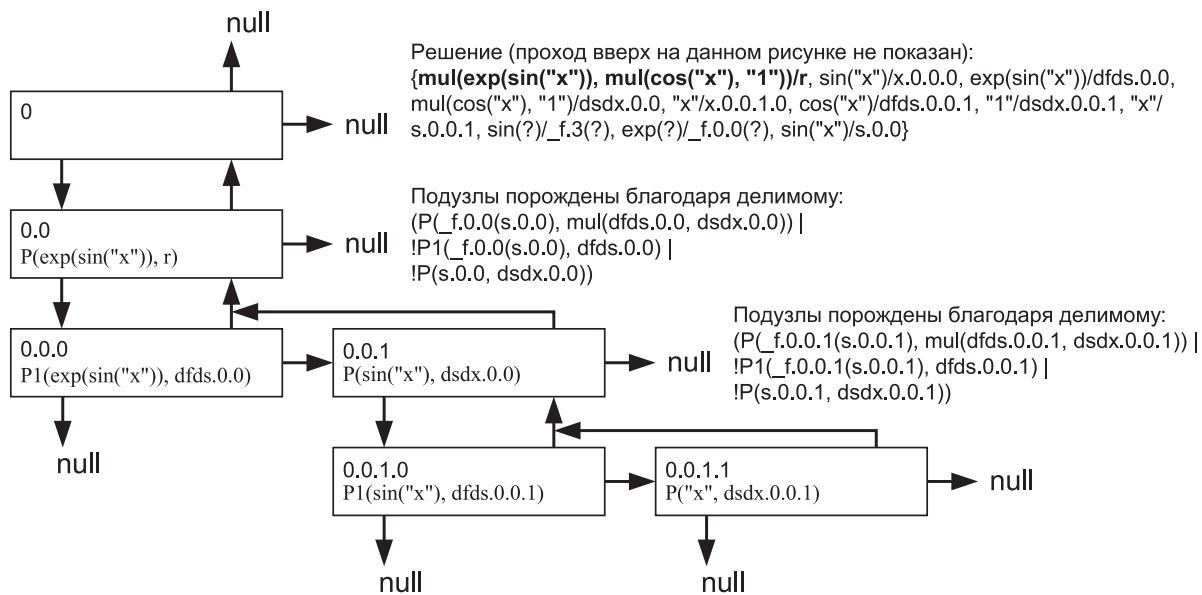


Рис. 6. Пример логического вывода

ство узлов в самой длинной ветви дерева логического вывода. Данные оценки достаточно грубы, так как не учитывают вероятности длительного пассивного ожидания, например в результате согласования решений.

При данном подходе значительную роль в процессе логического вывода играет менеджер агентов. В случае, если ресурс менеджера ограничен одним агентом, вывод, тем не менее, возможен за счет алгоритма работы агента, который будет осуществлять последовательный обход дерева.

Примечания

1. Вагин, В. Н. Достоверный и правдоподобный вывод в интеллектуальных системах [Текст] / В. Н. Вагин, Е. Ю. Головина, А. А. Загорянская, М. В. Фомина; под ред. В. Н. Вагина, Д. А. Поспелова. М.: Физматлит, 2004. 704 с.

2. Страбыкин, Д. А. Логический вывод в системах обработки знаний [Текст] / Д. А. Страбыкин; под ред. Д. В. Пузанкова; СПбГЭТУ. СПб., 1998. 164 с.

3. Шихов, М. М. Использование дополнительно типа переменных для унификации функторов в исчислении предикатов [Текст] / М. М. Шихов // Всероссийская научно-техническая конференция «Наука-производство технология-экология»: сб. м-лов: в 8 т. Киров: Изд-во ВятГУ, 2006. Т. 1. ФАВТ, ФПМТ. С. 227.

Е. В. Юзрина

МЕТОДИКА ИЗУЧЕНИЯ КОНФИГУРАЦИИ 1С:ТОРГОВЛЯ И СКЛАД НА ОСНОВЕ КОГНИТИВНО-ПРАГМАТИЧЕСКОГО ПОДХОДА

В статье рассмотрен новый подход к методике преподавания известных программных продуктов.

В приказе Министерства образования России о концепции модернизации российского образования на период до 2010 г. говорится, что «основная цель профессионального образования – подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности; удовлетворение потребностей личности в получении соответствующего образования».

ЮГРИНА Елена Вячеславовна – старший преподаватель кафедры прикладной информатики ВятГУ
© Югрина Е. В., 2008

На данный момент методика преподавания информационных технологий не в полной мере реализует основную цель профессионального образования – подготовку специалистов соответствующего уровня. Выполнение требований современного общества к выпускникам может обеспечить методика преподавания, реализующая когнитивно-прагматический подход, т. е. способствующая познанию, познавательной деятельности, в ее основе лежит непосредственный личный опыт, через призму которого оцениваются знания и отыскиваются подходы к решению проблем. С помощью этой методики будет достигнута осознанность изучения, связь теории с практикой, активация индивидуальных познавательных стилей и, как следствие, подготовка квалифицированного работника, с последовательным типом мышления, склонного к предварительному планированию всех аспектов своей деятельности, готового к сотрудничеству, обладающего высоким уровнем энтузиазма и умения при решении прикладных проблем, готового к постоянному профессиональному росту. Основной целью данной методики преподавания информационных технологий является привитие навыка к самостоятельному изучению новых программных продуктов, самостоятельному решению возникающих профессиональных проблем, развитие познавательных стилей при изучении информационных технологий.

Длительность жизненного цикла информационной технологии, используемой в экономической сфере деятельности общества, составляет от двух до пяти лет, поэтому нет смысла делать упор на обучении работе в конкретных программах, определенным способом действий при изучении информационных технологий. В настоящее время большинство педагогов по информационным технологиям обучают конкретным действиям в программах, навязывая свой опыт и приучая студентов изучать новое не думая, не развивая свои познавательные стили в полном объеме. Компьютерная программа изменилась, усовершенствовалась, появилась другая, и пользователю становится сложно, так как он привык к определенным командам, пунктам меню, к определенному способу действия. Согласно теории идентичных элементов когнитивного психолога Торндайка, тренировка в одной сфере деятельности будет перенесена в другую, только если эти деятельности имеют общие элементы типа «ситуация – реакция». Иными словами, студент, приученный в рамках традиционного обучения реагировать выполнением предложенного алгоритма действий, точно так же будет действовать и в своей профессиональной деятельности, не обладая, таким образом, свойством конкурентоспособности и профессиональной мобильности. Необходима

единая методика изучения экономических программных продуктов, ставящая целью развить познавательные стили студента до такого уровня, который позволит ему самостоятельно изучать любую информационную технологию по своей специальности, в частности экономическую информационную технологию. Этой методикой может быть методика, основанная на когнитивно-прагматическом подходе.

Когнитивно-прагматический подход – совокупность системы приемов и способов воздействия на обучаемого, которые способствуют познанию через личный опыт, через самостоятельные эксперименты, апробацию, через действия, учитывает и развивает индивидуальные познавательные стили, что позволяет формировать многогранный персональный познавательный стиль и такие знания, умения и навыки, которые применимы для решения задач на практике, в деле.

Когнитивность реализуется в когнитивно-прагматическом подходе в развитии индивидуальных познавательных стилей, способствующих формированию многогранного персонального познавательного стиля.

Прагматичность реализуется в когнитивно-прагматическом подходе, во-первых, в применяемых приемах и способах, способствующих познанию через самостоятельные действия, а во-вторых, и это главное, в получаемом результате: сформировавшихся знаний, умений и навыков, а также перешедшем на новый уровень развития персональном познавательном стиле, которые обучающийся будет применять в деле: в учебной, в будущей профессиональной деятельности, в сложившихся жизненных ситуациях.

О получении высокого результата при когнитивно-прагматическом подходе можно говорить только при условии применения данного подхо-

Таблица 1

Классификация уровней мышления Б. Блума

Уровень	Определение	Что делает учитель	Что делает учащийся	Какие ключевые термины используются для побуждения учащихся
Знание	Определение и отбор информации	Рассказывает, показывает, направляет	Воспринимает, запоминает, распознает	Перечислите, запомните, назовите
Понимание	Понимание предоставленной информации; формулирование проблемы собственными словами	Сравнивает, противопоставляет, демонстрирует	Объясняет, преобразовывает, демонстрирует	Обсудите, определите, расскажите
Применение	Использование понятий в новых ситуациях	Наблюдает, помогает, критикует	Решает проблемы, демонстрирует знания	Примените, вычислите, измените, выберите, классифицируйте, завершите, продемонстрируйте, обнаружьте, инсценируйте, задействуйте, исследуйте, проведите эксперимент, проиллюстрируйте, интерпретируйте, модифицируйте, оперируйте, потренируйте, соотнесите, спланируйте, покажите, сделайте набросок, решите, используйте
Анализ	Разбиение информации на связанные части	Направляет, исследует, информирует	Разделяет, обсуждает, раскрывает	Проанализируйте, оцените, сгруппируйте, вычислите, категоризируйте, классифицируйте, сравните, свяжите, противопоставьте, критикуйте, обсудите, дифференцируйте, различите, разделите, исследуйте, проведите эксперимент, объясните, выведите, упорядочьте, усомнитесь, соотнесите, выберите, разделите, проверьте
Синтез	Компиляция информации	Обобщает, оценивает, рассуждает	Обобщает, формулирует, планирует	Сгруппируйте, соберите, скомбинируйте, составьте, создайте, разработайте, сформулируйте, обобщите, объедините, придумайте, модифицируйте, организуйте, спланируйте, подготовьте, предложите, перегруппируйте, перепишите, установите, замените
Оценка	Оценивание на основе критериев	Уточняет, допускает, гармонизирует	Дискутирует, оценивает, выбирает	Докажите, выберите, сравните, сделайте вывод, убедите, решите, обоснуйте, объясните, измерьте, предскажите, проранжируйте, порекомендуйте, выделите, суммируйте, поддержите, проверьте, оцените

Классификация уровней мышления Андерсона и Красвола

Мыслительные процессы	Примеры
Запоминание – извлечение нужной (правильной) информации из памяти	
Узнавание	Ответы на вопросы типа «правильно/неправильно» или с множественным выбором. Выбор объекта среди ряда ему подобных
Перечисление	Перечисление фактов, событий, явлений
Понимание – представление собственного мнения об изученном материале	
Интерпретация	Пересказ какой-либо информации своими словами
Приведение примеров	Поиск и объяснение примеров, подтверждающих факты, события, явления
Классификация	Распределение информации по группам (классам)
Мыслительные процессы	Примеры
Суммирование	Придумывание заголовка к тексту. Перечисление ключевых позиций, относящихся к рассматриваемому явлению, процессу
Представление выводов	Анализ информации, представленной в каком-либо тексте, и представление выводов
Сравнение	Проведение сравнительного анализа явлений, процессов
Объяснение	Использование диаграмм, схем для представления информации
Применение – использование полученной информации	
Оформление результатов	Представление результатов эксперимента
Исполнение	Выполнение эксперимента
Анализ – разбиение целого на части и описание того, как части соотносятся с целым	
Различение	Составление списков наиболее важных фактов. Составление схем и диаграмм, представляющих наиболее и наименее существенные характеристики
Организация	Составление схем и диаграмм, показывающих место явления, процесса в его окружении
Атрибуция	
Оценка – высказывание мнения, базирующегося на критериях и стандартах	
Проверка	Проверка логики аргументов. Поиск критериев
Критика	Поиск лучшего метода доказательства. Поиск аргументов за и против
Создание – интеграция частей в новое целое, представление аргументов в новую структуру	
Генерация	Составление списка критериев. Использование нескольких гипотез, чтобы объяснить явление, процесс. Создание альтернативной гипотезы, базирующейся на критериях
Планирование	Создание мультимедийной презентации для представления исследования. Написание статьи
Производство	Создание журнала

Таблица 3

Классификации уровней усвоения учебного материала, разрабатываемые в России

В. П. Симонов	В. Г. Королева	В. П. Беспалько	В. Н. Максимова	М. Н. Скаткин
Различение	Репродуктивное самостоятельное воспроизведение	Ученический (деятельность по узнаванию)	Узнавание	Воспроизведение понятия
Запоминание	Репродуктивное алгоритмическое действие	Алгоритмический (решение типовых задач)	Запоминание	Распознавание понятия
Понимание	Продуктивное эвристическое действие	Эвристический (выбор действия)	Понимание	Применение понятия
Простейшие умения и навыки	Продуктивное творческое действие	Творческий (поиск решения)	Применение (тематическое, предметное, межпредметное обобщение)	Воспроизведение системы понятий
Перенос				Применение системы понятий

да системно, в течение продолжительного периода времени.

Современная эффективная методика – это комплекс педагогических технологий как более крупных методических единиц по сравнению с методами. Технологии, которые преподаватель будет использовать для изучения студентами дисциплины при когнитивно-прагматическом подходе, с одной стороны, должны быть развивающими, а с другой стороны, позволяющими формировать такие знания, умения и навыки, которые будут пригодны для решения задач на практике, в реальной жизни, в деле – профессиональном или учебном. Эти образовательные технологии должны содержать репродуктивный уровень заданий, эвристический (выбор действия) и творческий (поиск решения). В заданиях должны быть обязательно задействованы следующие уровни мыслительных умений: уровень применения, анализа, синтеза и оценки, а не только уровни знания и понимания. На выбор преподавателя предлагается несколько классификаций уровней мышления, которые помогут в составлении разноуровневых заданий (см. табл. 1, табл. 2, табл. 3).

К образовательным технологиям, позволяющим развивать индивидуальный познавательный стиль студента, можно отнести: когнитивную технологию обучения, технологию модульно-рейтингового обучения с вариативным составом учебных элементов (при условии, что студент выполняет все учебные элементы структуры модуля, т. е. выбирает для изучения самый высокий уровень, сочетающий в себе задания репродуктивного характера, эвристического и творческого), технологию проблемного обучения, технологию проектной деятельности.

В качестве примера рассмотрим использование технологии модульно-рейтингового обучения с вариативным составом учебных элементов при изучении конфигурации 1С:Торговля и склад.

• Специальность – прикладная информатика в экономике.

• Учебный курс – интегративные экономические системы.

• Название МП – «Ведение оперативного учёта в конфигурации 1С:Торговля и склад».

• Структура МП –
М1-М2-М3-М4-М5-М6-М7-М8-М9.

• Название и номер модуля – М1 «Запуск программы. Начальные настройки. Работа со справочниками».

• Структура модуля:

1-й уровень
УЭ0-УЭ1-УЭ2-УЭ3-УЭ4-УЭ7-УЭ8
(репродуктивный)

2-й уровень
УЭ0-УЭ1-УЭ2-УЭ3-УЭ4-УЭ5-УЭ7-УЭ8
(репродуктивный + эвристический)

3-й уровень
УЭ0-УЭ1-УЭ2-УЭ3-УЭ4-УЭ5-УЭ6-УЭ7-УЭ8
(репродуктивный + эвристический + творческий)

• Время на освоение материала – 4 часа.

• Необходимая для работы по модулю обязательная и дополнительная литература, оборудование, приборы, программы.

Обязательная литература:

1) Югина, Е. В. Практикум по программе 1С:Торговля и склад [Текст] : методические указания / Е. В. Югина. – Киров: Изд-во ВятГГУ, 2008. – 40 с.

Дополнительная литература:

2) Богачёва, Т. Г. 1С:Торговля и склад в вопросах и ответах [Текст] : практическое пособие / Т. Г. Богачёва. – М.: ООО «1С-Паблишинг», 2003.

3) Володина, Е. В. Практика применения программы 1С:Торговля и склад 7.7 редакция 9.2. Пример решения сквозной задачи – от ввода реквизитов до формирования отчётов [Текст] / Е. В. Володина. А. Д. Севостьянов, Ю. М. Севостьянова. – АУЦ фирмы «1С» ООО «Константа», 2004.

4) Линёва, Е. 1С:Торговля и склад 7.7 [Текст] : практический курс для начинающего пользователя / Е. Линёва, Э. Бойко, А. Ерёмин, Е. Мальчук. – М.: О-во «Знание» России. Правление, 2004.

Оборудование – персональный компьютер.

Программы – система 1С:Предприятие 7.7 с установленной конфигурацией 1С:Торговля и склад.

Содержание модуля

(Задания взяты из: *Андерсон, Дж. Когнитивная психология [Текст] / Дж. Андерсон. СПб.: Питер, 2002. 5-е изд. 496 с.)*

№ УЭ	Содержание материала	Методическое руководство по освоению материала модуля
УЭ0	<p>В результате освоения модуля Вам необходимо:</p> <p>1-й уровень</p> <ul style="list-style-type: none"> – знать основные объекты метаданных программы; – уметь создавать новую информационную базу; – чистить заполненную информационную базу; – производить первоначальный запуск программы; – выполнять начальные настройки программы; – заполнять справочники двумя способами: используя помощник начала работы и непосредственно через главное меню программы; – знать определение справочников, способы вызова справочников; – уметь приводить примеры справочников, документов, отчётов и объяснять их назначение, способы заполнения и формирования; – определять порядок работы в информационной базе; – знать основные алгоритмы работы в программе; – уметь выполнять операции в справочниках: создавать, редактировать, копировать, перемещать, искать (двумя способами) группы, подгруппы и элементы; <p>2-й уровень то же, что 1-й уровень +</p> <ul style="list-style-type: none"> – анализировать готовую базу данных; – выбирать, какой способ выполнения операции над объектом наиболее подходит в конкретном случае. <p>3-й уровень то же, что 2-й уровень +</p> <ul style="list-style-type: none"> – ставить новую задачу, решаемую средствами программы; – проектировать новую информационную базу, заполнять её необходимой информацией; – находить информацию о способах выполнения в программе поставленной задачи; – оценивать возможности, перспективы и проблемы оптимизации работы со справочниками и с информационной базой 	<p>Прочитайте цель Вашей работы над модулем, выберите уровень усвоения Вами материалов модуля, обратите внимание на то, какие учебные элементы Вам необходимо освоить.</p> <p>Если хотите, чтобы работа была увлекательной и развивающей, то выбирайте третий уровень</p>
УЭ1	<p>Ваша цель состоит в том, чтобы определить уровень готовности к освоению материала модуля.</p> <p>Вопросы:</p> <ol style="list-style-type: none"> 1. Сформулируйте определение информационной системы (в широком и узком смысле). 2. Что такое база данных? 3. Что такое СУБД? 4. Чем отличается юридическое лицо от физического? 5. Какие типы цен Вы знаете? 6. Какие существуют способы оплаты? 7. Какой документ формируется при поступлении денег в кассу? 8. Какой документ формируется при расходе денег из кассы? 9. Какой документ формируется при поступлении денег в банк? 10. Какие документы формируются при расходе денег из банка? 11. Какой документ формируется при поступлении товара на склад? 12. Какой документ формируется при списании товара со склада? 	<p>Работайте в парах, поочередно отвечая на вопросы и дополняя друг друга. После этого, работая всей группой, при участии преподавателя, обсудите вопросы, на которые Вы не смогли ответить или сомневаетесь, верно ответили на них или нет</p>
УЭ2	<p>Ваша цель – ознакомиться с первоначальными сведениями о работе в программе 1С:Торговля и склад; создать новую ИБ копированием к себе в папку папки ДВ из C:\Program Files\1Cv77; почистить базу, удалив из скопированной папки ДВ файлы *.cdx и *.dbf.</p> <p>Вопросы для закрепления:</p> <ol style="list-style-type: none"> 1. Перечислите компоненты системы 1С:Предприятие. 2. Приведите примеры типовых конфигураций системы 1С:Предприятие. 3. Перечислите, с какими объектами метаданных системы 1С:Предприятие необходимо научиться работать, чтобы освоить программу 1С:Торговля и склад? 4. Расскажите, как создать новую ИБ? 5. Объясните, как почистить заполненную ИБ? 	<p>Выслушайте новый материал, излагаемый преподавателем, отвечая на вопросы и выполняя задания. Работая индивидуально, ответьте на вопросы для закрепления (устно или письменно, на Ваш выбор)</p>
УЭ3	<p>Ваша цель – запустить программу 1С:Предприятие, добавить свою ИБ конфигурации 1С:Торговля и склад, выполнить первоначальное заполнение созданной ИБ, открыть период и выполнить предложенные начальные настройки ИБ</p>	<p>Работая индивидуально, выполните задания со с. 4–5</p>

УЭ4	Ваша цель состоит в том, чтобы выяснить, что такое справочник, каковы способы его вызова и заполнения, просмотреть предложенные справочники и заполнить незаполненные	Работая индивидуально, прочитайте теорию на с. 6–8. Составьте 2 вопроса по содержанию прочитанного. Время выполнения – 7 минут. Работая в группе, задавайте свои вопросы, отвечайте на вопросы других. Выполните задания (с. 6–11). Результат заполнения базы покажите преподавателю
УЭ5	Ваша цель – проанализировать готовую базу данных; выбрать, какой способ выполнения операции над объектом наиболее подходит в конкретном случае. Вопросы и задания: 1. Запустите ИБ DemoDB. Опишите, как Вы это сделали. 2. Исследуйте, какие настройки по умолчанию установлены в ИБ? 3. Какие справочники были заполнены? 4. Каков курс доллара? Как Вы это определили? 5. Определите номер банковского счёта организации. 6. Какие из покупателей являются физическими лицами? Как определили? 7. Объясните, какой способ поиска подходит в случае, если: а) необходимо найти наименование товара по корню второго слова в названии; б) необходимо найти товар по коду; с) необходимо найти товар по первым буквам первого слова в наименовании товара. Проверьте свой ответ. Объясните, как Вы выполнили проверку и какой результат получили. Сравните способы поиска, сделайте общий вывод о том, в каком случае какой способ лучше применять	Работая индивидуально, исследуйте и проанализируйте готовую ИБ, опишите, как Вы это сделали. Ответьте письменно на вопросы, выполните задания
УЭ6	Ваша цель – создание, проектирование и заполнение новой ИБ, оценка возможности, перспектив и проблем оптимизации работы со справочниками и с информационной базой. Задания: 1. Придумайте задачу, решаемую средствами программы 1С:Торговля и склад. 2. Спроектируйте новую ИБ для предложенной Вами задачи. Заполните её необходимой информацией. 3. Найдите в справке программы информацию о том, как задать рассчитываемые цены. Предусмотрите наличие товаров с рассчитываемыми ценами в своей базе. 4. Оцените возможность, перспективы и проблемы оптимизации работы со справочниками и с информационной базой	Работая индивидуально, выполните задания. Результат покажите преподавателю
УЭ7	Ваша цель состоит в том, чтобы определить уровень освоения материала модуля: 1. Перечислите, какие справочники необходимо заполнить перед занесением данных в справочник «Номенклатура». 2. Опишите, как удалить элемент. 3. Объясните, в какой справочник вносится банковский счёт организации. 4. Как осуществить быстрый поиск? 5. Выполните проверочную работу (с. 11) (оценивается преподавателем). 6. Исследуйте, как изменить вид ценников перед распечаткой. 7. Сравните справочники программ 1С:Торговля и склад и 1С:Бухгалтерия. Что общего и в чём отличия?	Ответьте на вопросы, выполните задания. Сверьте полученные результаты с ключом. Если Вы безошибочно выполнили задания 1–4, Вы получаете 2 балла, если 1–5 – 3 балла, 1–6 – 4 балла, 1–7 – 5 баллов. Выставьте полученные баллы в экран самоучёта
УЭ8	Сформулируйте вывод по модулю: чему Вы научились в результате выполнения модуля, все ли поставленные цели достигнуты? Опишите, какие методы работы Вы использовали при выполнении модуля?	Прочитайте вывод одногруппникам. Оцените степень удовлетворённости работой над модулем по критериям: Я, Мы, Дело – по 100-балльной системе

Условные обозначения:

МП – модульная программа, УЭ – учебный элемент,
М – модуль, ИБ – информационная база.

Примечание:

УЭ₀ – всегда интегрирующая дидактическая цель,
УЭ₁ – входная диагностика (репродуктивные задания),
УЭ₂ – УЭ_{n-2} – содержание модуля с разноуровневыми заданиями,
УЭ_{n-1} – выходная диагностика (так как при когнитивно-прагматическом подходе модуль обязательно разноуровневый, то диагностика также разноуровневая),
УЭ_n – резюме модуля.

М. А. Ямшанов, Е. В. Котельников

МЕТОД ОПРЕДЕЛЕНИЯ ТЕМАТИЧЕСКОЙ ЗНАЧИМОСТИ ЛЕКСИЧЕСКИХ ЕДИНИЦ

В работе предложен метод определения тематической значимости лексических единиц в задачах текстовой классификации на основе нейронных сетей. Приводится обоснование идеи использования нейросетевых технологий в подобных задачах, описываются структура, входные и выходные параметры, принципы функционирования нейронной сети для распознавания терминов.

1. Введение

Для определения весовых коэффициентов лексических единиц (ЛЕ) для задач текстовой классификации в настоящее время используется несколько подходов. Наибольшее распространение получил метод TF-IDF. Однако применение какого-либо одного метода статистического анализа далеко не всегда позволяет правильно определить вес ЛЕ с точки зрения тематической значимости.

В работе [1] приведены данные по результатам исследований различных ученых в области определения системы терминов текста. Отмечается, что законы распределения терминов, представленные в этих исследованиях, существенно различаются, а иногда даже противоречат друг другу. Кроме того, частоты большинства терминов в реальных текстовых выборках нерепрезентативны, следовательно, вообще не должны анализироваться на соответствие их распределения с тем или иным законом.

Таким образом, необходим метод, позволяющий учитывать характеристики ЛЕ, имеющие различную природу, и не требующий репрезентативности текстовой выборки на этапе классификации.

2. Применение нейронной сети для определения тематической значимости

В данной работе предлагается для определения тематической значимости ЛЕ использовать нейронную сеть (НС). Идея применить аппарат НС для разбиения множества ЛЕ текста по критерию тематической значимости возникла в связи с серьезными трудностями построения единого математического закона для выделения терминологической лексики. В то же время, поскольку существуют определенные закономерности в

статистических распределениях ЛЕ разных типов, есть вероятность получения более точных обобщений статистических параметров ЛЕ. На сегодняшний день наилучшими обобщающими свойствами обладают НС.

Как известно, применение НС для решения любой задачи включает два этапа: этап обучения и этап распознавания. На этапе обучения на вход НС подается обучающая выборка, состоящая из заранее отобранных и подготовленных входных и выходных векторов. В соответствии с выбранным алгоритмом обучения происходит настройка весовых коэффициентов НС, в результате которой при подаче на вход НС обучающего вектора на выходе появляется заданный выходной вектор, обозначающий класс входного вектора.

На этапе распознавания на НС поступает заранее неизвестный входной вектор. При этом на выходе появляется вектор – результат распознавания, в соответствии с которым входной вектор причисляется к одному из известных классов.

На вход НС предполагается подавать вектор, содержащий следующие параметры:

- статистические распределения характеристик ЛЕ, полученных с помощью различных методов статистического анализа, с учетом всех документов обучающей выборки;
- статистические распределения характеристик ЛЕ без учета всех документов обучающей выборки;
- признаки авторского форматирования документа (присутствие ЛЕ в заголовке, аннотации, списке ключевых слов и т. д.);
- морфологические признаки (часть речи, к которой относится ЛЕ).

Выходом НС должен являться вектор, состоящий из трех значений, оценивающих принадлежность данной ЛЕ к одному из трех множеств: L^T , L^K , L^I (см. рис. 1).

Для обучения НС необходимо подготовить обучающие выборки на основе обучающих наборов данных классификатора, в которых все ЛЕ разбиты экспертами на три множества: L^T – тематически значимые термины, L^K – контекст терминов, L^I – избыточные слова.

Таким образом, для реализации метода определения тематической значимости ЛЕ с помощью аппарата НС необходимо определить входные и выходные параметры НС, выбрать структуру НС, определить функцию активации, выбрать алгоритм обучения, подготовить обучающие выборки и провести обучение НС.

3. Формирование векторов входных и выходных параметров НС

Введем вектор X входных параметров НС для анализа ЛЕ l_k :

ЯМШАНОВ Михаил Львович – заместитель директора ФГУП НИИ СВТ по научной работе
КОТЕЛЬНИКОВ Евгений Вячеславович – кандидат технических наук, доцент по кафедре информатики и методики обучения информатике ВятГГУ
© Ямшанов М. А., Котельников Е. В., 2007

$$X_k = \{x_k^1, x_k^2, \dots, x_k^q, \dots, x_k^s\}, \quad (1)$$

где x_k^q – q -й параметр ЛЕ, s – общее количество входов НС.

Каждый параметр x_k^q вычисляется с помощью одного из статистических или лингвистических методов (признаков), которые разбиты на несколько групп.

1-я группа – методы статистического анализа параметров распределений ЛЕ с учетом всех документов обучающей выборки. К этой группе относятся параметры $x_k^1 - x_k^4$.

2-я группа – методы статистического анализа параметров распределений ЛЕ внутри документа, для которого определяется тематический профиль, без учета остальных документов обучающей выборки. Сюда относятся параметры $x_k^5 - x_k^7$.

3-я группа – признаки авторского форматирования текста. С помощью данных признаков вычисляются параметры $x_k^8 - x_k^{11}$.

4-я группа – морфологические признаки ЛЕ, учитываемые параметрами $x_k^{12} - x_k^{17}$.

Для статистического анализа обучающего набора документов будем использовать следующие известные методы:

- инверсную частоту ЛЕ (IDF);
- критерий взаимной информации (MI);
- коэффициент Galavotti – Sebastiani – Simi (GSS).

Кроме того, введем новый критерий – индекс тематичности ЛЕ.

Инверсная частота (Inverted Document Frequency, IDF). Инверсная частота ЛЕ представляет собой величину, обратную частоте, с которой данная ЛЕ встречается во всей совокупнос-

ти документов обучающего набора [2]. Соответственно, первый параметр входного вектора НС для l_k будет определяться как

$$x_k^1 = idf^k = \log \frac{m}{m_k}, \quad (2)$$

где idf^k – инверсная частота ЛЕ l_k , m_k – количество документов, содержащих l_k .

Критерий взаимной информации (Mutual Information Gain, MI). Данный параметр позволяет оценить степень взаимосвязи между тематической рубрикой c_j и ЛЕ l_k :

$$MI(l_k, c_j) = \log \frac{P(l_k, c_j)}{P(l_k)P(c_j)}, \quad (3)$$

где $P(l_k, c_j)$ – вероятность появления ЛЕ l_k в тематической рубрике c_j ,

$P(l_k)$ – вероятность появления l_k в наборе документов D ,

$P(c_j)$ – вероятность того, что документ из набора документов D принадлежит тематической рубрике c_j .

Данные вероятности будем с некоторой степенью приближения вычислять следующим образом:

$$P(l_k) \approx \frac{m_k}{m}, \quad P(c_j) \approx \frac{m_j}{m}, \quad P(l_k, c_j) \approx \frac{m_{jk}}{m}, \quad (4)$$

где m_k – количество документов, содержащих l_k , m_j – количество документов, принадлежащих рубрике c_j ,

m_{jk} – количество документов, принадлежащих рубрике c_j и содержащих ЛЕ l_k .

Для использования критерия взаимной информации в качестве входа НС усредним данный критерий по всем рубрикам [3]:

$$MI_{avg}(l_k) = \sum_{j=1}^r P(c_j) \log \frac{P(l_k, c_j)}{P(l_k)P(c_j)}. \quad (5)$$

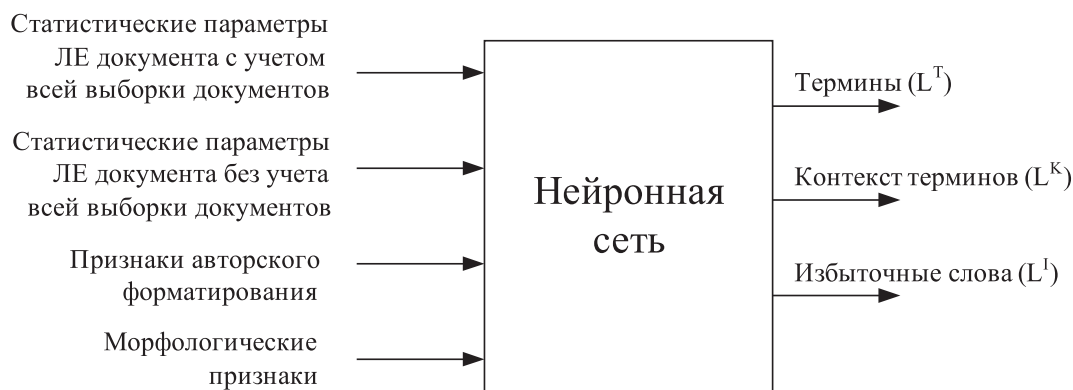


Рис. 1. Нейронная сеть для определения тематической значимости ЛЕ

Таким образом, значение второго параметра входного вектора НС с учетом формул (4) и (5) определяется по формуле:

$$x_k^2 = MI_{avg}(l_k) = \sum_{j=1}^r \frac{m_j}{m} \log \frac{m \cdot m_{jk}}{m_k \cdot m_j}. \quad (6)$$

Коэффициент GSS (коэффициент Galavotti – Sebastiani – Simi) или **упрощенный χ^2** (simplified χ^2 , $s\chi^2$). Данный параметр является более простым вариантом вычисления χ^2 – критерия и вычисляется по формуле [4]:

$$gss(l_k, c_j) = P(l_k, c_j) \cdot P(\bar{l}_k, \bar{c}_j) - P(l_k, \bar{c}_j) \cdot P(\bar{l}_k, c_j), \quad (7)$$

где $P(l_k, c_j)$ – вероятность появления ЛЕ l_k в тематической рубрике c_j ,

$P(\bar{l}_k, \bar{c}_j)$ – вероятность того, что ЛЕ l_k не появляется в документах, не принадлежащих тематической рубрике c_j ,

$P(l_k, \bar{c}_j)$ – вероятность того, что ЛЕ l_k появляется в документах, не принадлежащих тематической рубрике c_j ,

$P(\bar{l}_k, c_j)$ – вероятность того, что ЛЕ l_k не появляется в документах, принадлежащих тематической рубрике c_j .

С определенной степенью приближения можем записать:

$$\begin{aligned} P(l_k, c_j) &\approx \frac{m_{jk}}{m}, & P(\bar{l}_k, \bar{c}_j) &\approx \frac{m_{\bar{j}\bar{k}}}{m}, \\ P(l_k, \bar{c}_j) &\approx \frac{m_{j\bar{k}}}{m}, & P(\bar{l}_k, c_j) &\approx \frac{m_{\bar{j}k}}{m}, \end{aligned} \quad (8)$$

где $m_{\bar{j}\bar{k}}$ – количество документов, которые не содержат ЛЕ l_k и не входят в рубрику c_j ,

m_{jk} – количество документов, которые содержат ЛЕ l_k и при этом не входят в рубрику c_j ,

$m_{j\bar{k}}$ – количество документов, которые не содержат ЛЕ l_k , но входят в рубрику c_j .

Усредненное значение коэффициента GSS по всем тематикам:

$$gss_{avg}(l_k) = \sum_{j=1}^r P(c_j) gss(l_k, c_j). \quad (9)$$

Тогда третий параметр входного вектора НС будем вычислять следующим образом:

$$x_k^3 = \sum_{j=1}^r \frac{m_j}{m} \left(\frac{m_{jk} \cdot m_{\bar{j}\bar{k}}}{m^2} - \frac{m_{j\bar{k}} \cdot m_{\bar{j}k}}{m^2} \right). \quad (10)$$

Индекс тематичности ЛЕ. Пусть f_{ik} – частота (количество словоупотреблений) ЛЕ l_k в документе d_i .

Тогда суммарная частота l_k во всех документах обучающего набора, входящих в рубрику c_j :

$$f_{jk} = \sum_{i=1}^m f_{ik}, \quad \forall i: d_i \in c_j. \quad (11)$$

Пусть $f_{jk}^{\max} = \max_j(f_{jk})$ – максимальная суммарная частота l_k среди всех рубрик (т. е. находим рубрику, для которой количество словоупотреблений l_k максимально).

На основании этих данных будем вычислять индекс тематичности ЛЕ и, соответственно, четвертый параметр входного вектора НС следующим образом:

$$x_k^4 = \frac{f_{jk}^{\max}}{\sum_{j=1}^r f_{jk}}. \quad (12)$$

Параметры $x_k^1 \dots x_k^4$ входного вектора (1), получаемые на основе статистического анализа обучающего набора документов, носят статический характер. Они вычисляются один раз для каждой ЛЕ из единого словаря классификатора на стадии обучения и далее становятся элементами словаря (базы данных словаря) как характеристики ЛЕ.

Параметры второй группы входного вектора

(1) $x_k^5 - x_k^7$ вычисляются на основе анализа документа, для которого определяется тематический профиль. Здесь будем использовать три метода:

- частотный;
- дисперсионный;
- степенно-показательный.

Частотный метод. В качестве пятого элемента входного вектора НС будем использовать нормализованную частоту k -й ЛЕ: отношение частоты встречаемости f_{ik} ЛЕ l_k в документе d_i к максимальной частоте встречаемости ЛЕ в d_i :

$$x_k^5 = \frac{f_{ik}}{\max_{p=1 \dots n}(f_{ip})}, \quad (13)$$

где f_{ip} – частота встречаемости ЛЕ l_p в документе d_i ; n – количество ЛЕ в d_i .

Дисперсионный метод. Данный метод оценивает характеристики ЛЕ каждого документа независимо от других документов и опирается на предположение о том, что распределение общеупотребительных ЛЕ в научно-техническом тексте более равномерно, чем распределение терминов, и что с количественной стороны это может быть выражено, соответственно, меньшими и большими значениями дисперсии [1]. Преимущество данного метода перед методами типа TF-IDF – незначительная степень зависимости от частоты употребления ЛЕ в документе.

Для того чтобы определить дисперсию ЛЕ, разобьем текст документа d_i на n фрагментов.

Тогда шестой элемент входного вектора НС будем вычислять по следующей формуле:

$$x_k^6 = \sum_{r=1}^n (f_{ik}^{(r)} - \bar{f}_{ik})^2 \cdot \frac{f_{ik}^{(r)}}{m_{ir}}, \quad (14)$$

где $f_{ik}^{(r)}$ – частота употребления ЛЕ l_k в r -м фрагменте текста документа d_i ,

\bar{f}_{ik} – средняя частота употребления ЛЕ l_k в тексте документа d_i ,

m_{ir} – количество ЛЕ в r -м фрагменте документа d_i ,

Степенно-показательный метод. Здесь в качестве характеристики ЛЕ будем использовать показатель степени γ следующей функции:

$$v = b f_{ik}^\gamma, \quad (15)$$

где v – объем текста в лексических единицах, f_{ik} – частота употребления ЛЕ l_k в тексте документа d_i ,

b и γ – константы, которые следует вычислить.

Приведенная выше функция будет описывать распределение всех ЛЕ в тексте документа: чем больше показатель степени γ , тем более вероятно, что ЛЕ является тематически значимой. И наоборот, чем меньше показатель степени γ , тем более вероятно, что ЛЕ является неинформативной.

Подготовка и вычисление веса ЛЕ этим методом достаточны просты. Для этого возьмем две точки: максимальную частоту $\max f_{ik}^{(r)}$ употребления ЛЕ l_k в одном из фрагментов текста документа d_i (пусть это фрагмент r) вместе с объемом данного фрагмента n_r и абсолютную частоту f_{ik} ЛЕ l_k вместе с объемом текста n документа d_i . Таким образом, получим:

$$v = b f_{ik}^{\gamma r} \\ v_r = b (\max f_{ik}^{(r)})^\gamma.$$

Отсюда можно вычислить γ , и полагаем:

$$x_j^7 = \gamma. \quad (16)$$

Так же, как и в дисперсионном методе, этот метод оценивает веса ЛЕ каждого документа независимо от другого документа.

Третья группа параметров входного вектора НС $x_k^8 - x_k^{11}$ будет определяться на основе анализа авторского форматирования текста, поскольку такой анализ может дать дополнительную информацию о теме.

Пусть следующие элементы входного вектора НС, оценивающей тематическую значимость ЛЕ, принимают ненулевые значения в зависимости от местоположения или разметки ЛЕ:

$$x_k^8 = \eta_{k1}, \quad (17)$$

$$x_k^9 = \eta_{k2}, \quad (18)$$

$$x_k^{10} = \eta_{k3}, \quad (19)$$

$$x_k^{11} = \eta_{k4}, \quad (20)$$

где $z_{k1} = 1$, если l_k присутствует в заголовке (подзаголовках), и $z_{k1} = 0$ в противном случае;

$z_{k2} = 1$, если l_k присутствует в аннотации, и $z_{k2} = 0$ в противном случае;

$z_{k3} = 1$, если l_k присутствует в списке ключевых слов, и $z_{k3} = 0$ в противном случае;

$z_{k4} = 1$, если l_k присутствует в первых трех или последних трех предложениях текста (параграфа), и $z_{k4} = 0$ в противном случае.

Четвертая группа параметров состоит из шести элементов входного вектора НС $x_k^{12} - x_k^{17}$. Данный параметр будет принимать значение, зависящее от результатов морфологического анализа ЛЕ. В результате морфологического разбора текста ЛЕ может быть отнесена к одному из следующих множеств:

- S – существительные (в т. ч. аббревиатуры). В данном множестве можно выделить два подмножества: I – имена собственные, Ig – географические названия;

- G – глаголы;

- P – прилагательные;

- N – наречия;

- M – множество других частей речи (союзы, предлоги, междометия, артикли и т. п.).

Кроме того, при помощи одного из алгоритмов выделения составных терминов (например, [5]), ЛЕ может быть отнесена ко множеству составных терминов (устойчивых словосочетаний) St .

Определим следующие значения элемента $x_k^{12} - x_k^{17}$, зависящие от результата морфологического разбора текстового документа:

$$x_k^{12} = 1, \quad l_k \in St, \quad (21)$$

$$x_k^{13} = 1, \quad l_k \in S, \quad (22)$$

$$x_k^{14} = 1, \quad l_k \in (I \vee Ig), \quad (23)$$

$$x_k^{15} = 1, \quad l_k \in (G \vee P), \quad (24)$$

$$x_k^{16} = 1, \quad l_k \in N, \quad (25)$$

$$x_k^{17} = 1, \quad l_k \in M. \quad (26)$$

Таким образом, определено 17 параметров входного вектора (1), характеризующих различные свойства и признаки ЛЕ, которые НС должна обработать для определения тематической значимости ЛЕ.

Поскольку на синаптические веса НС обычно накладываются требования принадлежности некоторому диапазону значений, а наиболее часто используемые нелинейные функции активации

нейронов также обычно принимают значения из некоторого диапазона, на вход НС нельзя подавать входные сигналы в их истинном диапазоне величин. Поэтому необходимо провести нормирование значений векторов x_k в диапазон значений $[0,1]$:

$$x_k^q = \frac{x_k^q - \min(x_k^q)}{\max(x_k^q) - \min(x_k^q)}, \quad (27)$$

где $\max(x_k^q)$ и $\min(x_k^q)$ – соответственно максимальное и минимальное значения для q -го элемента входного вектора, вычисленные по всей обучающей выборке.

По результатам обработки вектора (1) нейронная сеть должна формировать вектор выходных параметров Y . Для оценки тематической значимости ЛЕ предлагается в выходном векторе Y использовать три элемента:

$$Y_k = \{y_k^1, y_k^2, y_k^3\}, \quad (28)$$

где y_k^1, y_k^2, y_k^3 показывают степень принадлежности ЛЕ l_k одному из множеств L^T, L^K, L^I соответственно.

При равных значениях двух параметров y , решение принимается в сторону менее тематически значимого множества, т. е. если, например, $y_k^1 = y_k^2 \Rightarrow l_k \in L^K$.

4. Структура нейронной сети

При разработке НС необходимо определить её тип, структуру, функцию активации и алгоритм обучения.

Задача определения тематической значимости ЛЕ по набору заданных параметров распределения и характеристик ЛЕ является задачей классификации. Поэтому в данном случае наилучшим решением будет выбор НС, обучающейся с учителем. Наиболее часто применяемым видом подобных НС является многослойный персептрон. В работе используется двухслойный персептрон, имеющий один скрытый слой.

Затем следует определить количество нейронов и связей между нейронами. На вход НС сети подается вектор X , состоящий из 17 различных статистических и структурных параметров ЛЕ. Соответственно, входной слой НС должен состоять из семнадцати нейронов.

Чем больше число скрытых нейронов в сети, тем более сложную задачу она может моделировать, но при этом требуется более длительное обучение и возникает опасность переобучения.

Выбор числа нейронов скрытого слоя является сложной задачей, не имеющей точного решения. Существуют лишь приблизительные оценки, связанные с количеством обучающих примеров.

Например, в [6] для числа скрытых нейронов k приводится следующая формула:

$$k = \sqrt{\frac{N}{n \cdot \log N}}, \quad (29)$$

где N – количество примеров обучающего множества, n – число входов НС.

В связи с тем, что для задач текстовой классификации количество примеров в обучающей выборке порядка 10^5 , установим число нейронов в скрытом слое равным 40.

С выходов нейронной сети необходимо получить вектор значений, позволяющий отнести ЛЕ к одному из классов: тематически значимых, контекстных или избыточных. Таким образом, необходимо три нейрона в выходном слое. Следует учесть, что множества ЛЕ, представляющие три класса, являются классическими нечеткими множествами. Действительно, на реальных текстах даже опытному эксперту будет трудно однозначно определить принадлежность ЛЕ к тому или иному классу. Поэтому целесообразно для сигналов выходных нейронов использовать значения в диапазоне $[0,1]$. Тем самым будет обеспечена возможность определять тип ЛЕ с некоторой степенью достоверности.

Разработанная НС представлена на рис. 2.

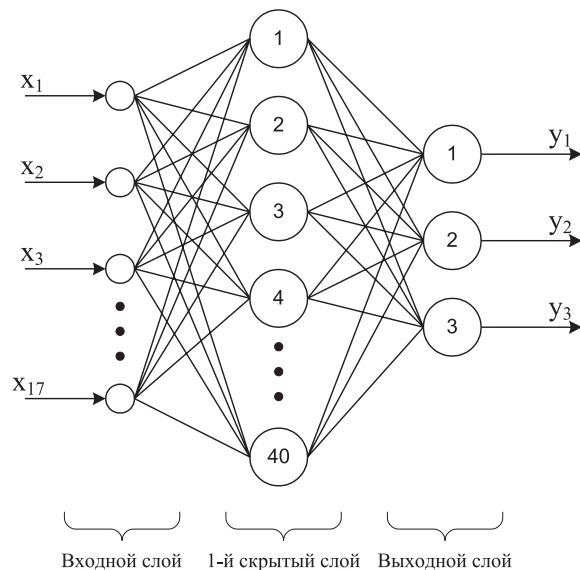


Рис. 2. Структура нейронной сети

Для выбора функции активации воспользуемся рекомендациями [7], в которых указано, что лучшим выбором для активационной функции является симметричный сигмоид вида:

$$f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right). \quad (30)$$

Сходимость НС с активационной функцией в виде гиперболического тангенса быстрее, чем у

НС с логистической функцией. Кроме того, активационная функция (30) имеет ряд полезных свойств:

- $f(1) = 1$ и $f(-1) = -1$;
- в начале координат тангенс угла наклона функции близок к единице:

$$f(0) = 1,1424;$$

- вторая производная $f(x)$ достигает своего максимального значения при $x = 1$.

Для обучения НС в данной работе будем использовать метод сопряженных градиентов (conjugate-gradient method) [8]. Этот метод более эффективен, чем классический алгоритм обратного распространения ошибки, реализованный на основе метода наискорейшего спуска, и рекомендуется для сетей с большим числом весов (большим, чем две-три сотни) и/или с несколькими выходными элементами [6, 9].

Основным преимуществом метода сопряженных градиентов является его более быстрая сходимость по сравнению с методом наискорейшего спуска. Кроме того, в методе сопряженных градиентов нет необходимости выбирать коэффициенты скорости обучения и инерции, поэтому его использование намного проще других методов.

Таким образом, для решения задачи определения тематической значимости ЛЕ выбран двухслойный перцептрон, имеющий 17 входов, 40 нейронов скрытого слоя и 3 выходных нейрона. Каждый нейрон обладает функцией активации вида (30), а для обучения НС используется метод сопряженных градиентов.

5. Функционирование нейронной сети

Функционирование разработанной НС осуществляется в два этапа: этап обучения и этап классификации. Этап обучения делится на два подэтапа – формирование обучающей выборки и собственно обучение НС. На этапе классификации обученная НС для требуемой ЛЕ выдает результат – к какому классу данная ЛЕ принадлежит: к терминам, к контексту терминов или к избыточным словам.

Рассмотрим *этап обучения*. Сначала создается обучающая выборка, состоящая из пар входной вектор – выходной вектор $\{X_k, Y_k\}$, $k = 1..n$, где n – число обучающих примеров. Для этого формируется массив документов и из каждого документа выделяются ЛЕ. При помощи методов, приводимых в первой главе, сокращается количество ЛЕ, а именно удаляются стоп-слова и слишком редко встречающиеся слова.

Затем для каждой ЛЕ вычисляется вектор входных параметров X_k , причем в зависимости от группы, к которой принадлежит параметр, расчет осуществляется либо с учетом всего мас-

сива документов, либо только на основе одного документа. Так, параметры первой группы $x_k^1 - x_k^4$ вычисляются с учетом всего массива документов и только один раз для данного массива. Вычисленные значения становятся элементами словаря как характеристики ЛЕ. Параметры второй, третьей и четвертой групп $x_k^5 - x_k^{17}$ рассчитываются по каждой ЛЕ для каждого документа отдельно. Таким образом, каждая ЛЕ в каждом документе формирует собственный входной вектор параметров.

Каждому сформированному вектору входных параметров экспертом сопоставляется выходной вектор Y_k , состоящий из трех элементов, обозначающих степень принадлежности ЛЕ одному из множеств L^T, L^K, L^I соответственно.

Таким образом, подэтап формирования обучающей выборки является полуавтоматическим: векторы входных параметров создаются автоматически на основе анализа массива документов, выходные векторы задаются вручную экспертом.

После формирования обучающей выборки наступает подэтап обучения НС – двухслойного перцептрона. Обучение проводится с использованием метода сопряженных градиентов.

Целью *этапа классификации* является установление степени принадлежности заданной ЛЕ из определенного документа одному из множеств L^T, L^K, L^I . Для достижения указанной цели на обученную НС подается вектор входных параметров ЛЕ, который формируется следующим образом. Первая группа параметров $x_k^1 - x_k^4$ извлекается из словаря (базы данных ЛЕ). Остальные параметры ЛЕ $x_k^5 - x_k^{17}$ вычисляются на основе данного документа. Выходом НС является вектор, состоящий из трех значений, указывающих принадлежность заданной ЛЕ множествам L^T, L^K, L^I .

6. Заключение

В предложенном методе для определения тематической значимости ЛЕ используется НС – двухслойный перцептрон. Применение НС можно обосновать тем, что при наличии определенных закономерностей в статистических распределениях ЛЕ попытки построения единого математического закона для выделения терминов пока не увенчались успехом. Использование НС позволяет учитывать помимо статистических параметров характеристики, связанные с авторским форматированием текста и морфологией ЛЕ. Кроме того, после этапа обучения НС не требуется, чтобы текстовая выборка являлась репрезентативной.

Таким образом, разработанная обученная НС позволяет определять степень принадлежности ЛЕ множеству терминов, множеству контекста терминов и множеству избыточных слов в автоматическом режиме без участия человека-эксперта.

Примечания

1. *Остапенко, В. Е.* Выделение и классификация терминов с помощью элементарных количественных моделей [Текст] / В. Е. Остапенко // НТИ. Сер. 2. 1989. № 11. С. 24–28.

2. *Солтон, Дж.* Динамические библиотечно-информационные системы [Текст] / Дж. Солтон. М.: Мир, 1979.

3. *Yang, Y., Pedersen J.* A comparative study on feature selection in text categorization [Text] / Y. Yang, J. Pedersen // Proc. of ICML-97, 14th International Conf. On machine Learning – Nashville, USA, 1997. P. 412–420.

4. *Galavotti, L.* Experiments on the use of feature selection and negative evidence in automated text categorization [Text] / L. Galavotti, F. Sebastiani, M. Simi // ECDL'00, 2000. P. 59–68.

5. *Добров, Б. В.* Формирование базы терминологических словосочетаний по текстам предметной области [Текст] / Б. В. Добров, Н. В. Лукашевич, С. В. Сыромятников // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды пятой Всероссийской научной конференции. СПб., 2003.

6. *Хайкин, С.* Нейронные сети: полный курс [Текст] : пер с англ. / С. Хайкин. М.: Изд. дом «Вильямс», 2006. 2-е изд. 1104 с.

7. *Le Cun, Y.* Efficient Back-prop. In Neural Networks, Tricks of the Trade, Lecture Notes in Computer Science 1524 [Text] / Y. Le Cun, L. Bottou, G. B. Orr, K.-R. Muller. Springer Verlag, 1998.

8. *Hestenes, M. R.* Methods of conjugate gradients for solving linear systems [Text] / M. R. Hestenes, E. Stiefel // Journal of Research of the National Bureau of Standards. 1952. Vol. 49. P. 135–153.

9. *Johanson, E. M.* Back-propagation learning for multi-layer feedforward neural networks using the conjugate gradient method [Text] / E. M. Johanson, F. U. Dowla, D. M. Goodman // Report UCRL-JC-104850, Lawrence Livermore National Laboratory, CA, 1990.

МАТЕМАТИКА

Э. Х. Акчурина

О ФОРМИРОВАНИИ САМОКОНТРОЛЯ У МЛАДШИХ ШКОЛЬНИКОВ В ПРОЦЕССЕ ОБУЧЕНИЯ МАТЕМАТИКЕ

В статье обосновывается, что самоконтроль, являясь одним из важнейших компонентов учебной деятельности, оказывает значительное влияние на более эффективное ее осуществление, так как позволяет ученику не только проследить ход своих действий и оценить их результаты, но и планировать возможные альтернативные действия, переосмысливать план деятельности в зависимости от изменения условий задачной ситуации. При организации работы по формированию самоконтроля младших школьников необходимо учитывать их возрастные особенности, стиль мышления и особенности эмоциональной сферы.

Современная педагогическая концепция носит гуманистический характер и определяет главной целью образования актуализацию и самоактуализацию заложенного в человеке личностного потенциала. В новой парадигме образования личность учащегося кроме социальных качеств наделяется субъективными свойствами, характеризующими ее автономию, самобытность, способность к выбору, рефлексии, саморазвитию. Главными ценностями являются личностные смыслы учения в жизни ребенка, самостоятельная учебная деятельность и жизненный опыт учащегося.

Известно, что учебная деятельность – ведущая деятельность учащихся младшего школьного возраста. Под ведущей деятельностью в детской психологии понимают такую деятельность, в процессе которой происходит формирование основных психических процессов и свойств личности, появляются главные новообразования возраста (произвольность, рефлексия, самоконтроль, внутренний план действий).

Учебная деятельность как специфическая форма индивидуальной активности осуществляется на протяжении всего обучения ребенка в школе. Младший школьный возраст – период наиболее интенсивного формирования учебной деятельности.

Целостная учебная деятельность включает в себя следующие основные потребности и мотивы:

АКЧУРИНА Эльмира Хусяиновна – аспирант кафедры теории и методики обучения математике Пензенского государственного педагогического университета
© Акчурина Э. Х., 2008

учебную задачу, учебные действия, действия контроля и оценки.

В достижении учебной самостоятельности, инициативности и ответственности младшего школьника особое значение имеет умение контролировать свою деятельность, устанавливать и устранять причины возникающих трудностей. Младший школьник, становясь субъектом учебной деятельности, занимает в ней активную творческую позицию, обнаруживает свои потенциально новые возможности. Результатом данного вида деятельности становится изменение самого ученика, уровня его развития. Поэтому формирование учебной деятельности становится приоритетной целью начального образования.

Важным компонентом учебной деятельности ученика является действие самоконтроля, когда получаемые им образовательные результаты рефлексивно выявляются и оцениваются им самим по отношению к индивидуально формулируемым целям. В отличие от контроля, осуществляемого учителем с целью проверки знаний и получения представления о возможности дальнейшего изучения материала, действие самоконтроля должно осуществляться на протяжении формирования, развития и становления учебной деятельности.

Формирование умений самоконтроля как проявление активности младшего школьника в учебном процессе является одним из условий повышения эффективности обучения, прочности и сознательности усвоения знаний учащимися, развития их познавательных способностей. Эти умения воспитывают в детях такие важные качества, как критичность ума, чувство ответственности за свою работу, уверенность в своих действиях, а также повышают интерес к учебе.

Несмотря на очевидную важность самоконтроля в учебной деятельности младших школьников, еще Ю. К. Бабанский подчеркивал, что все же мало работ, посвященных изучению проблем, связанных с формированием у школьников общеучебных умений, умений осуществлять самоконтроль и т. д.

Психологические основы самоконтроля раскрываются в трудах В. В. Давыдова, П. Я. Гальперина, А. Я. Арет, Н. Д. Левитова, Т. И. Гавваковой, В. В. Чебышевой, Л. Б. Ительсона, Г. А. Собиевой и др. Педагогические аспекты самоконтроля как существенного звена учебного

процесса освещены в работах Ю. К. Бабанского, С. И. Архангельского, Б. П. Есипова, Л. В. Жарова. В исследованиях А. С. Лынды, Н. И. Кувшинова, П. М. Эрдниева определены закономерности формирования умений и навыков самоконтроля у учащихся при изучении различных учебных предметов.

Однако результаты этих исследований недостаточно полно раскрывают проблему самоконтроля в процессе обучения математике младших школьников. Между тем самоконтроль, являясь одним из важнейших компонентов учебной деятельности, оказывает значительное влияние на более эффективное ее осуществление, так как позволяет ученику не только проследить за ходом своих действий и оценить их результаты, но и планировать действия, прогнозировать результаты, перестраивать план действий в зависимости от изменения условий, что в конечном итоге может оказать значительное воздействие на совершенствование знаний школьников.

Практика свидетельствует, что на сегодняшний день эффективность формирования самоконтроля в учебной деятельности младших школьников остается достаточно низкой. Об этом наглядно говорит большое количество слабоуспевающих и неорганизованных учеников, переходящих в среднюю школу, хотя к концу младшего школьного возраста ребенок должен хотеть учиться, уметь учиться и верить в свои силы. Известно, что многие ученики боятся приступить к решению задач, алгоритм решения которых им неизвестен. Иногда проявляется страх перед трудностями, неумение преодолевать их самостоятельно. Это и многое другое наталкивает учителя на *необходимость решения следующих проблем:*

- ученики представляют себе учебную деятельность нерасчлененной, несамостоятельной, непланируемой;
- учебные действия контролируются, как правило, педагогом, а не самими учениками;
- допущенные ошибки не замечаются и не исправляются;
- отсутствуют или не совсем четко сформулированы критерии контроля учебной деятельности;
- педагоги видят эти затруднения учащихся и способны их снимать.

Учитель должен четко поставить перед собой *задачи:*

- развивать потребности детей в самоконтроле через анализ собственных действий и их результатов;

- развивать способности к поэтапному (пооперационному) самоконтролю действий по достижению цели;

- развивать потребности в самоорганизации, самоизменении...

Приучать учащихся к самоконтролю следует уже на занятиях по арифметике, где это особенно просто, и продолжать в течение изучения всего курса математики. С первого класса необходимо нацеливать детей на то, что контролировать себя нужно сразу же, как только решили самостоятельно хотя бы один пример. Этим реализуется принцип немедленной проверки решения (решил пример – проверь себя; убедился, что твое решение верное – приступай к решению следующего примера). Такое положение в классе создается при определенных условиях. В качестве внешних условий вначале выступают материализованные индивидуальные средства обучения и использование их при самоконтроле на этапе объяснения и первичного закрепления нового учебного материала. Обучая элементам самоконтроля на этом этапе, следует стремиться выработать у детей потребность контролировать правильность полученных результатов. Этап самоконтроля с конкретными предметами должен перейти в этап самоконтроля заместителями предметов в виде рисунков, схем, чертежей и т. д. Здесь методические усилия учителю целесообразно направить, главным образом, на понимание детьми соответствия между математическими записями, образцами математических выражений и их иллюстрациями в учебниках, тетрадах на печатной основе, дидактических материалах. Эти виды работ целесообразно применять на начальной стадии формирования вычислительных приемов с постепенным уменьшением вспомогательных наглядных элементов в обучении, переходя к обучению самоконтролю, в основе которого лежат закономерности, свойства арифметических действий, взаимосвязь между компонентами, состав чисел.

При планировании работы по формированию самоконтроля необходимо учитывать возрастные особенности младшего школьника, характер его мышления и особенности эмоциональной сферы. Так, если на начальном этапе обучения идет формирование представления итогового самоконтроля, то к концу IV класса важно, чтобы дети могли использовать приемы прогнозирующего и пошагового самоконтроля. Постепенное усложнение заданий от I к IV классу способствует четкой отработке каждого действия самоконтроля, работы и ответы детей станут более грамотными.

М. С. Бердников

**ТЕОРЕМА
О ВЫПУКЛОСТИ И α -ВЫПУКЛОСТИ**

Данная работа посвящена доказательству следующей теоремы: функция f выпукла на отрезке $[a; b]$ тогда и только тогда, когда она является α -выпуклой функцией на данном отрезке. Этот критерий позволяет ослабить условия выпуклости функции в соответствующих вопросах.

Классическое определение выпуклости функции, описываемое неравенством Иенсена, было дано более ста лет назад, но наибольший интерес к этому понятию стал проявляться, начиная с 50-х гг. прошлого столетия. Постепенно стало зарождаться новое направление в математике – «выпуклый анализ». На текущий момент написано несколько книг, посвящённых теории выпуклых функций, доказано множество теорем, использующих выпуклость. В связи с этим представляется актуальным дальнейшее изучение понятия выпуклости.

В данной работе будет рассмотрен критерий, в котором доказывается, что классическая выпуклость эквивалентна более слабому понятию α -выпуклости. Прежде чем приступить к рассмотрению данной теоремы, введем необходимые понятия и докажем вспомогательную лемму.

Напомним, что функция f , определённая на выпуклом множестве $C \subseteq \mathbb{R}^n$, является:

выпуклой, если

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad (1)$$

для любых $x, y \in C$ и любого $\lambda \in [0; 1]$;

α -*выпуклой*, если

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad (2)$$

для любых $x, y \in C$ и $\alpha \in (0; 1)$.

Лемма 1. Если для функции f , определённой на выпуклом множестве $C \subseteq \mathbb{R}$, выполняются три следующих условия:

$$1) c = \alpha_1 a + (1 - \alpha_1)b; \alpha_1 \in (0; 1); a, b \in C:$$

$$f(c) \leq \alpha_1 f(a) + (1 - \alpha_1) f(b), \quad (3)$$

$$2) d = \alpha_2 a + (1 - \alpha_2)b; \alpha_2 \in (0; 1); a, b \in C:$$

$$f(d) \leq \alpha_2 f(a) + (1 - \alpha_2) f(b), \quad (4)$$

$$3) e = \alpha_3 c + (1 - \alpha_3)d; \alpha_3 \in (0; 1); c, d \in [a; b]:$$

$$f(e) \leq \alpha_3 f(c) + (1 - \alpha_3) f(d), \quad (5)$$

то $\exists \alpha_4 \in (0; 1)$, такое, что

$$e = \alpha_4 a + (1 - \alpha_4)b;$$

$$\alpha_4 \in [0; 1]; a, b \in C,$$

$$f(e) \leq \alpha_4 f(a) + (1 - \alpha_4) f(b).$$

Доказательство леммы 1. Подставим неравенства (3) и (4) в (5), приведём подобные слагаемые:

$$\begin{aligned} f(e) &\leq \alpha_3 f(c) + (1 - \alpha_3) f(d) \leq \\ &\leq \alpha_3 (\alpha_1 f(a) + (1 - \alpha_1) f(b)) + (1 - \alpha_3) \cdot \\ &\quad \cdot (\alpha_2 f(a) + (1 - \alpha_2) f(b)) = \\ &= (\alpha_1 \alpha_3 + (1 - \alpha_3) \alpha_2) f(a) + \\ &\quad + (1 - (\alpha_1 \alpha_3 + (1 - \alpha_3) \alpha_2)) f(b) \end{aligned}$$

Положим $\alpha_4 = \alpha_1 \alpha_3 + (1 - \alpha_3) \alpha_2$. По условиям леммы $\alpha_1, \alpha_2, \alpha_3 \in (0; 1)$, значит $\alpha_4 \in (0; 1)$. Подставляя α_4 в предыдущее неравенство, получим:

$$f(e) \leq \alpha_4 f(a) + (1 - \alpha_4) f(b).$$

Лемма 1 доказана.

Приступим к формулировке и доказательству отмеченного в аннотации к статье критерия.

Теорема 1. f выпукла на $[a; b]$ тогда и только тогда, когда является α -выпуклой функцией на данном отрезке.

Доказательство теоремы 1.

Из определений (1) и (2) легко видеть, что выпуклость влечёт α -выпуклость.

Очевидно, что необходимым условием как выпуклости, так и α -выпуклости функции является её непрерывность, поэтому функция f является непрерывной на $[a; b]$.

Нам остаётся доказать, что если f α -выпукла и непрерывна на $[a; b]$, то она также будет выпукла на данном отрезке.

Проведём доказательство методом от противного. Предположим, что f непрерывна и α -выпукла на $[a; b]$ и при этом не будет выпукла на нём. Сформулируем отрицание выпуклости функции f : $\exists \lambda_1 \in [0; 1], \exists x_1, y_1 \in [a; b]$:

$$f(\lambda_1 x_1 + (1 - \lambda_1) y_1) > \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1). \quad (6)$$

Предположим $x_1 < y_1$, положим

$$c = \lambda_1 x_1 + (1 - \lambda_1) y_1. \quad (7)$$

Очевидно $x_1 \leq c \leq y_1$. Подставляя (7) в (6), имеем:

$$f(c) > \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1). \quad (8)$$

Построим три бесконечные последовательности $\{x_n\}, \{y_n\}, \{d_n\}$, используя нижеприведённую схему из составляющих условных «шагов».

Шаг № 1. Запишем (2) для x_1, y_1 :

$$f(\alpha x_1 + (1 - \alpha) y_1) \leq \alpha f(x_1) + (1 - \alpha) f(y_1). \quad (9)$$

Положим $d_1 = \alpha x_1 + (1 - \alpha) y_1$, тогда $x_1 < d_1 < y_1$.

Возможны три случая взаимного расположения c и d_1 .

1) Если точки c и d_1 совпадают, то $\alpha = \lambda_1$. Заменим α на λ_1 в неравенстве (9):

$$\begin{aligned} f(\lambda_1 x_1 + (1 - \lambda_1) y_1) &\leq \\ &\leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1). \end{aligned}$$

Объединим предыдущее неравенство и (6) в двойное неравенство:

$$\begin{aligned} & \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1) < \\ & < f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \\ & \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1) \end{aligned}$$

Мы пришли к противоречию, теорема 1 доказана.

2) Если точка c лежит левее d_1 , то положим $x_2 = x_1, y_2 = d_1$.

3) Если точка c лежит правее d_1 , то положим $x_2 = d_1, y_2 = y_1$.

Шаг № 2. Из пунктов два и три шага номер один следует, что точка c лежит на отрезке $[x_2; y_2]$, длина которого строго меньше длины отрезка $[x_1; y_1]$:

$$x_1 \leq x_2 \leq c \leq y_2 \leq y_1, |x_2 - y_2| < |x_1 - y_1|.$$

Запишем (2) для $x_2; y_2$:

$$f(\alpha x_2 + (1 - \alpha) y_2) \leq \alpha f(x_2) + (1 - \alpha) f(y_2).$$

Положим $d_2 = \alpha x_2 + (1 - \alpha) y_2$, тогда $x_1 \leq x_2 < d_2 < y_2 \leq y_1$. После применения леммы 1, найдётся такое $\alpha_2 \in (0; 1)$, что:

$$f(\alpha_2 x_1 + (1 - \alpha_2) y_1) \leq \alpha_2 f(x_1) + (1 - \alpha_2) f(y_1). \quad (10)$$

Возможны три случая взаимного расположения точек c и d_2 .

1) Если точки c и d_2 совпадают, то $\alpha_2 = \delta_1$. Заменяем α_2 на λ_1 в неравенстве (10):

$$f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1).$$

Объединим предыдущее неравенство и (6) в двойное неравенство:

$$\begin{aligned} & \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1) < f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \\ & \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1). \end{aligned}$$

Мы пришли к противоречию, теорема 1 доказана.

2) Если точка c лежит левее d_2 , то положим $x_3 = x_2, y_3 = d_2$.

3) Если точка c лежит правее d_1 , то положим $x_3 = d_2, y_3 = y_2$.

Шаг № 3. Из пунктов два и три шага номер два следует, что точка c лежит на отрезке $[x_3; y_3]$, длина которого строго меньше длины отрезка $[x_2; y_2]$:

$$\begin{aligned} & x_1 \leq x_2 \leq x_3 \leq c \leq y_3 \leq y_2 \leq y_1, \\ & |x_3 - y_3| < |x_2 - y_2| < |x_1 - y_1|. \end{aligned}$$

Запишем (2) для $x_3; y_3$:

$$f(\alpha x_3 + (1 - \alpha) y_3) \leq \alpha f(x_3) + (1 - \alpha) f(y_3).$$

Положим $d_3 = \alpha x_3 + (1 - \alpha) y_3$, тогда $x_1 \leq x_2 \leq x_3 < d_3 < y_3 \leq y_2 \leq y_1$.

После применения леммы 1 найдётся такое $\alpha_3 \in (0; 1)$, что:

$$f(\alpha_3 x_1 + (1 - \alpha_3) y_1) \leq \alpha_3 f(x_1) + (1 - \alpha_3) f(y_1). \quad (11)$$

Возможны три случая взаимного расположения точек c и d_3 .

1) Если точки c и d_3 совпадают, то $\alpha_3 = \lambda_1$. Заменяем α_3 на λ_1 в неравенстве (11):

$$f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1).$$

Объединим предыдущее неравенство и (6) в двойное неравенство:

$$\begin{aligned} & \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1) < \\ & < f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1). \end{aligned}$$

Мы пришли к противоречию, теорема 1 доказана.

2) Если точка c лежит левее d_3 , то положим $x_4 = x_3, y_4 = d_3$.

3) Если точка c лежит правее d_3 , то положим $x_4 = d_3, y_4 = y_3$.

Рассмотрим шаг с номером n .

Из пунктов два и три шага номер $n - 1$ следует, что точка c лежит на отрезке $[x_n, y_n]$, длина которого строго меньше длины отрезка $[x_{n-1}, y_{n-1}]$:

$$x_1 \leq x_2 \leq \dots \leq x_{n-1} \leq x_n \leq c \leq y_n \leq y_{n-1} \leq \dots \leq y_2 \leq y_1,$$

$$|x_n - y_n| < |x_{n-1} - y_{n-1}| < \dots < |x_2 - y_2| < |x_1 - y_1|.$$

Запишем (2) для x_n, y_n :

$$f(\alpha x_n + (1 - \alpha) y_n) \leq \alpha f(x_n) + (1 - \alpha) f(y_n).$$

Положим $d_n = \alpha x_n + (1 - \alpha) y_n$, тогда

$$x_1 \leq x_2 \leq \dots \leq x_{n-1} \leq x_n < d_n < y_n \leq y_{n-1} \leq \dots \leq y_2 \leq y_1.$$

После применения леммы 1 найдётся такое $\alpha_n \in (0; 1)$, что:

$$f(\alpha_n x_1 + (1 - \alpha_n) y_1) \leq \alpha_n f(x_1) + (1 - \alpha_n) f(y_1), \quad (12)$$

$$d_n = \alpha_n x_1 + (1 - \alpha_n) y_1. \quad (13)$$

Подставим (13) в (12):

$$f(d_n) \leq \alpha_n f(x_1) + (1 - \alpha_n) f(y_1) \quad (14)$$

Возможны три случая взаимного расположения точек c и d_n .

1) Если точки c и d_n совпадают, то $\alpha_n = \lambda_1$. Заменяем α_n на λ_1 в неравенстве (12):

$$f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1).$$

Объединим предыдущее неравенство и (6) в двойное неравенство:

$$\begin{aligned} & \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1) < \\ & < f(\lambda_1 x_1 + (1 - \lambda_1) y_1) \leq \lambda_1 f(x_1) + (1 - \lambda_1) f(y_1). \end{aligned}$$

Мы пришли к противоречию, теорема 1 доказана.

2) Если точка c лежит левее d_n , то положим $x_{n+1} = x_n, y_{n+1} = d_n$.

3) Если точка c лежит правее d_n , то положим $x_{n+1} = d_n, y_{n+1} = y_n$.

В итоге мы имеем три бесконечные последовательности $\{x_n\}, \{y_n\}, \{d_n\}$. Последовательность $\{x_n\}$ неубывающая, а $\{y_n\}$ невозрастающая, при этом $\{x_n\}$ ограничена точками c и d_n сверху, а $\{y_n\}$ ограничена этими точками снизу:

$$\begin{aligned} & x_1 \leq x_2 \leq \dots \leq x_{n-1} \leq x_n \leq \dots \leq c \leq \dots \leq \\ & \leq y_n \leq y_{n-1} \leq \dots \leq y_2 \leq y_1, \end{aligned} \quad (15)$$

$$\begin{aligned} & x_1 \leq x_2 \leq \dots \leq x_{n-1} \leq x_n \leq \dots < d_n < \\ & < \dots \leq y_n \leq y_{n-1} \leq \dots \leq y_2 \leq y_1. \end{aligned} \quad (16)$$

Докажем, что последовательность отрезков $[x_n; y_n]$ является стягивающейся, то есть:

$$y_n - x_n \xrightarrow{n \rightarrow \infty} 0. \quad (17)$$

Проведём доказательство методом от противного. Предположим, что (17) не верно:

$$y_n - x_n \xrightarrow{n \rightarrow \infty} p > 0.$$

Ввиду монотонности и ограниченности последовательностей $\{x_n\}$ и $\{y_n\}$, обе они сходятся к некоторым различным действительным числам s и q соответственно:

$$x_n \xrightarrow{n \rightarrow \infty} s, y_n \xrightarrow{n \rightarrow \infty} q, s < q.$$

Воспользуемся определением предела последовательности:

$$\forall \varepsilon > 0 \exists n_1, \forall n > n_1: s - \varepsilon < x_n \leq s; \quad (18)$$

$$\forall \varepsilon > 0 \exists n_2, \forall n > n_2: q \leq y_n < q + \varepsilon. \quad (19)$$

Будем выбирать ε , исходя из следующего условия:

$$\varepsilon < \min\left(\frac{\alpha(q-s)}{1-\alpha}, \frac{(1-\alpha)(q-s)}{\alpha}\right).$$

Оно эквивалентно двум простым неравенствам:

$$\varepsilon < \frac{\alpha(q-s)}{1-\alpha}, \quad (20)$$

$$\varepsilon < \frac{(1-\alpha)(q-s)}{\alpha}. \quad (21)$$

Проведем алгебраические преобразования (20):

$$\begin{aligned} \varepsilon < \frac{\alpha(q-s)}{1-\alpha}, \quad (1-\alpha)\varepsilon < \alpha(q-s), \\ (1-\alpha)\varepsilon - \alpha(q-s) < 0, \\ (1-\alpha)\varepsilon - \alpha q + \alpha s < 0, \\ (1-\alpha)\varepsilon - \alpha q + q + \alpha s < q, \\ (1-\alpha)\varepsilon + (1-\alpha)q + \alpha s < q, \\ \alpha s + (1-\alpha)(q+\varepsilon) < q. \end{aligned} \quad (22)$$

Аналогично преобразуем (21):

$$\begin{aligned} \frac{(1-\alpha)(q-s)}{\alpha} > \varepsilon, \\ (1-\alpha)(q-s) > \alpha\varepsilon, \\ (1-\alpha)(q-s) - \alpha\varepsilon > 0, \\ (1-\alpha)q - s + \alpha s - \alpha\varepsilon > 0, \\ (1-\alpha)q + \alpha s - \alpha\varepsilon > s, \\ \alpha(s-\varepsilon) + (1-\alpha)q > s. \end{aligned} \quad (23)$$

Точку $d_n = \alpha x_n + (1-\alpha)y_n$ можно рассматривать как функцию двух переменных $d_n = g(x_n; y_n)$, определенную в прямоугольной области, определенной четырьмя точками: $(s-\varepsilon, q)$, (s, q) , $(s-\varepsilon, q+\varepsilon)$, $(s, q+\varepsilon)$. Функция g линейна, а это значит, что минимум и максимум достигаются в этих точках. Очевидно, что $g(s; q+\varepsilon) = \max(g(x_n; y_n))$, $g(s-\varepsilon, q) = \min(g(x_n; y_n))$, то есть:

$$\alpha(s-\varepsilon) + (1-\alpha)q \leq d_n \leq \alpha(s) + (1-\alpha)(q+\varepsilon). \quad (24)$$

Объединим (22), (23) и (24):

$$\begin{aligned} s < \alpha(s-\varepsilon) + (1-\alpha)q \leq d_n \leq \\ \leq \alpha(s) + (1-\alpha)(q+\varepsilon) < q. \end{aligned}$$

Запишем предыдущее неравенство более компактно:

$$s < d_n < q. \quad (25)$$

В зависимости от взаимного расположения точек d_n и c , либо $x_{n+1} = d_n$, либо $y_{n+1} = d_n$. В первом случае неравенство (25) противоречит с (18), а во втором с (19). Утверждение (17) доказано.

Последовательность отрезков $[x_n; y_n]$ есть стягивающаяся последовательность вложенных отрезков, которая обладает только одной общей для всех отрезков точкой. Исходя из неравенств (15) и (16), пределом последовательности $\{d_n\}$ является точка c :

$$d_n \xrightarrow{n \rightarrow \infty} c. \quad (26)$$

Воспользуемся теоремой о предельном переходе под знаком непрерывной функции:

$$f(d_n) \xrightarrow{n \rightarrow \infty} f(c). \quad (27)$$

Подставим (7) и (13) в (26):

$$\alpha_n x_1 + (1-\alpha_n)y_1 \xrightarrow{n \rightarrow \infty} \lambda_1 x_1 + (1-\lambda_1)y_1.$$

Проведем алгебраические преобразования:

$$\begin{aligned} \alpha_n(x_1 - y_1) + y_1 &\xrightarrow{n \rightarrow \infty} \lambda_1(x_1 - y_1) + y_1, \\ \alpha_n(x_1 - y_1) &\xrightarrow{n \rightarrow \infty} \lambda_1(x_1 - y_1), \quad \alpha_n \xrightarrow{n \rightarrow \infty} \lambda_1, \\ \alpha_n(f(x_1) - f(y_1)) &\xrightarrow{n \rightarrow \infty} \lambda_1(f(x_1) - f(y_1)), \\ \alpha_n(f(x_1) - f(y_1)) + f(y_1) &\xrightarrow{n \rightarrow \infty} \lambda_1(f(x_1) - f(y_1)) + f(y_1), \\ \alpha_n f(x_1) + (1-\alpha_n)f(y_1) &\xrightarrow{n \rightarrow \infty} \lambda_1 f(x_1) + (1-\lambda_1)f(y_1). \end{aligned} \quad (28)$$

Запишем определение предела последовательности для (27) и (28):

$$\forall \varepsilon > 0 \exists n_3, \forall n > n_3: \quad f(c) - \varepsilon < f(d_n) < f(c) + \varepsilon \quad (29)$$

$$\begin{aligned} \forall \varepsilon > 0 \exists n_4, \forall n > n_4: \\ \lambda_1 f(x_1) + (1-\lambda_1)f(y_1) - \varepsilon < \\ < \alpha_n f(x_1) + (1-\alpha_n)f(y_1) < \\ < \lambda_1 f(x_1) + (1-\lambda_1)f(y_1) + \varepsilon \end{aligned} \quad (30)$$

Будем выбирать ε , удовлетворяющее следующему условию:

$$\varepsilon \leq \frac{f(c) - \lambda_1 f(x_1) - (1-\lambda_1)f(y_1)}{2}. \quad (31)$$

Заметим, что $\varepsilon > 0$, так как $f(c) > \lambda_1 f(x_1) + (1-\lambda_1)f(y_1)$.

Преобразуем неравенство (31):

$$\begin{aligned} \varepsilon \leq \frac{f(c) - \lambda_1 f(x_1) - (1-\lambda_1)f(y_1)}{2}, \\ 2\varepsilon \leq f(c) - \lambda_1 f(x_1) - (1-\lambda_1)f(y_1), \\ \lambda_1 f(x_1) + (1-\lambda_1)f(y_1) + \varepsilon \leq f(c) - \varepsilon. \end{aligned}$$

На основании (29), (30) и предыдущего неравенства можно заключить, что для $\forall n > \max(n_3, n_4)$ справедливо $\alpha_n f(x_1) + (1-\alpha_n)f(y_1) < f(d_n)$, а это противоречит (14). Теорема доказана.

Н. А. Бушмелева, Е. В. Разова

ПРОБЛЕМЫ МАТЕМАТИЧЕСКОЙ ПОДГОТОВКИ СТУДЕНТОВ И ПУТИ ИХ РЕШЕНИЯ

Одной из характерных черт качественно нового уровня системы образования является глубокая и всесторонняя подготовка студентов в области математики и фундаментальных естественных наук. Однако планирование базовой подготовки в области фундаментальных наук находится в ведении УМО, и в новых учебных планах уровень математической подготовки предусмотрен существенно ниже, чем в прежних.

В этой статье мы проанализируем некоторые тенденции в математической подготовке студентов и связанные с ними проблемы, наблюдающиеся в последнее время: уменьшение количества часов, выделяемых на изучение математики; углубление разрыва между уровнем математических знаний выпускников школы и требованиями вуза; углубление разрыва между уровнем математических знаний студентов-выпускников и объективными потребностями практики.

1. Уменьшение объема математических дисциплин

Естественно, существуют субъективные, безусловно, негативные причины уменьшения количества часов, выделяемых на математику, такие, как внутривузовская «борьба» за перераспределение часов в пользу специальных дисциплин, нежелание самой математической кафедры искать и предлагать современные курсы и новые формы обучения и т. п.

Постараемся выявить объективные причины возникшей проблемы и найти стимулируемые ими пути совершенствования математического образования.

Многолетняя ориентация на сугубо технократический аспект образования, относительно большое количество часов на математику, отсутствие необходимости доказывать состоятельность программ по дисциплинам математического цикла привели к застою и девальвации математического, естественнонаучного образования в глазах стремительно меняющегося общества.

Кроме этого безусловным велением времени является расширение гуманитарной составляющей любого образования. Адекватным ответом было бы включение в традиционные программы по математике сведений по истории и методологии науки математики, экологии, безопасности жизнедеятельности и т. п. Но этого не было сделано, и в результате в учебных планах появились

новые дисциплины: экология, ОБЖ, основы права, культурология и др. Это повлекло за собой уменьшение объема блока естественнонаучных дисциплин в учебном плане, и математики в том числе.

Широкое внедрение компьютерных технологий в учебный процесс должно было побудить математические кафедры уделить большее внимание использованию численных методов и алгоритмов при изучении традиционных разделов. Такие попытки предпринимались и предпринимаются, но они натолкнулись на упорное сопротивление многих преподавателей. В результате появились новые дисциплины «Численные методы» и «Применение ЭВМ в...» и, как следствие, новые кафедры. Математики потеряли часы и штатные единицы.

Появление и распространение новых информационных технологий привели к осознанию новых задач образования, таких, как информатизация образования, формирование информационной культуры. В школах и вузах на всех специальностях введена новая дисциплина «Информатика». И вновь сопротивление преподавателей математики привело к тому, что они все больше отеснялись на обочину учебного процесса.

Сегодня пропагандируется и осуществляется переход на двухступенчатую систему подготовки специалистов. Вместо усиления роли математики и фундаментальных наук в базовом образовании на младшие курсы переместились специальные дисциплины, еще больше потеснив математику и другие естественнонаучные дисциплины. Правда, появилась магистратура, а с ней и ценнейшая возможность познакомить студентов с достижениями современной математики и возможностями ее применения в различных областях профессиональной деятельности и в творчестве.

Еще одной причиной уменьшения количества часов на математику явилось перераспределение учебной нагрузки в пользу самостоятельной работы студентов. В вузах были внедрены курсовые работы по математике, открывавшие новые возможности для совершенствования математического образования. Однако этот вид работ не регламентируется стандартом и учебным планом и, как следствие, не организован надлежащим образом.

Наконец, все более широкая компьютеризация влечет за собой уменьшение потребности в практических вычислительных навыках: сейчас нажатием нескольких клавиш компьютера можно решить такую задачу, с которой раньше мог справиться только хорошо подготовленный студент или специалист. Не случайно уменьшение количества часов, выделяемых на математику, сопровождается рекомендациями использовать

БУШМЕЛЕВА Наталья Александровна – кандидат педагогических наук, доцент по кафедре прикладной математики ВятГГУ

РАЗОВА Елена Владимировна – кандидат педагогических наук, доцент по кафедре прикладной математики ВятГГУ

© Бушмелева Н. А., Разова Е. В., 2008

компьютеры и увеличить долю самостоятельных занятий. К сожалению, в настоящее время, такие рекомендации являются, мягко говоря, лукавством, так как не отвечают реальной ситуации по следующим причинам.

Во-первых, отсутствует лицензионное программное и учебно-методическое обеспечение, позволяющее действительно эффективно использовать компьютеры в математическом образовании.

Во-вторых, выборочные проверки студентов первого курса показали, что уровень информационной культуры, в том числе компьютерной подготовки, вчерашних школьников удручающе низок.

В-третьих, вузовский курс информатики читается параллельно с курсом высшей математики, и учебные планы не предусматривают каких-либо форм самостоятельной работы студентов по изучению математики, за исключением обычных домашних заданий и типовых расчетов, причем из-за перегрузки преподавателей и их материальной незаинтересованности фактический объем даже этих заданий сокращается. Поэтому на первом курсе рассчитывать на интенсивное использование компьютеров пока не приходится.

Наметим некоторые пути реорганизации учебного процесса, изменения учебных программ и методики преподавания математики. По нашему мнению, необходимо сделать следующее.

- Пересмотреть содержание курса высшей математики, уменьшив его техническую составляющую и перенеся акцент с вопроса «как» (решить, вычислить и т. п.) на вопросы «что?» и «зачем?». Высвобождающееся время можно использовать для обсуждения и исследования полученных результатов, а также для включения в программу изучения важных тем и разделов современной математики, которые в настоящее время не изучаются вовсе или изучаются недостаточно.

- Подготовить и внедрить в учебный процесс учебные комплексы, содержащие лекции и практические занятия, соответствующие примерным учебным программам, а также большое количество задач для самостоятельного решения и разнообразных материалов для контроля и самоконтроля. Такие комплексы смогут одновременно выполнять функции учебника, задачника и репетитора-тренажера, не снижая значение традиционной учебной литературы.

- Разработать на основе учебных комплексов специальные рабочие тетради для студентов, которые могут распространяться в электронном виде, допускающем переработку и адаптацию. Эти файлы распечатываются и выдаются студентам *перед* лекциями (и практическими занятиями). Такие рабочие тетради должны содержать крат-

кий конспект именно тех лекций (занятий), которые будут прочитаны, и иметь широкие поля для заметок, пояснений, примеров и проч. Тогда студенты будут избавлены от необходимости записывать каждое слово преподавателя и смогут активно участвовать в интеллектуальном процессе, а преподаватели избавятся от вечной спешки, получат возможность опускаться на наиболее существенном, а также вести диалог со студентами, отвечая на их вопросы и задавая им свои.

- Оборудовать компьютерные классы для проведения некоторых (не всех!) аудиторных занятий, контрольных мероприятий и самостоятельной работы студентов, так как успешная компьютеризация зависит не от количества компьютеров, а от качества программного и методического обеспечения и создание *единой образовательно-научной информационной среды* является одним из важнейших условий успешной компьютеризации образования.

- Увеличить объем обязательных заданий с тем, чтобы студенты решали по несколько задач каждого типа и исследовали полученные решения, используя компьютерную поддержку.

Отметим, что при надлежащем аппаратном, программном и методическом обеспечении использование компьютеров позволяет экономить 60–70% времени.

2. Разрыв между уровнем математических знаний выпускников школы и требованиями вузов

Отметим наиболее характерные черты, присущие большинству первокурсников, которые не позволяют им надлежащим образом изучать высшую математику и затем эффективно применять математические методы в решении прикладных задач:

- неумение студентов отличать то, что они понимают, от того, чего они не понимают;

- неумение логически мыслить, отличать истинное рассуждение от ложного, необходимые условия от достаточных; неправильное представление о главном и второстепенном, о том, что необходимо помнить, а что можно и забыть;

- неумение устанавливать причинно-следственные связи между объектами;

- неумение вести диалог: понять вопрос преподавателя и ответить именно на него, а также сформулировать свой вопрос;

- неумение найти несколько ответов на один вопрос;

- стереотипность восприятия информации, искаженные и даже неверные стереотипы; снижение общего культурного уровня и, как следствие, непонимание литературных и исторических реминисценций, невозможность воспринять связи с законами физики и других наук;

- низкая компьютерная грамотность;
- недостаточное владение английским языком, чтобы пользоваться компьютерами и Интернетом.

Углубление разрыва между уровнем математической подготовкой выпускников школы и потребностями вузов определяется многими причинами, среди которых мы выделим следующие:

- недостаточность и неоднородность математической подготовки абитуриентов;
- взаимная несогласованность школьной и вузовской программ по математике;
- недостаточная квалификация учителей и отсутствие удобной и доступной им системы повышения квалификации и переподготовки, в частности в дистанционной форме;
- нежелание математических кафедр при составлении планов занятий учитывать уровень подготовки абитуриентов и устранять существующий разрыв;
- увеличение количества студентов в связи с потребностью общества в массовом высшем образовании.

Низкая информационная культура выпускников школ в сочетании с их завышенной самооценкой довольно часто шокирует многих преподавателей, воздвигает психологический барьер, вынуждает всячески избегать компьютеров в обучении и создает почти непреодолимые в настоящее время препятствия для эффективной компьютерной поддержки.

Предложим теперь некоторые организационно-методические мероприятия, направленные на сокращение разрыва между школьным и вузовским математическим образованием.

- Школьный курс математики должен создавать у учащегося максимально полное и цельное восприятие математической науки (от Евклида и Архимеда до наших дней).

- Целесообразно отказаться от утомительных технических подробностей, устаревших или второстепенных сведений. Напротив, представления о дискретной математике (комбинаторика, элементы теории вероятностей), об истории математической мысли, увлекательной и полной драматизма, как история любой сферы человеческой деятельности, хотя бы краткий обзор применения математики в различных областях современной науки и технологии, на наш взгляд, должны быть включены в программы школьного курса математики.

- Необходимо вернуть в школу хотя бы начальный курс логики, текстовые задачи и вообще все то, что способствует умению логически мыслить, понимать суть поставленной задачи, сосредоточиться на главном и отбросить второстепенное, развивает способность понять

мысль другого и правильно сформулировать свою.

- Программа по математике для первого курса вузов должна быть скорректирована таким образом, чтобы студенты ощущали непрерывность математического образования: то, что они уже изучали в школе, не повторяется, а рассматривается на качественно новом уровне, с иной степенью глубины и новыми целями.

Таким образом, математика на первом курсе должна стать связующим звеном между школой и вузом, «научить учиться»: планировать свое время, самому отвечать за уровень своих знаний, уметь осмыслить что и зачем (а не только, как) решается и где можно применить полученные результаты.

Студенты должны убедиться в том, что для грамотного и эффективного использования компьютеров необходимы

- знание математической терминологии, причем содержательное, а не поверхностное;
- умение правильно сформулировать задачу, которую поручается выполнить компьютеру;
- способность предвидеть конечный результат;
- умение проконтролировать правильность решения на промежуточных этапах;
- умение анализировать и исследовать полученный результат, а также оценить возможности его практического применения.

Всему этому надо учиться на лекциях и практических занятиях по математике, а в компьютерных классах – применять полученные знания и умения для выполнения заданий и контрольных работ по математике, причем компьютеры используются как эффективные помощники.

Компьютерная поддержка курса математики позволяет индивидуализировать работу со студентами особенно в части, касающейся домашних заданий и контрольных мероприятий, таким образом, чтобы каждый студент ощущал, что задания ему по силам и он продвигается от успеха к успеху. Это стимулирует интерес к предмету и делает учебу осмысленной и эффективной.

3. Разрыв между уровнем математических знаний выпускников вузов и потребностями современной науки и технологии

Основные причины углубления разрыва между уровнем математической подготовки выпускников вузов и потребностями современной науки и технологии, на наш взгляд, состоят в следующем.

1. Содержание вузовского курса математики во многом архаично и не вмещает многие достижения науки. Большое число важных разделов современной математики (качественная теория дифференциальных уравнений, функци-

ональный анализ, случайные процессы, прикладная математическая статистика, теория принятия решений и т. д.) не изучаются вовсе или изучаются часто только на лекциях без поддержки на практических занятиях и без домашних заданий и, следовательно, быстро стираются из памяти студентов и не могут быть использованы при изучении других дисциплин, как естественнонаучных и общепрофессиональных, так и специальных.

2. Учебные математические задачи дидактичны и оторваны от реальных потребностей современной науки и технологии. Чрезмерное внимание, уделяемое стандартным математическим методам решения задачи, игнорирование альтернативных методов мешает студентам применять математику при решении нестандартных творческих задач. А это приводит на практике к неоправданному упрощению постановки задачи, ненужным приближениям с единственной целью упростить процедуру решения.

3. Игнорируются или неправильно используются компьютеры из-за отсутствия методических материалов, а также удобных и эффективных форм повышения квалификации преподавателей.

4. Математическое образование оказалось, в основном, сосредоточенным на начальных курсах, а математические дисциплины на старших курсах преподаются не математиками, а преподавателями специальных кафедр.

5. В учебных планах отсутствуют новые математические курсы, в том числе гуманитарного характера, и нестандартные формы самостоятельной работы студентов.

Возражая против продвижения математики на старшие курсы, обычно говорят, что математике надо научить как можно раньше, чтобы затем в других дисциплинах можно было бы на нее опереться. Такая позиция отдает лукавством и не выдерживает критики. Во-первых, та математика, на которую опираются общетехнические дисциплины, изучается в школе, а математика, необходимая для профилирующих дисциплин, может изучаться позднее. Во-вторых, трудно учить студентов постановке задач и, особенно, осмыслению полученных результатов, когда они не знают предметного языка.

Заметим, что при определении содержания курса высшей математики часто обращаются к преподавателям специальных дисциплин. Выступая в роли экспертов, те нередко заявляют, что одни разделы у них не используется, а другие они лучше расскажут сами и т. д. Это – порочный метод: новые специалисты как раз и должны использовать то, что до них не использовалось. Иначе нынешние специалисты будут воспроизводить себе подобных.

Для преодоления разрыва между уровнем математических знаний выпускников вузов и потребностями современной науки и технологии необходимо, на наш взгляд сделать следующее.

1. Кардинально пересмотреть содержание курса математики, существенно сократив технические вопросы и избавившись от рутины; исключить или сократить разделы, дублирующие школьную программу; включить важнейшие разделы современной математики, уделив большее внимание решению задач синтеза, причем планы лекций и практических занятий должны быть разработаны с учетом компьютерной поддержки, что высвободит необходимое для новых разделов время.

2. Перенести акцент в триаде «что – как – зачем?» с вопроса «как?» (решить, вычислить и т. п.) на вопросы «что?» и «зачем?», во многих случаях оставляя решение вопроса «как» профессионалам-математикам и компьютерам, причем проблема «что делает человек, а что – компьютер» является предметом специального изучения и нуждается в методическом обосновании в каждом конкретном случае.

3. Подготовить и внедрить учебные комплексы для студентов, использовать безбумажные технологии (например, прием домашних заданий, типовых расчетов, контрольных работ в виде файлов, проверка которых будет поручена компьютеру), разработать методические пособия (печатные и электронные), содержащие подробные рекомендации по каждому занятию с учетом компьютерной поддержки, внедрить современные формы изучения математики.

4. Подготовить разнообразные математические курсы для повышения квалификации различных специалистов, в том числе в дистанционной форме.

Вывод

Выделим основные направления практической реализации наших предложений по преодолению негативных тенденций в математическом образовании:

1. Привести в соответствие программы изучения математики в вузе и в школе. Модернизировать курс математики, освободив его от рутины и перенести акцент с вопроса «как?» (решить, вычислить и т. п.) на вопросы «что?» и «зачем?». Использовать высвобождающееся время для обсуждения и исследования полученных результатов, а также для включения в программу новых разделов современной математики. Расширить преподавание математики на старших курсах и в продолжающемся образовании (переподготовке и повышении квалификации различных специалистов).

О ПУЧКОВОМ ПРЕДСТАВЛЕНИИ РЕДУЦИРОВАННЫХ ОГРАНИЧЕННЫХ ПОЛУТЕЛ

Для произвольного редуцированного ограниченного полутела U построен структурный пучок Π , являющийся аналогом пучка Ламбека в теории колец, и дано соответствующее функциональное представление полутела U .

1. Исходные понятия. Полутелом называется алгебраическая структура, являющаяся одновременно мультипликативной группой и аддитивной коммутативной полугруппой, причем операция умножения дистрибутивна относительно сложения с обеих сторон. Полутело с коммутативным умножением называется *полуполем*.

Полутела можно рассматривать и как группы с дополнительной бинарной операцией, и как делимые полукольца без нуля [7].

Класс единицы произвольной конгруэнции на полутеле называется *ядром* полутела. Подмножество A полутела U будет его ядром тогда и только тогда, когда A – нормальная подгруппа мультипликативной группы U с условием выпуклости: $u, v \in U, u + v = 1, a, b \in A$ влекут $au + bv \in A$ [4].

Решетку всех ядер (конгруэнций) полутела U относительно бинарных операций пересечения и умножения ядер (композиции конгруэнций) обозначим через $\text{Con}U$. Ядро полутела U , порожденное элементом u , назовем *главным ядром* и обозначим (u) . Произведение конечного числа главных ядер полутела назовем *конечнопорожденным ядром*. Ядро $A \in \text{Con}U$ называется *дополняемым*, если существует такое ядро $B \in \text{Con}U$, что $AB = U$ и $A \cap B = \{1\}$. Ядро (2) полутела U , где $2 = 1 + 1$, служит наименьшим подполутелом в U , являющимся ядром. Если $U = (2)$, то полутело называется *ограниченным*.

Полутело U называется *сократимым*, если в нем выполняется квазитожество $u + w = v + w \Rightarrow u = v$; *редуцированным*, если верно квазитожество $u^2 + v^2 = uv + vu \Rightarrow u = v$.

Заметим, что решетка $\text{Con}U$ не обязательно дистрибутивна, но всегда модулярна [10].

Ядро P полутела U называется *неприводимым*, если $A \cap B \subseteq P$ влечет $A \subseteq P$ или $B \subseteq P$ для любых $A, B \in \text{Con}U$. Пространство $\text{Sp}(U)$ всех неприводимых ядер полутела U , рассматриваемое со

2. Оборудовать компьютерные классы надлежащим программным и методическим сопровождением и создать единую образовательно-научную информационную среду, позволяющую эффективно использовать компьютеры для проведения аудиторных занятий, контрольных мероприятий и особенно для самостоятельной работы студентов.

3. Разработать и внедрить учебную литературу как на традиционных бумажных, так и на электронных носителях (учебные комплексы, электронные учебники и учебные пособия, рабочие тетради для студентов), а также контролирующие и тренирующие компьютерные пакеты.

Если предложенные нами меры будут реализованы, то математическое образование приобретет новый облик, основные черты которого состоят в следующем:

1. Изучение математики освобождено от рутины, повторов, технических подробностей, его форма и содержание соответствуют современным достижениям науки и технологии.

2. Студенты полностью обеспечены современной учебной литературой, имеют рабочие тетради для лекций, практических занятий и домашних заданий, в том числе и электронные. Изменились функции преподавателя: он теперь выступает в роли руководителя и консультанта, а студенты в основном работают самостоятельно.

3. Компьютерные классы с надлежащим программным обеспечением позволяют преподавателям проводить некоторые практические занятия и контрольные мероприятия, а студентам – выполнять все домашние задания, причем программное обеспечение дает возможность преподавателю выбирать темы, содержание и уровень сложности контрольных работ и индивидуальных заданий, а проверку осуществляют компьютеры.

4. Преподаватели получают различные дополнительные доходы (гонорары, гранты, плату за консультации и пр.), избавлены от спешки, рутины, имеют возможность повышать свою квалификацию в удобной для них форме, в том числе и дистанционной, заниматься научной работой, создавать учебные пособия (печатные и электронные), статьи и книги.

ВЕЧТОМОВ Евгений Михайлович – доктор физико-математических наук, профессор, зав. кафедрой высшей математики ВятГГУ, академик Европейской академии естествознания
© Вечтомов Е. М., 2008

стоуновской топологией, назовем *неприводимым спектром* полутела U . Открытыми множествами топологического пространства $\text{Sp}(U)$ служат в точности множества $D(A) = \{P \in \text{Sp}(U) : A \notin P\}$, взятые по всем ядрам A полутела U . Его подпространство $\text{Max}U$, состоящее из всех максимальных ядер, называется *максимальным спектром* полутела U . Аналогичные обозначения сохраним для пространств неприводимых идеалов и максимальных идеалов кольца $T : \text{Sp}(T)$ и $\text{Max}T$.

Псевдодополнением ядра $A \in \text{Con}U$ называется наибольшее ядро A^* , такое, что $A \cap A^* = \{1\}$. Для произвольного неприводимого ядра P полутела U положим

$$O_P = \{u \in U : \exists v \in U \setminus P (u) \cap (v) = \{1\}\}.$$

Для неприводимого идеала Q редуцированного кольца T множество

$$O_Q = \{t \in T : \exists s \in T \setminus Q TsT \cap TtT = 0\} = \{t \in T : s \in T \setminus Q st = 0\} = \{t \in T : \text{Ann}t \notin Q\}$$

является идеалом кольца T , содержащимся в Q . Идеал Q кольца T называется *первичным*, если для любых $a, b \in T$ включение $aTb \subseteq Q$ влечет $a \in Q$ или $b \in Q$. Первичные идеалы в любом кольце неприводимы. Как обычно, через $\text{Sp}T$ обозначим *первичный спектр* кольца T – подпространство неприводимого спектра $\text{Sp}(T)$ кольца T , состоящее из всевозможных его первичных идеалов.

2. Редуцированные ограниченные полутела.

Всякое сократимое полутело U вкладывается в свое кольцо разностей $R = R(U)$. Любое ограниченное полутело U сократимо, и решетка $\text{Con}U$ его ядер канонически изоморфна решетке $\text{Id}R$ всех идеалов кольца разностей R [8, теорема 1]. Именно [6, п. 3], следующие отображения $\gamma : \text{Id}R \rightarrow \text{Con}U$ и $\delta : \text{Con}U \rightarrow \text{Id}R$ устанавливают изоморфизм решеток $\text{Id}R$ и $\text{Con}U$:

$$\gamma(I) = (I + 1) \cap U \text{ для всех } I \in \text{Id}R,$$

$$\delta(A) = (A - 1)U \text{ для всех } A \in \text{Con}U.$$

При этом $\gamma(I \cap J) = \gamma(I) \cap \gamma(J)$ и $\gamma(I + J) = \gamma(I) \cdot \gamma(J)$ для любых $I, J \in \text{Id}R$.

Редуцированность сократимого полутела U эквивалентна редуцированности его кольца разностей R , т. е. отсутствию в R ненулевых нильпотентных элементов. Редуцированные кольца обладают свойством симметричности (коммутативности в нуле): если произведение нескольких элементов кольца, взятых в некотором порядке, равно 0, то равно 0 и произведение этих элементов, взятых в любом другом порядке.

Предложение 1. Для любого неприводимого идеала Q редуцированного кольца T с единицей равносильны следующие условия:

- 1) Q – минимальный неприводимый идеал;
- 2) Q – минимальный первичный идеал;
- 3) $Q = O_Q$.

Доказательство проведем по циклу $1) \Rightarrow 2) \Rightarrow 3) \Rightarrow 1)$.

$1) \Rightarrow 2)$. Докажем первичность минимального неприводимого идеала Q . Для этого образуем мультипликативное множество M , элементами которого служат всевозможные произведения элементов из $T \setminus Q$. Покажем, что $0 \notin M$. Предположим от противного, что $0 = a_1 \cdot \dots \cdot a_n$ для некоторых $a_1, \dots, a_n \in T \setminus Q$. Тогда $Ta_1T \cap \dots \cap Ta_nT = 0 \subseteq Q$ в редуцированном кольце T . Поэтому какой-то главный идеал Ta_kT ($k = 1, \dots, n$) лежит в неприводимом идеале Q , что невозможно. Значит, существует первичный (неприводимый) идеал P , не пересекающийся с M , а тем более с $T \setminus Q$. Откуда $P \subseteq Q$. Стало быть, $Q = P$ – минимальный первичный идеал кольца T .

Импликация $2) \Rightarrow 3)$ – это известный факт [5, лемма 8.1].

$3) \Rightarrow 1)$. Пусть $Q = O_Q$ и $P \subseteq Q$ для неприводимого идеала P кольца T . Если $t \in Q$, то $TsT \cap TtT = 0$ для некоторого элемента $s \in T \setminus Q \subseteq T \setminus P$, т. е. $TtT \subseteq P$ и $t \in P$. Значит, $P = Q$, и Q – минимальный неприводимый идеал кольца T .

Далее, U – произвольное *редуцированное ограниченное полутело*, и R – его кольцо разностей. Изоморфизмы δ и γ сохраняют неприводимость и конечную порожденность ядер и идеалов – элементов решеток $\text{Con}U$ и $\text{Id}R$ соответственно. Заметим, что конечнопорожденные ядра (идеалы) полутела U (кольца R) суть в точности компактные элементы алгебраической решетки $\text{Con}U$ ($\text{Id}R$) [2, глава VIII, пункты 4, 5].

Предложение 2. *Неприводимые спектры полутела U и его кольца разностей R канонически гомеоморфны: $\text{Sp}(U) \approx \text{Sp}(R)$.*

Для любых $A, B \in \text{Con}U, I, J \in \text{Id}R, u, v \in U$ имеем: $A \cap B = \{1\} \Leftrightarrow \delta(A) \cap \delta(B) = \{0\} = 0 \Leftrightarrow \delta(A) \cdot \delta(B) = 0$; (1) $I^* = \text{Ann}I = \{r \in R : rI = \{0\}\}$ и $\delta(A^*) = \text{Ann}\delta(A)$; (2) $\text{Ann}(I + J) = \text{Ann}I \cap \text{Ann}J$ и $(AB)^* = A^* \cap B^*$; (3) $\delta((u)) = R(u - 1)R$ – главный идеал кольца R ; (4) $(u) \cap (v) = \{1\} \Leftrightarrow (u - 1)(v - 1) = 0 \Leftrightarrow uv + 1 = u + v$. (5)

Проверим два последних равенства. Для $u \in U$ верна цепочка включений

$$(u) = \gamma(\delta((u))) = \gamma(((u) - 1)U) \supseteq \gamma(R(u - 1)R) = (R(u - 1)R + 1) \cap U \supseteq (u),$$

откуда $\delta((u)) = \delta(\gamma(R(u - 1)R)) = R(u - 1)R$.

Из (1) и (4) вытекает (5):

$$(u) \cap (v) = \{1\} \Leftrightarrow R(u - 1)R \cdot R(v - 1)R = 0 \Leftrightarrow (u - 1)(v - 1) = 0.$$

В силу (4) получаем также

Предложение 3. *Если каждое конечнопорожденное ядро полутела U является главным ядром, то и все конечнопорожденные идеалы его кольца разностей – главные.*

Поскольку на основании (3) для любых $P \in \text{Sp}(U)$ и $Q \in \text{Sp}(R)$

$O_P = \cup \{A = (u_i) \cdot \dots \cdot (u_n) \in \text{Con}U : A^* \notin P\}$ и $O_Q = \cup \{I = Rr_1R + \dots + Rr_nR \in \text{Id}R : I^* \notin Q\}$, то имеет место равенство

$$\delta(O_P) = O_{\delta(P)}. \quad (6)$$

Как следствие указанных соотношений получаем

Предложение 4. $\text{Con}U$ – решетка с псевдодополнениями, для любого неприводимого ядра P полутела U множество O_p является ядром в U . Кроме того,

$$\bigcap \{O_p : P \in \text{Sp}(U)\} = \{1\}. \quad (7)$$

Остается проверить равенство (7). Для $u \in \bigcap O_p$ имеем $(u)^* \not\subseteq P$ при всех $P \in \text{Sp}(U)$. Поэтому $(u)^* = U$ и $(u) = (u) \cap (u)^* = \{1\}$, т. е. $u = 1$.

Из предложения 1 и формулы (6) вытекает

Предложение 5. Минимальность неприводимого ядра P полутела U эквивалентна равенству $P = O_p$.

3. Структурный пучок и функциональное представление полутела U . Пучки полутел и соответствующие пучковые представления полутел начали изучаться в работах [9]–[12].

Дадим определение пучка полутел (см. [3], [5]). Пучком полутел называется тройка $\langle X, \Pi, \rho \rangle$, удовлетворяющая следующим условиям:

- 1) X и Π – топологические пространства, называемые соответственно базисным и накрывающим пространствами пучка;
- 2) $\rho: \Pi \rightarrow X$ – локальный гомеоморфизм;
- 3) для каждой точки $x \in X$ множество $U_x = \rho^{-1}(x)$ является полутелом, называемым слоем пучка в точке x ;
- 4) операции полутела непрерывны в Π ;
- 5) отображение, переводящее каждую точку $x \in X$ в единицу $1_x \in U_x$, непрерывно.

Тогда $\Pi = \bigcup U_x$, и мы говорим, что Π – пучок полутел U_x над пространством X . Сечением пучка Π над $Y \subseteq X$ называется любое непрерывное отображение $s: Y \rightarrow \Pi$, удовлетворяющее условию $\rho \circ s = 1_Y$. Сечения над X будем называть просто сечениями пучка Π . Множество $\Gamma(X, \Pi)$ всех сечений пучка Π с поточечно определенными операциями сложения и умножения сечений будет полутелом. Пучок Π полутел U_x называется факторным, если для любых точки $x \in X$ и элемента $u \in U_x$ существует сечение $s \in \Gamma$, проходящее через u : $s(x) = u$.

Рассмотрим семейство ядер (O_p) данного полутела U , индексированное точками P топологического пространства $\text{Sp}(U)$. Оно является открытым семейством ядер ρ , т. е. при любом $u \in U$ множество $\{P \in \text{Sp}(U) : u \in O_p\}$ открыто в неприводимом спектре $\text{Sp}(U)$. Действительно,

$$\begin{aligned} & \{P \in \text{Sp}(U) : u \in O_p\} = \\ & = \{P \in \text{Sp}(U) : (u)^* \not\subseteq P\} = D((u)^*). \end{aligned}$$

Поэтому существует пучок $\Pi = \Pi(U)$ факторполутел U/O_p полутела U над компактными T_0 -пространством $\text{Sp}(U)$ (см. [5, с. 19–20]). Получаем факторный структурный пучок полутела U , аналогичный пучку Ламбека для колец [1], [5].

Положим $\Gamma = \Gamma(\text{Sp}(U), \Pi)$ – полутело всех сечений пучка Π . Сечениями нашего пучка Π являются непрерывные отображения $s: \text{Sp}(U) \rightarrow \Pi$, для которых $s(P) \in U/O_p$ при любого $P \in \text{Sp}(U)$. Зададим отображение $\alpha: U \rightarrow \Gamma$ формулой $\alpha(u)(P) = uO_p \in U/O_p$ для всех $P \in \text{Sp}(U)$.

Легко видеть, что α будет гомоморфизмом полутел, причем в силу (7) – гомоморфным вложением полутела U в полутело сечений Γ . Гомоморфизм α и является искомым функциональным представлением данного полутела U сечениями пучка Π .

Существует изоморфное пучковое представление Ламбека $\hat{}$ редуцированного кольца разностей R над первичным спектром $\text{Spec}R$ [1]. Как и для полутела U , строится функциональное представление $\alpha': R \rightarrow \Gamma(\text{Sp}(R), \Pi')$. Слоями пучка Π' служат факторкольца R/O_Q кольца R по всем неприводимым идеалам Q в R . Для $r \in R$ имеем $\alpha'(r) = r^{\hat{}}$ на подпространстве $\text{Spec}R$ неприводимого спектра $\text{Sp}(R)$.

Предложение 6. Представление α' есть изоморфизм.

Доказательство. Возьмем $t \in \Gamma(\text{Sp}(R), \Pi')$. Сужение t на $\text{Spec}R$ совпадает с $r^{\hat{}}$ для подходящего элемента $r \in R$. Сечения t и $\alpha'(r)$ пучка Π' равны на $\text{Spec}R$, а значит, равны на открытом множестве $W \supseteq \text{Spec}R \supseteq \text{Max}R$ пространства $\text{Sp}(R)$. Имеем $W = D(I)$ для некоторого идеала I кольца R . Этот идеал I не содержится ни в одном максимальном идеале кольца R , т. е. $I = R$. Поэтому $W = D(I) = \text{Sp}(R)$, и $t = \alpha'(r)$.

Предложение 7. Для любого неприводимого ядра P полутела U факторкольцо $R/O_{\delta(P)}$ служит кольцом разностей факторполутела U/O_p , полутело U/O_p редуцировано, ограничено и $O_{p/O_p} = \{1\}$.

Доказательство. Пусть φ – ограничение на U канонического гомоморфизма $R \rightarrow R/O_{\delta(P)}$. Покажем, что $\varphi^{-1}(1) = O_p$. Тогда факторполутело U/O_p будет вложено в $R/O_{\delta(P)}$ и $R/O_{\delta(P)} = U/O_p - U/O_p$, так как $R = U - U$. Для произвольного $u \in U$ имеем:

$$\begin{aligned} u \in \varphi^{-1}(1) & \Leftrightarrow u + O_{\delta(P)} = 1 + O_{\delta(P)} \Leftrightarrow \\ & \Leftrightarrow u - 1 \in O_{\delta(P)} \Leftrightarrow \exists r \in R \setminus O_{\delta(P)} (u - 1)r = 0 \Leftrightarrow \\ & \Leftrightarrow \exists v \in U (v - 1 \in R \setminus O_{\delta(P)} \& (u - 1)(v - 1) = 0) \Leftrightarrow \\ & \Leftrightarrow \exists v \in U \setminus P (u - 1)(v - 1) = 0 \Leftrightarrow \\ & \Leftrightarrow \exists v \in U \setminus P (u) \cap (v) = \{1\} \Leftrightarrow u \in O_p. \end{aligned}$$

Поскольку факторкольцо редуцированного кольца R по идеалу вида $O_{\delta(P)}$ также редуцировано, то и полутело U/O_p будет редуцированным. Остается заметить, что ограниченность полутела сохраняется при переходе к гомоморфным образам.

Вернемся к пучковому представлению α полутела U . В силу предложений 2 и 7 справедливо включение $\Gamma = \Gamma(\text{Sp}(U), \Pi) \subseteq \Gamma(\text{Sp}(R), \Pi')$ и

МИНИ-ТОПОЛОГИИ

В статье обосновывается роль конечных математических объектов и их моделей в обучении математике. На наглядных диаграммах проиллюстрированы основные типы абстрактных математических структур. При этом наибольшее внимание уделено описанию топологических пространств, имеющих не более четырех элементов.

1. О роли конечных объектов в математике

В математической науке и дидактике математики одинаково значимы интуитивно-содержательная и абстрактно-логическая составляющие, индуктивный и дедуктивный способы формирования и развития знания. Весьма важно гармоничное сочетание дискретного и непрерывного в изучении математики и в понимании ее характера. Человек – существо конечное, проявляемое дискретно, посредством конечных последовательностей интеллектуальных и физических актов. Мыслящий субъект порционно воспринимает и познает мир. Познаваемый объект предстает в обозримом виде, ограниченном во времени и в пространстве. Таковы психика человека и психология познания. Давид Гильберт и Герман Вейль называли математику наукой о бесконечном. И действительно, содержательной основой математики служат счетный натуральный ряд и несчетный континуум (числовой и геометрический). Но и эти фундаментальные понятия становятся, а уж тем более формализуются и строго определяются, на уровне элементов и их классов, с помощью конечных схем аксиом.

Моделирование через конечное, посредством конечных объектов – универсальный метод научного познания. В математике моделирование проявляется в построении примеров и контрпримеров понятий и теорий. Освоение модельных примеров основополагающих математических теорий – отправной и важнейший этап обучения математике, это тот предметный фундамент, на котором стоит весь курс математики и отдельные его части. Моделирование включает также реализацию и визуализацию математических абстракций посредством осязаемых, наглядных – конечных и ограниченных – образов: различных графов, диаграмм Эйлера-Венна для множеств, таблиц Кэли конечных алгебр, диаграмм Хассе конечных упорядоченных множеств и т. п. К мо-

$\alpha' = \alpha$ на полутеле U . По предложению 6 кольца $\Gamma(\text{Sp}(R), \Pi')$ и R канонически изоморфны и для любого $s \in \Gamma$ имеем $s = \alpha'(u - v) = \alpha'(u) - \alpha'(v) = \alpha(u) - \alpha(v)$ для некоторых $u, v \in U$. Поэтому полутела $\alpha(U) \subseteq \Gamma$ обладают одним и тем же кольцом разностей $\Gamma(\text{Sp}(R), \Pi') - R$.

Предложение 8. Полутело U изоморфно подполутелу полутела $\Gamma(\text{Sp}(U), \Pi)$ всех сечений пучка редуцированных ограниченных факторполутел $U/O_p, P \in \text{Sp}(U)$.

Следующей задачей является исследование условий на полутело U , при выполнении которых $\alpha(U) = \Gamma$, т. е. условий полноты (изоморфности) пучкового представления α . Пока приведем одно достаточное условие изоморфности α .

Следствие. Если U – максимальное подполутело своего кольца разностей, то α – изоморфизм.

Примечания

1. Lambek, J. On representations of modules by sheaves of factor modules [Text] / J. Lambek // Can. Math. Bull. 1971. V. 14. № 3. P. 359–368.

2. Биркгоф, Г. Теория решеток [Текст] / Г. Биркгоф. М.: Наука, 1984. 568 с.

3. Бредон, Г. Теория пучков [Текст] / Г. Бредон. М.: Наука, 1988. 312 с.

4. Hutchins, H. C. Homomorphisms and kernels of semifields [Текст] / H. C. Hutchins H. J. Weinert // Periodica Mathematica Hungarica. 1990. V. 21 (2). P. 113–152.

5. Вечтомов, Е. М. Функциональные представления колец [Текст] / Е. М. Вечтомов. М.: МПГУ, 1993. 191 с.

6. Варанкина, В. И. Полукольца непрерывных неотрицательных функций: делимость, идеалы, конгруэнции [Текст] / В. И. Варанкина, Е. М. Вечтомов, И. А. Семенова // Фундаментальная и прикладная математика. 1998. Т. 4. № 2. С. 493–510.

7. Golan, J. S. Semirings and their applications [Текст] / J. S. Golan. Kluwer Academic Publishers. Dordrecht; Boston; London, 1999. 381 p.

8. Черанева, А. В. О конгруэнциях на полутелах [Текст] / А. В. Черанева // Чебышевский сборник. 2005. Т. 6. Вып. 4 (16). С. 164–171.

9. Вечтомов, Е. М. К теории полутел [Текст] / Е. М. Вечтомов, А. В. Черанева // Успехи математических наук. 2008. Т. 63. Вып. 2. С. 161–162.

10. Вечтомов, Е. М. Неприводимые ядра полутел [Текст] / Е. М. Вечтомов, А. В. Черанева // Математический вестник педвузов и университетов Волго-Вятского региона. 2008. Вып. 10. С. 25–31.

11. Вечтомов, Е. М. Пучки полутел над нульмерным компактом [Текст] / Е. М. Вечтомов, А. В. Черанева // Математический вестник педвузов и университетов Волго-Вятского региона. 2008. Вып. 10. С. 32–44.

12. Вечтомов, Е. М. Функциональные представления полутел [Текст] / Е. М. Вечтомов // Тезисы докладов Международной алгебраической конференции, посвященной 100-летию А. Г. Куроша. М.: МГУ, 2008. С. 56–58.

ВЕЧТОМОВ Евгений Михайлович – доктор физико-математических наук, профессор, зав. кафедрой высшей математики ВятГГУ, академик Европейской академии естествознания

ПУРТОВА Светлана Анатольевна, студентка V курса физико-математического факультета ВятГГУ
© Вечтомов Е. М., Пуртова С. А., 2008

делированию можно отнести и разнообразные теоремы представления и координатизации математических объектов, в том числе функциональные представления алгебраических систем сечениями пучков [1].

Математические рассуждения финитарны, алгоритмичны и принципиально конечны. Доказательство есть конечная последовательность определенным образом связанных формул. Вычисления и преобразования таковы же. Геометрически непрерывные объекты изучаются посредством производных от них дискретных объектов – математических структур арифметической или алгебраической природы. Сами бесконечные объекты рассматриваются в свете различных условий конечности. Достаточно вспомнить компактность в топологии, различные формы математической индукции, условия обрыва цепочек подалгебр или конгруэнций (нетеровость и артиновость) и конечную порожденность в общей алгебре.

Выделим основные мотивационные аспекты конечного моделирования в обучении математике.

1. Модели в математике могут выступать в качестве подтверждающих примеров (верификация) или опровергающих контрпримеров (фальсифицируемость). Понимание и построение простых моделей, в том числе простейших конечных объектов, – необходимая составляющая математического образования в вузе. Процесс математического познания есть чередование моделей-объектов и моделей-теорий, происходящее при смене формализации, или знакового, идеального моделирования, и интерпретации, или предметного, материального моделирования [2, § 5].

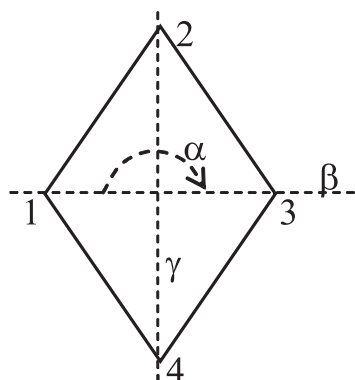
2. Конкретные математические модели – это содержательно-значимая сторона математики. Исходные модели служат основой и предпосыл-

кой интуитивно-образного познания. Таковыми являются основные числовые системы, обычное евклидово пространство, кольца классов вычетов целых чисел, булеаны и т. п.

3. Абстрактные математические объекты – это фундаментальные математические структуры и их конгломераты. По Бурбаки, к ним относятся алгебраические, порядковые и топологические структуры, взятые в чистом виде. Сюда следует добавить пространства с мерой и структуры инцидентности [3]. Простые примеры и контрпримеры групп, колец, решеток, метрических и топологических пространств, геометрий и графов служат интерпретацией фундаментальных математических теорий.

4. Велика логическая роль математических моделей как в научном, так и в методическом плане. Дидактически и психологически никакая формальная аксиоматика невозможна без семантики. Доказательство непротиворечивости и независимости аксиом заключается в построении соответствующих моделей, зачастую простейших моделей. Назовем аксиоматику Пеано натурального ряда, определения группы, порядка, метрики и топологии, исчисление высказываний, теорию конечных плоскостей. И здесь желательно ознакомление студентов с языком математической логики и с началами теории моделей.

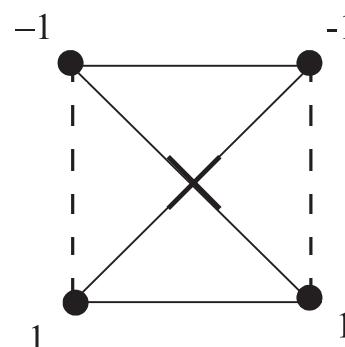
5. С развитием компьютерной алгебры постоянно возрастает значение конечных математических объектов. Различные мини-структуры (то есть конечные алгебры, графы и пространства с небольшим числом элементов и т. п.) важны также в методологическом, эвристическом и прикладном отношении. Компьютерное моделирование выполняет функции математического эксперимента и прогнозирования, служит методом так называемой супериндукции.



Группа симметрий ромба

–	1	α	β	γ
1	1	α	β	γ
α	α	1	γ	β
β	β	γ	1	α
γ	γ	β	α	1

Таблица Кэли G



$G \cong \{1, -1\} \times \{1, -1\}$

Рис. 1

2. Некоторые конечные модели

Группа Клейна. Изобразим четверную группу Клейна G в трех видах. Абстрактно группа G является нециклической группой четвертого порядка (см. рис. 1).

Группа Клейна изоморфна: группе $\{1, \alpha, \beta, \gamma\}$ самосовмещений ромба с операцией суперпозиции движений, прямому произведению двух экземпляров мультипликативной группы $\{1, -1\}$, или $G \cong \mathbf{Z}_2 \times \mathbf{Z}_2$, а также подгруппе $\{1, (13), (24), (13)(24)\}$ симметрической группы S_3 .

Упорядоченные множества с $n \leq 4$ элементами. Их удобно изображать диаграммами Хассе. Укажем – с точностью до порядкового изомор-

физма – все упорядоченные множества с $n \leq 4$ элементами (рис. 2).

Среди них только 5 решеток: 4 цепи и «ромбик». Легко подсчитать число упорядочений n -элементного множества: при $n = 1$ существует 1 упорядочение, при $n = 2 - 3$ упорядочения, при $n = 3 - 19$ упорядочений и при $n = 4 - 219$ упорядочений.

Конечные решетки. Все решетки с n элементами ($n \geq 3$) получаются из $(n - 2)$ -элементных упорядоченных множеств добавлением к ним новых наибольшего и наименьшего элементов. Так, все пять пятиэлементных решеток получаются из соответствующих упорядоченных множеств (1)–(5) (рис. 3).

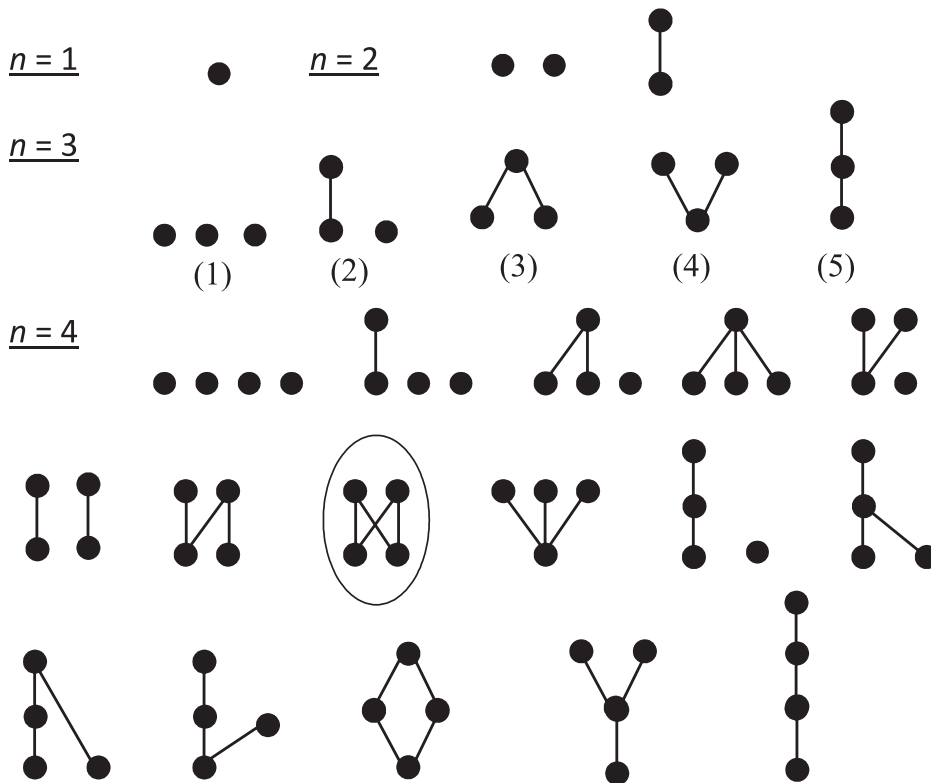


Рис. 2

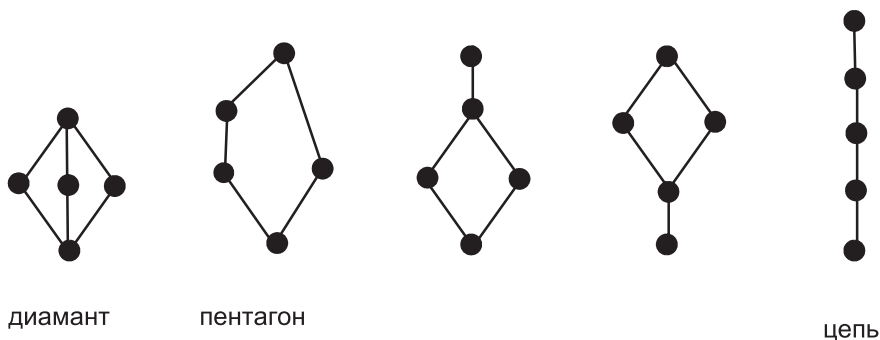


Рис. 3

А всевозможные 15 шестиэлементных решеток строятся из 15 четырехэлементных упорядоченных множеств, только одно «обведенное» множество дает шестиэлементное упорядоченное множество, не являющееся решеткой.

Графы с 4 элементами. Изобразим с точностью до изоморфизма все 11 различных простых графов с четырьмя вершинами (рис. 4).

Конечные плоскости

На примерах конечных плоскостей можно просто, наглядно и увлекательно показать и сопоставить главные геометрии в миниатюре: евклидову (аффинную) и проективную геометрии, геометрию Лобачевского, указать другие логические возможности (рис. 5). На моделях I и II

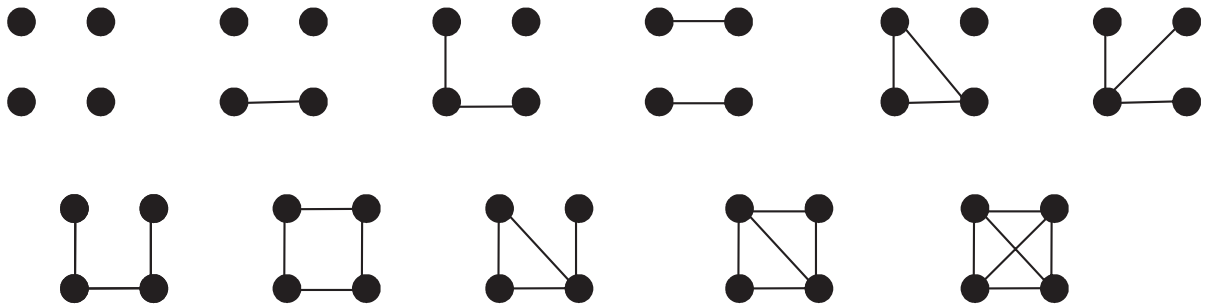


Рис. 4

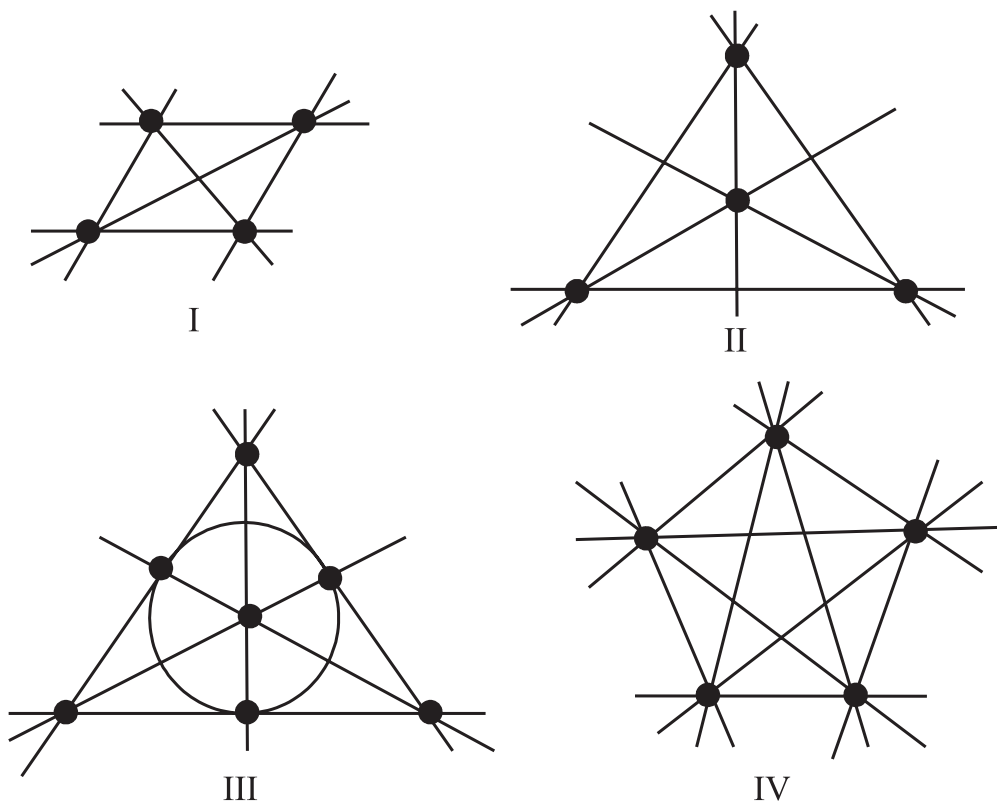


Рис. 5

показана аффинная плоскость порядка 2, имеющая 4 точки и 6 прямых, а на модели III – проективная плоскость порядка 2 с 7 точками и 7 прямыми. На модели IV изображена «плоскость» Π с 5 точками и 10 прямыми, в которой выполняется первая аксиома аффинной плоскости: через любые две различные точки проходит одна-единственная прямая. Более того, никакие три различные точки плоскости Π не лежат на одной прямой. Плоскость Π не является ни аффинной, ни проективной плоскостью, но служит мини-аналогом плоскости Лобачевского: через любую точку, не лежащую на произвольно взятой прямой, проходят ровно две прямые, не пересекающие данную.

3. Мини-топологии

Нас интересуют топологические пространства с небольшим числом элементов. Напомним определения исходных топологических понятий.

Топологическим пространством называется пара $\langle X, \tau \rangle$, где τ – некоторое множество подмножеств множества X , удовлетворяющее следующим аксиомам:

- (1) $\emptyset, X \in \tau$;
- (2) объединение любого непустого семейства множеств из τ принадлежит τ ;
- (3) пересечение любых двух множеств из τ принадлежит τ .

Множество τ выделенных множеств называется *топологией* на X . Элементы пространства X называют его *точками*. Множества $U \in \tau$ называются *открытыми*, а их дополнения $X \setminus U$ – *замкнутыми*. Топология τ является ограниченной дистрибутивной решеткой – подрешеткой булеана $B(X)$ всех подмножеств в X относительно отношения включения \subseteq .

Отображение $f: X \rightarrow Y$ топологических пространств называется *непрерывным*, если прообраз $f^{-1}(U)$ любого открытого множества U в Y есть открытое множество в X . Взаимно однозначное и взаимно непрерывное отображение $f: X \rightarrow Y$ топологических пространств называется их *гомеоморфизмом* (X на Y), а сами пространства – *гомеоморфными* ($X \approx Y$). Отношение гомеоморфности является отношением эквивалентности на классе всех топологических пространств. Это проявление общенаучного понятия изоморфизма в топологии. Гомеоморфные пространства математически «равны», обладают одними и теми же абстрактными топологическими свойствами.

Точку x топологического пространства назовем *открытой* (*замкнутой*, *изолированной*), если одноточечное множество $\{x\}$ открыто (соответственно: замкнуто, открыто-замкнуто). *Замыканием* подмножества A топологического

пространства X называется наименьшее замкнутое множество в X , содержащее A (равное пересечению всех замкнутых множеств пространства X , содержащих A). Открытое множество топологического пространства, содержащее данную точку, называется ее *окрестностью*.

Топологическое пространство $\langle X, \tau \rangle$ называется:

T_0 -*пространством*, если для любых двух его различных точек x и y найдется открытое множество U в X , содержащее ровно одну из них: $x \in U, y \notin U$ или $y \in U, x \notin U$; это равносильно тому, что замыкания различных точек в нем различны;

T_1 -*пространством*, если для любых его различных точек x и y существует открытое множество в X , содержащее x и не содержащее y , что эквивалентно замкнутости всех его точек.

Введем на пространстве $\langle X, \tau \rangle$ бинарное отношение \sim «неотделимости» точек: $x \sim y$ означает, что $\forall U \in \tau (x \in U \Leftrightarrow y \in U)$ для любых точек $x, y \in X$. Легко видеть, что \sim есть отношение эквивалентности на X . Поэтому пространство X разбивается на (непересекающиеся) классы попарно неотделимых точек.

Пусть $\langle X, \tau \rangle$ – конечное топологическое пространство. Любая точка из X имеет наименьшую окрестность. В T_0 -пространстве X наименьшие окрестности различных точек не совпадают. Обозначим через n число элементов в X , а через k – число классов эквивалентности пространства X по отношению \sim . Число k равно числу замыканий одноэлементных подмножеств пространства X .

Изобразим на диаграммах Эйлера-Венна все топологические пространства, имеющие не более четырех элементов (точек). Топологии на n -элементном множестве будем задавать предбазами открытых множеств. Напомним, что *базой* топологического пространства X называется такое множество \mathcal{S} его открытых множеств, что любое открытое множество пространства X является объединением некоторых множеств из \mathcal{S} , а *предбазой* пространства X называется всякое множество его открытых подмножеств, конечные пересечения которых образуют базу в нем (см. рис. 6).

При $n = 1$ получается 1 топологическое пространство, при $n = 2$ имеем 3 пространства, а при $n = 3$ существует 9 попарно негомеоморфных пространств. Пространства 5–9 суть T_0 -пространства, отвечающие трехэлементным упорядоченным множествам (5), (3), (4), (2) и (1) соответственно. Имеются: 1 топологизация одноэлементного пространства, 4 топологизации двухэлементного пространства и 29 топологий на трехэлементном множестве $\{a, b, c\}$: $1 + 3 + 3 + 6 + 3 + 3 + 3 + 6 + 1$.

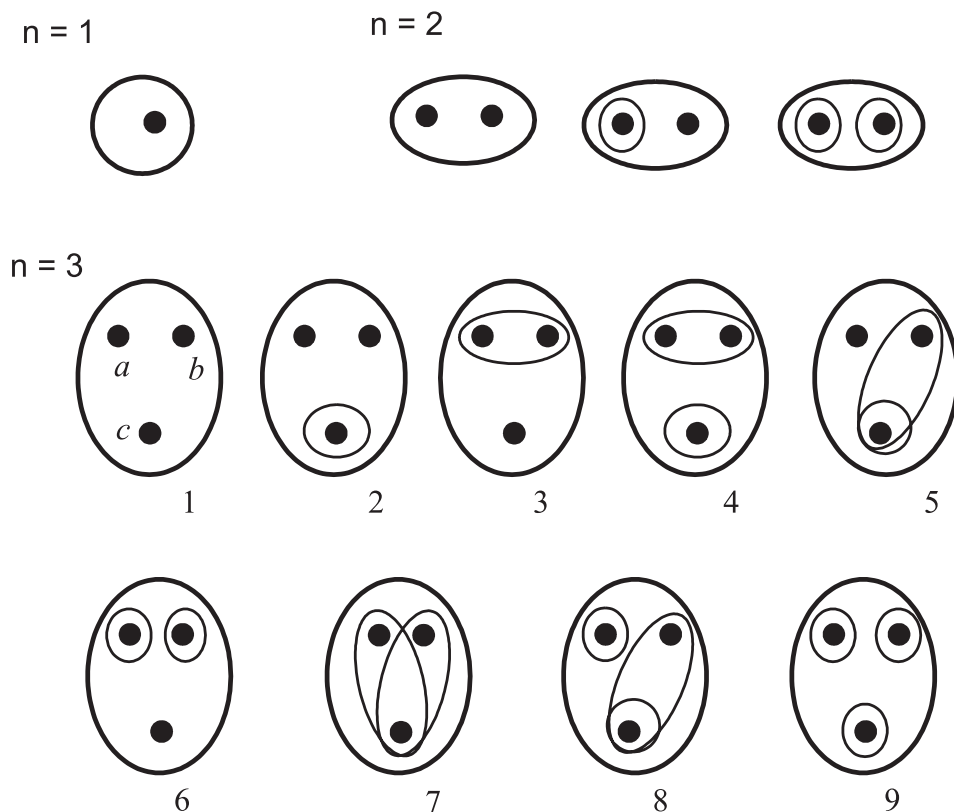


Рис. 6

$k = 1$ или 2

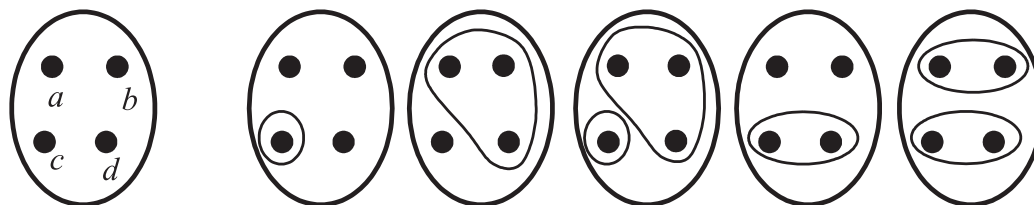


Рис. 7

Последовательно построим все четырехэлементные топологические пространства на множестве $X = \{a, b, c, d\}$. Число k принимает значения 1, 2, 3 или 4 (рис. 7).

Первые 6 пространств получаются совсем легко. При этом число соответствующих топологизаций равно $1 + 4 + 4 + 4 + 6 + 6 = 25$.

В случае $k = 3$ множество X разбивается на $1 + 1 + 2$ точки, и пространства $\langle X, \tau \rangle$ получаются из пяти трехэлементных T_0 -пространств с номерами 5–9 добавлением одной точки к каждому классу соответствующего разбиения; их $11 = 3 + 2 + 2 + 3 + 1$. А число соответствующих топологизаций равно $12 \cdot 3 + (6 + 12) + (12 + 6) + 12 \cdot 3 + 6 = 114$ (см. рис. 8).

Случай $k = 4$ включает все четырехэлементные T_0 -пространства, которые находятся в естественном взаимно однозначном соответствии со всеми четырехэлементными упорядоченными множествами (см. рис. 9).

Получаем 33 топологических пространства с четырьмя элементами. Известно, что существует ровно 219 упорядочений, стало быть, столько же и T_0 -топологизаций четырехэлементного множества. Поэтому всего имеем $25 + 114 + 219 = 358$ топологизаций четырехэлементного множества.

Замечание. Мы излагаем данные вопросы на спецкурсе «Основные математические структуры» для студентов-математиков в 8-м семестре

(см. [3]). На наш взгляд, эта тематика весьма подходит для курсовых и дипломных работ как по математике, так и по методике ее преподавания. По этой теме первый из авторов выступил с получасовым докладом на международной научной конференции в Российском университете дружбы народов 26 марта 2008 г. [4].

Примечания

1. Вечтомов, Е. М. Функциональные представления колец [Текст] : монография / Е. М. Вечтомов. М.: МПГУ, 1993. 191 с.

2. Вечтомов, Е. М. Метафизика математики [Текст] : монография / Е. М. Вечтомов. Киров: ВятГГУ, 2006. 508 с.

3. Вечтомов, Е. М. Основные структуры классической математики: учебное пособие по спецкурсу [Текст] / Е. М. Вечтомов. Киров: ВятГГУ, 2007. 252 с.

4. Вечтомов, Е. М. Обучение математике через простейшие модели [Текст] // Е. М. Вечтомов // Тезисы докладов III Международной конференции «Функциональные пространства. Дифференциальные операторы. Общая топология. Проблемы математического образования», посвященной 85-летию А. Д. Кудрявцева. М.: МФТИ, 2008. С. 597–599.

$k = 3$

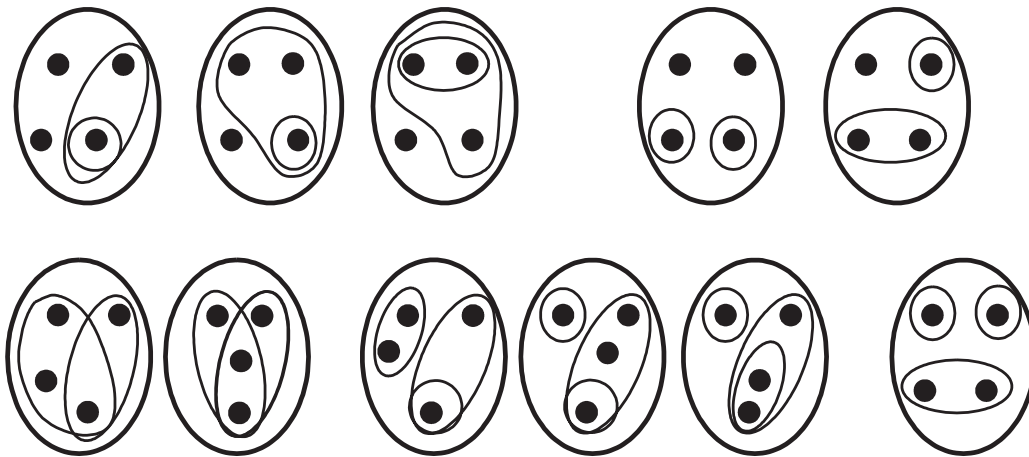


Рис. 8

$k = 4$

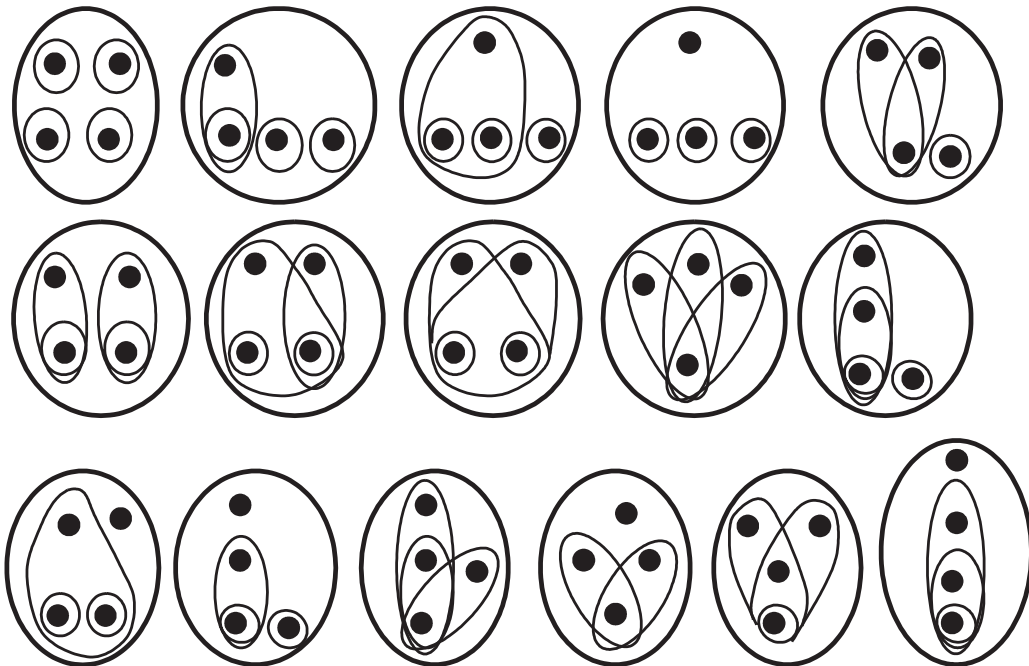


Рис. 9

В. Ю. Иномистов, А. Н. Рапопорт

ПРИМЕНЕНИЕ СПЕЦИАЛИЗИРОВАННЫХ ПАКЕТОВ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ В УЧЕБНОМ ПРОЦЕССЕ

В статье рассмотрены основные идеи и их реализация по применению специализированных пакетов компьютерной математики в учебном процессе.

Современные государственные образовательные стандарты предусматривают сокращение аудиторных часов, отведенных на изучение таких фундаментальных дисциплин, как математика и физика. Вместе с тем в них велика доля рутинного труда, связанного с проведением технически несложных, но объемных вычислительных расчетов. Останавливать решение задачи на этапе построения конечных формул нецелесообразно, так как это не позволит оценить значимость получаемых результатов, доводить же решение до конца не всегда возможно в силу дефицита времени. Кроме того, традиционный подход к решению задач не обеспечивает как в рамках аудиторных занятий, так и в рамках самостоятельной работы студентов, возможности рассмотреть большое число схем решений, подходов и примеров, требуемых для глубокого и всестороннего понимания изучаемых разделов математики и физики. Часто наиболее сложные модели иллюстрируются частными, сильно упрощенными примерами, не позволяющими продемонстрировать всю мощь и красоту того или иного подхода к решению задач, важности теоретических построений.

Широкие возможности для преодоления большинства проблем, присущих традиционному подходу к преподаванию математики, физики, а также ряда специальных дисциплин, предлагают различные математические пакеты. К наиболее известным из них относятся MathCAD, MatLab и Maple. Все они имеют одинаковую библиотеку аналитических вычислений, разработанную компанией Waterloo Maple (создателем Maple), что предопределяет одинаковые возможности всех трех указанных пакетов в области аналитической математики. При этом MathCAD и MatLab содержат традиционно сильные библиотеки для проведения численных расчетов.

В Вятском государственном университете на факультетах прикладной математики и телекоммуникаций и автоматики и вычислительной техники предусмотрено изучение математических пакетов MathCAD и MatLab. Возможности этих

приложений способствуют внедрению компьютерной математики в учебный процесс исходя из соображений разумной достаточности и необходимости. Их возможности в области «чистой» математики уступают возможностям таких пакетов, как Maple и Mathematica, но не следует забывать того, что эти пакеты ориентированы больше на ведение аналитических расчетов. Подготовка инженерных кадров требует большего внимания к овладению навыками построения математических моделей с проведением впоследствии расчетов с привлечением численных методов решения задач. С точки зрения курса математики более предпочтительным оказывается математический пакет Maple, содержащий библиотеку student, которая позволяет не только получить решение какой-либо задачи, например нахождение неопределенного интеграла, но и построить схему пошагового получения конечного результата. Наличие таких преимуществ Maple не может перекрыть возможностей ведения инженерных расчетов, которые предоставляет MatLab.

При использовании в учебном процессе математических пакетов следует тщательно выбирать подходы к решению тех или иных задач. Решение задачи с использованием вычислительной техники не должно сводиться к принципу «черного ящика»: подаем на вход набор числовых параметров, получаем на выходе результат. Следует разумно сочетать использование готовых комплексов лабораторных работ с самостоятельным построением студентами всех этапов решения задачи. При таком подходе MatLab опять оказывается в номинантах на звание лучшего, сочетая в себе возможности работы с готовым комплексом, снабженным кнопками выбора, системой меню и помощи, с возможностью самостоятельного построения алгоритма решения задачи. Это позволяет преподавателю построить методически выдержанную, законченную, снабженную демонстрацией всех возможностей изучаемого подхода или метода лабораторную работу или предложить студентам, используя входящие в MatLab многочисленные функции, самостоятельно разработать последовательность действий, приводящую в конечном итоге к требуемому решению.

Таким образом, изучение математических пакетов на младших курсах приносит неоспоримую пользу, повышает качество преподавания математики и физики при одновременном снижении аудиторной нагрузки. При этом появляется возможность включить в рассмотрение большее число задач, охватывающих как общие, так и частные случаи, не останавливаясь перед большими объемами вычислений. Самостоятельное построение студентами решения поставленных задач с

ИНОМИСТОВ В. Ю. – кандидат технических наук, доцент по кафедре ПМИИ ВятГУ,
РАПОПОРТ А. Н. – доктор физико-математических наук, профессор, зав. кафедрой ПМИИ ВятГУ
© Иномистов В. Ю., Рапопорт А. Н., 2008

использованием возможностей математических пакетов направляет усилия не на рассмотрение частной, локальной проблемы, а активно развивает широкий, творческий подход к достижению поставленной цели.

Еще одним доводом в пользу применения математических пакетов в учебном процессе является наличие в них встроенных средств построения графики, в том числе и трехмерной. Никакие художественные дарования преподавателя не могут ему помочь наглядно изобразить на доске объемную фигуру сложной структуры. Любой же современный математический пакет позволит построить область интегрирования тройного интеграла, а некоторые – еще и поворачивать ее вокруг произвольных осей. Большие возможности перед преподавателем и студентами открывает наличие средств анимации, с помощью которых можно наглядно проследить динамику развития изучаемого процесса. В частности, на факультете ПМТ ВятГУ уже не первый год на лабораторных работах рассматривается задача распространения волн в рояльной струне (полубесконечной и конечной) при ударе молоточком, гитарной струне при щипке.

Не стоит думать, что математические пакеты можно использовать только для поддержки изучения физико-математических дисциплин на младших курсах. Необходимость применения всех возможностей математических пакетов возникает на последующих этапах обучения, начиная с III курса, когда студент должен демонстрировать умение практически применять полученные теоретические знания для решения уже достаточно сложных прикладных задач. При работе над курсовыми проектами нет времени для составления собственных программ на языках высокого уровня для решения систем линейных или нелинейных алгебраических, дифференциальных уравнений, для построения математических моделей исследуемых процессов и т. п. Нужно уметь пользоваться специально разработанными для этих целей процедурами для получения конечного результата.

Более 30 лет тому назад ставилась задача создания инструмента, который помогал бы исследователю самостоятельно решать свои задачи на компьютере без привлечения промежуточного звена в лице программиста. Тогда это вызвало появление языка Фортран и большого числа специализированных библиотек к нему. Тем не менее от специалиста требовалось не только гра-

мотно построить модель, но и написать достаточно сложную программу, реализующую необходимые расчеты. На сегодня быстрое действие даже рядовых персональных компьютеров позволяет использовать математические пакеты для решения сложных задач, представляющих не только образовательную, но и научную ценность. При использовании математического пакета основная тяжесть построения решения приходится на этап создания математической модели, а не написания программы, которая в данном случае зачастую сводится к последовательности вызова стандартных функций. При этом абсолютно отсутствует такая обычная и наиболее трудоемкая составляющая программ на языках высокого уровня, как объявление переменных, приведение типов, захват и освобождение ресурсов, создание функций, реализующих необходимые алгоритмы вычислительной математики.

Таким образом, к достоинствам использования математических пакетов в учебном процессе следует отнести как минимум следующее:

- повышается интенсивность занятий;
- повышается возможность углубленного анализа вариантов задач в процессе занятий, анимации задач, повышается их информативность;
- повышается возможность (при необходимости) разработки собственных программ с использованием авторских алгоритмов для создания математических и имитационных моделей.

В заключение придется добавить несколько ложек дегтя к этой бочке меда:

- возрастают требования к подготовке преподавателя, он должен не только хорошо владеть знаниями в своей области, но знать и активно применять в своей повседневной работе специализированные пакеты компьютерной математики;
- использование математического пакета в учебных целях требует осторожного подхода к подготовке и проведению лабораторных работ, нельзя допустить сползания от получения высоких теоретических знаний и практических навыков к бездумному нажиманию на большую кнопку с надписью «Получить ответ»;
- закупка большого числа лицензий на использование математических пакетов, даже исключительно в учебных целях, является очень дорогим удовольствием для учебного заведения;
- покупка специализированного математического пакета (даже одной лицензии) предприятием является скорее исключением, чем правилом.

С. И. Калинин

О ПРЕДМЕТЕ МАТЕМАТИЧЕСКОГО АНАЛИЗА

В работе осмысливается влияние предмета математического анализа на содержание обучения студентов математических специальностей дифференциальному и интегральному исчислению функций

1. Объект и предмет математики. Метод математического моделирования. «Чистая математика имеет своим объектом пространственные формы и количественные отношения действительного мира, стало быть – весьма реальный материал. Тот факт, что этот материал принимает чрезвычайно абстрактную форму, может лишь слабо затушевать его происхождение из внешнего мира. Но чтобы быть в состоянии исследовать эти формы и отношения в чистом виде, необходимо совершенно отделить их от их содержания, оставить это последнее в стороне как нечто безразличное», – так писал о математике Ф. Энгельс в своей известной работе «Анти-Дюринг» [1, с. 37]. Со времен Энгельса, естественно, математика сильно изменилась, она шагнула в своем развитии далеко вперед. Многие ее разделы стали еще более абстрактными, появились и совершенно новые, расширился круг приложений этой науки. Возникает вопрос – насколько адекватно определение объекта математики, данное Энгельсом, соответствует сегодняшнему представлению о нем?

Ответить на него позволяют следующие рассуждения А. Н. Колмогорова [2, с. 4]: «Абстрактность математики, однако, не означает ее отрыва от материальной действительности. В неразрывной связи с запросами техники и естествознания запас количественных отношений и пространственных форм, изучаемых математикой, непрерывно расширяется, так что данное выше (определение Энгельса. – С. К.) общее определение математики наполняется все более богатым содержанием». А. Н. Колмогоров подчеркивает [2, с. 38], что при «широком понимании терминов «количественные отношения» и «пространственные формы» приведенное Ф. Энгельсом определение математики применимо и на новом современном этапе ее развития». На с. 39 цитируемого источника автор замечает, что пространственные формы можно рассматривать как частный случай количественных отношений, «если этому последнему термину придать достаточно широкое толкование, так что с этой точки зре-

ния включение в определение математики особого упоминания «пространственных форм» является лишь указанием на относительную самостоятельность геометрических отделов математики». Далее он пишет: «Количественные отношения (в общем философском понимании этого термина) характеризуются, в отличие от качественных, лишь своим безразличным отношением к конкретной природе тех предметов, которые они связывают. Поэтому они и могут быть совершенно отделены от их содержания как от чего-то безразличного для дела».

Е. М. Вечтомов в [3, с. 19] объект математики характеризует так: «Объектом математики как науки являются фундаментальные категории формы и количества, взятые в наиболее общем и чистом виде, и всевозможные их проявления». Эта емкая и лаконичная характеристика объекта математики не противоречит взгляду Ф. Энгельса на математику.

Остановимся на предмете математики. Л. Д. Кудрявцев в своей широко известной работе [4, с. 86] обращает внимание на то, что «само понимание предмета математики, а значит, его содержание и расстановка в нем акцентов, вызывает разногласие». Сам автор цитируемой работы, характеризуя предмет математики, исходит из положения: математика изучает математические структуры. Мы условимся придерживаться такой же трактовки понятия предмета математики. По Л. Д. Кудрявцеву, математические структуры – это «абстрактные логические объекты, у которых описан ряд отношений между их элементами».

Приведем взгляд Г. Е. Шилова на понятие математической структуры. На с. 50 своего учебника [5] он пишет: «С некоторой общей точки зрения математика имеет дело только с множествами. Но богатство той или иной математической теории зависит от дальнейших связей между элементами (и подмножествами) множеств, изучаемых в данной теории. Эти связи формируются абстрактным образом с помощью аксиом...» Поэтому Г. Е. Шиллов математической структурой называет множество с наложенными условиями на его элементы и подмножества. Автор признает, что такое определение понятия математической структуры не является вполне точным, ибо оно не содержит общего описания упоминаемых условий на элементы и подмножества данного множества. Но это, в принципе, и не обязательно, поскольку в конкретных ситуациях эти условия можно соответствующим образом описать.

Математическая структура может быть непосредственной математической моделью какого-либо реального явления. Если даже математическая структура не является таковой, то она в той

КАЛИНИН Сергей Иванович – кандидат физико-математических наук, доцент по кафедре прикладной математики ВятГГУ
© Калинин С. И., 2008

или иной степени может служить математическим аппаратом для изучения моделей реальных явлений, поскольку связана с реальным процессом «посредством цепи понятий и логических структур».

Обычно под математической моделью понимают «приближенное описание какого-либо класса явлений внешнего мира, выраженное с помощью математической символики» [6, стб. 574]. Анализ математической модели позволяет постичь суть изучаемого явления или процесса.

В. И. Арнольд математические модели условно подразделяет на «жесткие» и «мягкие» [7]. Жесткие модели реальные явления описывают грубо, крайне приближенно, мягкие же, наоборот, – достаточно точно, адекватно. Автором приведенной работы подчеркивается, что возможности полезности теории мягких моделей открыты сравнительно недавно. Он считает, что основной целью математического образования должно быть воспитание умения исследовать явления реального мира. Искусство составлять и исследовать мягкие математические модели есть составная часть этого умения.

Е. М. Вечтомов [3, с. 61] отмечает, что жесткая математическая модель достаточно примитивна и проста, она зависит лишь от небольшого числа информационных параметров. Мягкая же модель в структурном отношении более сложная, она может зависеть от большего числа параметров, при этом последние могут являться переменными величинами. Такие параметры учитывают изменяющееся состояние изучаемого процесса или явления, обеспечивая обратную связь.

Освещая вопрос классификации математических моделей [3, с. 61–62], Е. М. Вечтомов такие модели подразделяет не только на жесткие и мягкие, но и на грубые и тонкие, одномерные и многомерные, непрерывные и дискретные, простые и сложные и др.

Изучение явления посредством математической модели называется *математическим моделированием*. Сейчас каждому выпускнику вуза, какую бы квалификацию он ни получал, хорошо известно, что математическое моделирование есть эффективный метод познания внешнего мира. В современных условиях он имеет принципиальное значение в вопросах планирования, прогнозирования, управления и т. д.

Воспроизведем общую схему применения метода математического моделирования.

1) Сначала реальная ситуация заменяется ее предметной моделью, т. е. реальное изучаемое явление заменяется приближенным идеализированным образом, выраженным на предметном языке (например, на языке физики или механики);

2) по составленной предметной модели конструируется адекватная математическая модель;

3) в рамках полученной математической модели решается соответствующая математическая задача;

4) найденное решение математической задачи интерпретируется в предметной модели.

Именно по такой схеме решается, например, хорошо известная транспортная задача.

Авторы работы [8], говоря о стиле преподавания математики в школе, схеме применения метода математического моделирования трактуют проще. Они пишут, что «процесс применения математики к решению реальной задачи состоит из трех основных этапов: формулировки математической модели, ее изучения (решения математической задачи) и практических выводов». При обучении школьников математике такого подхода к применению метода математического моделирования достаточно. Важно, чтобы обучение ему учащихся охватывало все перечисленные этапы.

2. О предмете математического анализа. В предисловии к своему учебнику по математическому анализу [9, т. 1, с. 7] А. Д. Кудрявцев пишет, что «математический анализ изучает функциональные зависимости и является той частью классической математики, которая является основой почти для любой математической дисциплины». Приводимую характеристику предмета анализа А. Д. Кудрявцевым можно несколько дополнить и уточнить характеристикой А. Н. Шерстнева. В учебном пособии [10, с. 3] предметом математического анализа автор называет «изучение функций с помощью процессов предельного перехода». Следуя статье С. М. Никольского в [6, стб. 591], предметом математического анализа условимся называть изучение функций и их обобщений методом пределов.

Напомним, дифференциальное и интегральное исчисление функций, применявшееся для решения задач геометрии и механики, было создано трудами математиков XVII–XVIII столетий (И. Ньютон, Г. Лейбниц, Я. и И. Бернулли, Л. Эйлер, Ж. Лагранж, Ж. Д. Аламбер). Ссылаясь на работу [2], подчеркнем, что в отношении публикации исследований по дифференциальному и интегральному исчислению приоритет принадлежит Лейбницу, который дал развернутое изложение основных идей нового исчисления в своих статьях в 1682–1686 гг. По времени же получения основных результатов первенство следует отдать Ньютону, ибо он к основным идеям дифференциального и интегрального исчисления пришел в 1665–1666 гг., в то время как Лейбниц начал свои исследования по анализу бесконечно малых в 1673 г. В обозреваемый период были заложены основы и некоторых других разделов анализа.

В XIX в. дифференциальное и интегральное исчисление, продолжая развиваться с большой

интенсивностью, находя новые направления своего приложения в естествознании и технике, преобразуется в новую дисциплину – классический математический анализ, фундаментом, основанием которого стало понятие системы действительных чисел.

Нужно заметить, что классический анализ возник не только в силу новых запросов естествознания и техники. С самого начала XIX в. ведущих математиков привлекают вопросы строгого обоснования анализа. В частности, О. Л. Коши выпускает следующие свои работы: «Курс анализа» (1821 г.) «Резюме лекций по исчислению бесконечно малых» (1823 г.), «Лекции по приложениям анализа к геометрии» (1826–1828 гг.) [21], которые были основаны на систематическом применении понятия предела. Анализ стал располагать строгим изложением теории пределов, теории рядов, дифференциального и интегрального исчисления и пр. В 1834 г. Н. И. Лобачевский и в 1837 г. П. Дирихле четко сформулировали определение функции как совершенно произвольного соответствия [2, с. 46].

Таким образом, в XIX в. предмет анализа значительно расширился, при этом существенно выросла степень строгого обоснования его теорий. Последнему обстоятельству, безусловно, способствовало создание в 1872 г. теории иррациональных чисел (Р. Дедекинд, Г. Кантор, К. Вейерштрасс) [2].

Учитывая то, что развитие анализа предопределило возможности изучать его методами более сложных образований, чем функция [6, стб. 591] (напр., посредством операторов, функционалов и т. д.), то термин «математический анализ» можно понимать весьма широко. Анализ охватывает большую часть математики, помимо дифференциального и интегрального исчисления к нему можно отнести теорию функций действительного переменного, теорию функций комплексного переменного, теорию приближений, теорию обыкновенных дифференциальных уравнений и уравнений с частными производными, теорию интегральных уравнений, дифференциальную геометрию, вариационное исчисление, функциональный анализ и некоторые другие математические дисциплины.

Именно такую широкую трактовку анализа несет учебное пособие в трех частях Г. Е. Шилова [5], [11]. Автор этого пособия поясняет, в частности, принцип введения основных структур в математическом анализе [11, с. 9]. Воспроизведем его.

Напомним, математический анализ имеет дело с числами, функциями и действиями над ними. Связи между этими объектами описываются теорией множеств, поскольку числа, функции образуют различные множества, а операции объеди-

нения, пересечения и дополнения вместе с отношением включения позволяют описать некоторые общие свойства таких множеств. При наложении на множества дополнительных условий, формулируемых в виде определенных систем аксиом, и получают основные структуры математического анализа. Упоминаемые системы аксиом должны соответствовать свойствам или операциям, которые применяются в классическом анализе. Например, линейные пространства получают посредством аксиоматизирования линейных операций – сложения элементов и умножения их на числа, а метрические – с помощью введения понятия расстояния между элементами и аксиоматизирования операции предельного перехода. Если брать такую структуру как линейное нормированное пространство, то в нем определяются и линейные операции над элементами, и предельный переход. При добавлении к операциям, присущим линейному нормированному пространству, операции умножения элементов друг на друга получаем нормированную алгебру. Отметим также, что при аксиоматизировании понятия скалярного произведения получается структура, именуемая предгильбертовым пространством. И т. д...

Мы сознательно опускаем описание и упоминание ряда промежуточных структур. Многие из приводимых структур уже изучаются не в рамках классического анализа.

Подводя итог описанию предмета анализа, подчеркнем, что термин «математический анализ» все же чаще употребляется при наименовании лишь основ математического анализа, которые объединяют в себе теорию действительных чисел, теорию пределов, рядов, дифференциальное и интегральное исчисление и их непосредственные приложения (теория максимумов и минимумов, теория неявных функций, ряды и интегралы Фурье).

3. Влияние предмета математического анализа на содержание обучения студентов математических специальностей дифференциальному и интегральному исчислению функций. Так как математический анализ как часть математики изучает соответствующие математические структуры, то в первую очередь студенты должны изучать те из них, которые непосредственно моделируют реальные процессы и явления окружающего нас мира. Приведем несколько примеров моделей, относящихся к дифференциальному и интегральному исчислению функций, которые иллюстрируют это положение.

1) Само понятие производной функции моделирует скорость движущейся материальной точки (движущегося тела), а понятие второй производной – ее ускорение.

2) Понятие определенного интеграла Римана есть модель работы переменной силы при пря-

молинейном перемещении точки под действием этой силы на фиксированном участке пути.

3) Дифференциал функции одной переменной предстает моделью равномерного движения.

4) Производная вектор-функции также моделирует скорость механического движения, только движения в пространстве.

5) Производная по направлению может быть рассмотрена как модель скорости изменения скалярного поля в заданном направлении. В этом случае градиент будет описывать направление наибольшей скорости изменения такого поля.

6) Двойной интеграл является моделью массы неоднородной пластинки, а тройной интеграл – массы неоднородного тела.

7) Дифференциальное соотношение (уравнение) $y' = ky$ есть модель описания демографического процесса. Здесь $y = y(t)$ есть функция, характеризующая число жителей региона в момент времени t , k – соответствующий коэффициент пропорциональности.

8) Производная Каратеодори моделирует описание изменения положения секущей плоской кривой, когда секущая стремится занять положение касательной.

Многие из моделей приведенного перечня рассматриваются, в частности, в школьном курсе математики (в началах математического анализа), следовательно, их должны знать особенно хорошо будущие учителя математики.

При рассмотрении соответствующих структур дифференциального и интегрального исчисления студентам-математикам важно уяснить то, что иногда одни и те же структуры способны моделировать совершенно разные реальные явления. Так, например, производная может моделировать не только скорость, но и угловой коэффициент касательной к плоской кривой в заданной точке, линейную плотность в точке неоднородного стержня, силу тока в данный момент времени и т. д. Аналогично дифференциал может быть рассмотрен не только моделью равномерного движения, но и моделью массы однородного стержня малой длины или моделью количества электричества, протекающего через поперечное сечение проводника за малый промежуток времени.

В известной терминологии М. М. Постникова приведенные примеры есть примеры схем моделей. У него подход к толкованию предмета математики несколько отличается от подхода Л. Д. Кудрявцева, описанного выше. По мнению М. М. Постникова, математика есть наука о схемах моделей окружающего нас мира. Г. И. Саранцев в [12, с. 57] отмечает, что примером схем моделей служат математические понятия. Иллюстрации отмеченному мы уже видели. Приведем еще один пример: обыкновенное линейное однородное дифференциальное уравнение второго

порядка с постоянными коэффициентами $y'' + ay = 0$ есть схема всех моделей колебательного движения, в какой бы соответствующей конкретной ситуации они ни возникали.

Для теории дифференциального и интегрального исчисления в отношении рассматриваемых примеров совершенно не важно, какое явление моделируется той или иной структурой, ибо эта теория изучает свойства структур, количественные и качественные связи между элементами структур, а также связи между самими структурами. К тому же (это мы отмечали ранее), в рамках математики, в частности в рамках дифференциального и интегрального исчисления, имеются такие структуры, которые непосредственно не выступают моделями реальных явлений, но они являются удобным аппаратом изучения таковых. При изучении студентами обозреваемого раздела анализа преподавателем должен делаться акцент на разъяснение истинного смысла математического понятия или какой-то структуры, что будет способствовать постижению сущности математической науки.

На содержание математической подготовки студентов-математиков по дифференциальному и интегральному исчислению функций непрерывное воздействие должны оказывать, безусловно, динамичное расширение предмета математического анализа и тенденции развития ряда его направлений в сочетании с определенным консерватизмом, связанным с продолжением российских традиций обучения студентов анализу в высшей школе.

При изложении курса математического анализа мы считаем целесообразным более обстоятельно знакомить студентов с неравенствами, которые играют весьма и весьма большую роль в вопросах приложений дифференциального и интегрального исчисления. В большинстве учебных пособий по анализу неравенствам уделяется крайне мало места. Например, исследования по уточнениям классических неравенств, активно проводимые в последние годы, не отражаются практически ни в одном отечественном учебном пособии по математическому анализу.

Кроме того, на наш взгляд, студентов при изучении дифференциального исчисления функций полезно более обстоятельно знакомить со свойствами выпуклых функций, также имеющих значительные применения.

При изучении основ анализа студентам полезно иметь дело не только с классическими утверждениями и фактами, но и развитием последних, получаемым в рамках исследований в соответствующих направлениях.

Представленная выше трактовка предмета анализа обуславливает в определенной полноте отражение в новых учебных пособиях современного состояния фундаментальных исследований по рассматриваемому разделу математики. В но-

вых учебниках по дифференциальному и интегральному исчислению должны быть отражены и определенные виды деятельности, в частности, эвристическая составляющая математической деятельности. В вопросе приобщения студентов при обучении математике к творчеству, к научным исследованиям перечисленное является очень важным и актуальным.

Характеризуя влияние предмета математического анализа на содержание обучения дифференциальному и интегральному исчислению функций студентов-математиков, мы должны отметить, что в это содержание помимо предметных знаний должны также включаться и действия, адекватные понятиям и утверждениям выделенного раздела анализа. Скажем, при рассмотрении понятия производной к действиям, адекватным этому понятию, можно отнести:

1) рассмотрение различных определений понятия производной функции одной переменной в точке [13], [14] – определений по Коши, по Каратеодори, определений через условие дифференцируемости функции, через подходящую линейную функцию, определений двусторонней производной [15], [16], симметрической производной, Π -производной [15], [16], l -производной [17], верхней и нижней производных Дини для негладкой функции [18];

2) сопоставление вышеупомянутых определений производной с определениями производной в смысле Фреше и Гато [19];

3) выведение следствий из факта существования той или иной производной функции (например, наличие производной функции в точке по Коши или по Каратеодори влечет ее непрерывность и дифференцируемость в этой точке);

4) выявление соотношений между объемами и содержаниями различных понятий производной (скажем, понятие l -производной обобщает понятие производной по Коши, поскольку объем первого включает в себя объем второго: если $l(x) = x$, то l -производная функции есть обычная производная; подробное сравнение понятий производных функции нами проведено в работе [14]);

5) описание поведения функции, если известно аналитическое выражение ее производной или задан график производной (заметим, соответствующие задания об исследовании функции на монотонность и экстремумы по графику производной часто предлагаются в вариантах ЕГЭ по математике в средней школе);

6) вычисление производной функции в точке по определению (в смысле какого-либо конкретного определения);

7) построение графика производной;

8) установление связи между существованием производной функции в точке и ее дифференцируемостью в ней;

9) геометрическое интерпретирование производных;

10) описание различных моделей реальных явлений с помощью производной в механике, физике, экономике (см., напр., [20]), химии, экологии и т. д.;

11) осмысление построения дифференциального исчисления функций в терминах соответствующей производной (скажем, l -производной).

Перечень такого рода действий можно продолжить. Указанные действия и другие включаются в содержание обучения студентов дифференциальному и интегральному исчислению функций через специально составляемые упражнения (см., напр., [21]). В таких упражнениях ведущая роль должна отводиться теоретическим задачам.

Аналогично иллюстрируются действия, адекватные какой-либо теореме дифференциального исчисления, скажем теореме Лагранжа. Можно осмыслить следующие задачи, соответствующие этой теореме:

1) применить формулу Лагранжа $f(b) - f(a) = f'(\xi)(b - a)$ к конкретным функциям;

2) дать геометрическую интерпретацию формулы Лагранжа;

3) установить факт симметричности формулы Лагранжа относительно точек a и b ;

4) обосновать формулу Лагранжа конечных приращений

$$\Delta f(x_0) = f'(x_0 + \theta \Delta x) \Delta x, 0 < \theta < 1;$$

5) рассмотреть возможности обобщения формулы Лагранжа;

6) используя метод доказательства теоремы Лагранжа, привести какое-либо развитие этой теоремы.

Таким образом, предмет математического анализа действительно оказывает значительное влияние на содержание обучения студентов математических специальностей дифференциальному и интегральному исчислению функций.

Примечания

1. *Маркс, К.* Соч. [Текст] / К. Маркс, Ф. Энгельс. Т. 20. Изд-е 2-е. М., 1961. 827 с.

2. *Колмогоров, А. Н.* Математика. Исторический очерк [Текст] / А. Н. Колмогоров. М.: Анабасис, 2006. 60 с.

3. *Вечтомов, Е. М.* Метафизика математики [Текст]: монография / Е. М. Вечтомов. Киров: Изд-во ВятГГУ, 2006. 508 с.

4. *Кудрявцев, Л. Д.* Современная математика и ее преподавание [Текст] / учеб. пособие для вузов / Л. Д. Кудрявцев. М.: Наука, 1985. 170 с.

5. *Шилов, Г. Е.* Математический анализ (функции одного переменного) [Текст] / Г. Е. Шилов. Ч. 1–2. М.: Наука, 1969. 528 с.

6. Математическая энциклопедия [Текст] / гл. ред. И. М. Виноградов. Т. 3. Коо – Од. М.: Сов. энцикл., 1982. 1184 стб.

7. Арнольд, В. И. «Жесткие» и «мягкие» математические модели [Текст] / В. И. Арнольд. М.: МЦНМО, 2000. 32 с.

8. Виленкин, Н. Я. Научно-техническая революция и школьный курс математики [Текст] / Н. Я. Виленкин, А. Д. Мышкис // Математика в школе. 1987. № 3. С. 40–43.

9. Кудрявцев, Л. Д. Математический анализ [Текст] / Л. Д. Кудрявцев. Т. I–II. М.: Высш. шк., 1970. 588 с., 420 с.

10. Шерстнев, А. В. Конспект лекций по математическому анализу [Текст]: учеб. пособие / А. В. Шерстнев. Казань: Изд-во Казан. ун-та, 1993. 301 с.

11. Шилов, Г. Е. Математический анализ (функции одного переменного) [Текст] / Г. Е. Шилов. Ч. 3. М.: Наука, 1970. 352 с.

12. Саранцев, Г. И. Методология методики обучения математике [Текст] / Г. И. Саранцев. Саранск, 2001. 141 с.

13. Калинин, С. И. Определения производной функции в контексте логических характеристик понятий в математике [Текст] / С. И. Калинин // Проблемы историко-научных исследований в математике и математическом образовании: м-лы Междунар. науч. конф. (г. Пермь, 7–9 сентября 2007 г.) / отв. ред. А. Е. Малых; Перм. гос. пед. ун-т. Пермь, 2007. С. 71–72.

14. Калинин, С. И. Об определениях понятия производной функции [Текст] / С. И. Калинин // Математический вестник педвузов и университетов Волго-Вятского региона. Вып. 9: Периодич. межвуз. сб. науч.-метод. работ. Киров: Изд-во ВятГГУ, 2007. С. 104–116.

15. Попов, В. А. Элементарная математика и начала анализа [Текст]: методические статьи и задачи / В. А. Попов. Сыктывкар: Коми гос. пед. ин-т, 2002. 300 с.

16. Попов, В. А. Новые основы дифференциального исчисления [Текст]: учеб. пособие для спецкурсов / В. А. Попов. Сыктывкар: «ПОЛИГРАФ-СЕРВИС», 2002. 64 с.

17. Брайчев, Г. Г. Об одном обобщении понятия производной и его применения в математическом анализе [Текст] / Г. Г. Брайчев, А. А. Меньшикова // Научные труды математического факультета Моск. пед. гос. ун-та: Юбилейный сб. 100 лет. М.: Прометей, 2000. С. 27–30.

18. Демьянов, В. Ф. Обобщение понятия производной в негладком анализе [Текст] / В. Ф. Демьянов // Соросовский образовательный журнал. 1996. № 5. С. 121–127.

19. Колмогоров, А. Н. Элементы теории функций и функционального анализа [Текст] / А. Н. Колмогоров, С. В. Фомин. М.: Наука, 1972. 496 с.

20. Калинин, С. И. Об экономической интерпретации производной [Текст] / С. И. Калинин // Актуальные проблемы гуманитарного и экономического образования в негосударственных вузах: тез. докл. Первой науч.-практ. конф. Ч. 2. Киров: Киров. филиал Моск. гуманит.-экон. ин-та, 1998. С. 8–11.

21. Калинин, С. И. Задачи и упражнения по началам математического анализа [Текст]: пособие для учащихся школ и классов с углубленным изучением математики и для внекл. занятий математикой / С. И. Калинин, Е. С. Канин, Г. М. Маянская, Л. В. Ончукова, И. И. Подгорная, С. А. Фалелеева. М.: Московский Лицей, 2001. 208 с.

М. В. Клековкина, А. Г. Яшина

ЧЕТЫРЕ МЕТОДА ДОКАЗАТЕЛЬСТВА ОДНОГО НЕРАВЕНСТВА

Работа рассматривает разные подходы к доказательству одного известного числового неравенства.

0. Введение. Вниманию читателей представляется следующее неравенство

$$\frac{1}{4} < \arcsin 0,8 - \arcsin 0,6 < \frac{1}{3} \quad (1)$$

Ранее оно было установлено в работах [1] и [2].

В [1] данное неравенство доказывалось посредством применения классической теоремы Лагранжа, а в [2] – с помощью свойств определенного интеграла Римана и его геометрической интерпретации.

В настоящей заметке будут воспроизведены упоминаемые методы и дополнительно приведены иные подходы, которые опираются на соответствующие сведения из теории числовых и функциональных рядов, а также элементарные геометрические рассуждения.

Рассмотрим по порядку анонсируемые выше методы доказательства неравенства (1).

1. Доказательство посредством теоремы Лагранжа [1]. Напомним формулировку теоремы Лагранжа: Пусть функция $f(x)$ определена и непрерывна на отрезке $[a; b]$ и дифференцируема внутри него. Тогда выполняется равенство

$$\frac{f(b) - f(a)}{b - a} = f'(c), \text{ где } c \in (a; b).$$

Применим теорему Лагранжа к функции $f(x) = \arcsin(x)$, рассматриваемой на отрезке $[0,6; 0,8]$. Очевидно, она удовлетворяет всем усло-

виям этой теоремы. Так как $f'(x) = \frac{1}{\sqrt{1-x^2}}$, то

$$\frac{f(0,8) - f(0,6)}{0,8 - 0,6} = f'(c), \text{ где } 0,6 < c < 0,8. \text{ Имеем:}$$

$$\frac{\arcsin 0,8 - \arcsin 0,6}{0,2} = \frac{1}{\sqrt{1-c^2}}. \text{ Число } \frac{1}{\sqrt{1-c^2}}$$

оценим сверху и снизу: так как $0,6 < c < 0,8$, то $0,36 < 1 - c^2 < 0,64$ и $0,6 < \sqrt{1-c^2} < 0,8$, следо-

вательно, $\frac{5}{4} < \frac{1}{\sqrt{1-c^2}} < \frac{5}{3}$. Таким образом, имеем

КЛЕКОВКИНА Мария Викторовна – студентка III курса факультета информатики ВятГГУ
ЯШИНА Александра Геннадьевна – студентка III курса факультета информатики ВятГГУ
 © Клековкина М. В., Яшина А. Г., 2008

оценку $\frac{5}{4} < \frac{\arcsin 0,8 - \arcsin 0,6}{0,2} < \frac{5}{3}$, и оконча-

тельно получаем $\frac{1}{4} < \arcsin 0,8 - \arcsin 0,6 < \frac{1}{3}$. Неравенство (1) доказано.

2. Доказательство методом интеграла Римана [2]. Согласно геометрическому смыслу определённого интеграла значение $\arcsin 0,8 - \arcsin 0,6$ равно S – площади криволинейной трапеции АСFB (см. рис. 1), ограниченной сверху графиком функции

$f(x) = \frac{1}{\sqrt{1-x^2}}$. То есть:

$S = \arcsin \Big|_{0,6}^{0,8} = \int_{0,6}^{0,8} \frac{dx}{\sqrt{1-x^2}}$. Найдём площади двух прямоугольников АЕFB и АСDB.

$$S_1 = S_{ABCD} = (0,8 - 0,6) \cdot \frac{1}{\sqrt{1-0,6^2}} = \frac{2}{10} \cdot \frac{5}{4} = \frac{1}{4}.$$

$$S_2 = S_{ABFE} = (0,8 - 0,6) \cdot \frac{1}{\sqrt{1-0,8^2}} = \frac{2}{10} \cdot \frac{5}{3} = \frac{1}{3}.$$

Таким образом, получаем соотношение площадей $S_2 < S < S_1$. Неравенство (1) доказано.

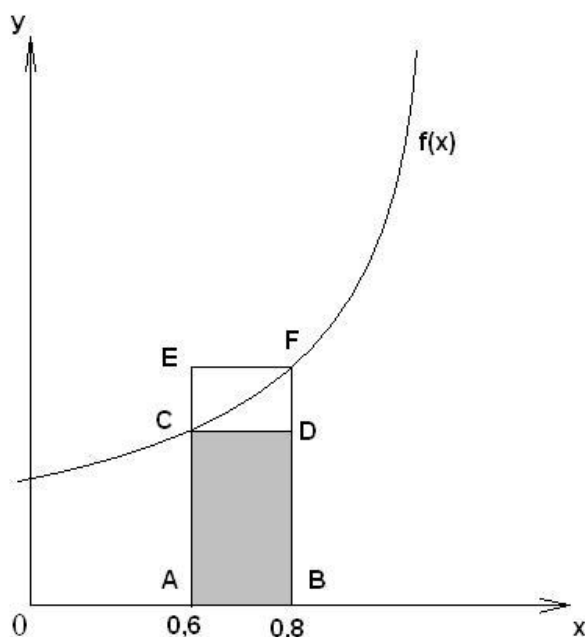


Рис. 1

Перейдём к рассмотрению доказательств, полученных нами.

3. Доказательство посредством рядов

I способ. Найдём синусы всех частей неравенства (1). Имеем:

$$\sin(\arcsin 0,8 - \arcsin 0,6) = \sin(\arcsin \frac{4}{5}) \cdot \cos(\arcsin \frac{3}{5}) -$$

$$- \sin(\arcsin \frac{3}{5}) \cdot \cos(\arcsin \frac{4}{5}) =$$

$$= \frac{4}{5} \cdot \cos(\arcsin \frac{3}{5}) - \frac{3}{5} \cdot \cos(\arcsin \frac{4}{5}).$$

$$\cos(\arcsin \frac{3}{5}) = \sqrt{1 - \sin^2(\arcsin \frac{3}{5})} = \sqrt{1 - \frac{9}{25}} = \frac{4}{5}.$$

$$\cos(\arcsin \frac{4}{5}) = \sqrt{1 - \sin^2(\arcsin \frac{4}{5})} = \sqrt{1 - \frac{16}{25}} = \frac{3}{5}.$$

Таким образом, получаем:

$$\sin(\arcsin 0,8 - \arcsin 0,6) = \frac{4}{5} \cdot \frac{4}{5} - \frac{3}{5} \cdot \frac{3}{5} = \frac{7}{25}.$$

Для вычисления $\sin \frac{1}{4}$ и $\sin \frac{1}{3}$ воспользуемся формулой разложения функции $\sin x$ в ряд Маклорена:

$$\sin x = \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{x^{2n-1}}{(2n-1)!} \quad (2)$$

Положим в (2) $x = \frac{1}{4}$ и найдём значение $\sin \frac{1}{4}$

с точностью до 0,001 (точность возьмём «с запасом», так как это не усложнит расчёты). Стоит заметить, что значения углов, синусы которых мы находим, даны в радианах, а не градусах, поэтому формулу разложения можно применить. Имеем:

$$\begin{aligned} \sin \frac{1}{4} &= \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{1}{4^{2n-1} \cdot (2n-1)!} = \frac{1}{4} - \frac{1}{4^3 \cdot 3!} + \\ &+ \frac{1}{4^5 \cdot 5!} - \frac{1}{4^7 \cdot 7!} + \dots = \frac{1}{4} - \frac{1}{6 \cdot 64} + \frac{1}{120 \cdot 1024} - \dots \end{aligned}$$

Пользуясь хорошо известным следствием теоремы Лейбница о знакопередающихся рядах,

оценим: $|\frac{1}{120 \cdot 1024}| < 0,001$. Тогда можно опустить члены, начиная с четвёртого. Теперь рассмотрим (2) при $x = \frac{1}{3}$. Имеем:

$$\begin{aligned} \sin \frac{1}{3} &= \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{1}{3^{2n-1} \cdot (2n-1)!} = \\ &= \frac{1}{3} - \frac{1}{3^3 \cdot 3!} + \frac{1}{3^5 \cdot 5!} - \frac{1}{3^7 \cdot 7!} + \dots = \\ &= \frac{1}{3} - \frac{1}{27 \cdot 6} + \frac{1}{243 \cdot 120} - \dots \end{aligned}$$

Очевидно, что $|\frac{1}{243 \cdot 120}| < 0,001$, следовательно, для точности 0,001 достаточно подсчитать сумму первых трёх членов. Таким образом, $\sin \frac{1}{4} \cong \frac{30401}{122880}$ и $\sin \frac{1}{3} \cong \frac{9541}{29160}$. Нетрудно проверить, что $\frac{30401}{122880} < \frac{7}{25} < \frac{9541}{29160}$. Поскольку функция $\sin(x)$ на отрезке $[0, \frac{\pi}{2}]$ возрастает, и значения $\frac{1}{4}$, $\arcsin 0,8 - \arcsin 0,6$, $\frac{1}{3}$ входят в рассматриваемый промежуток, то неравенство (1) выполняется.

II способ. Он будет аналогичен предыдущему способу. На этот раз сравним значения косинусов всех частей неравенства (1).

$$\begin{aligned} \cos(\arcsin 0,8 - \arcsin 0,6) &= \\ &= \cos(\arcsin 0,8) \cdot \cos(\arcsin 0,6) + \\ &+ \sin(\arcsin 0,8) \cdot \sin(\arcsin 0,6) = \\ &= \cos(\arcsin 0,8) \cdot \cos(\arcsin 0,6) + \frac{4}{5} \cdot \frac{3}{5}. \\ \cos(\arcsin 0,8) &= \sqrt{1 - \sin^2(\arcsin 0,8)} = \sqrt{1 - \frac{16}{25}} = \frac{3}{5} \\ \cos(\arcsin 0,6) &= \sqrt{1 - \sin^2(\arcsin 0,6)} = \sqrt{1 - \frac{9}{25}} = \frac{4}{5}. \\ \cos(\arcsin 0,8 - \arcsin 0,6) &= \frac{3}{5} \cdot \frac{4}{5} + \frac{4}{5} \cdot \frac{3}{5} = \frac{24}{25}. \end{aligned}$$

Напомним формулу разложения в ряд Маклорена для функции $\cos x$:

$$\cos x = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n}}{(2n)!} \quad (3)$$

Найдём значение (3) при $x = \frac{1}{3}$ с точностью до 0,001.

$$\begin{aligned} \cos \frac{1}{3} &= \sum_{n=0}^{\infty} (-1)^n \cdot \frac{1}{3^{2n} \cdot (2n)!} = \\ &= 1 - \frac{1}{3^2 \cdot 2!} + \frac{1}{3^4 \cdot 4!} - \frac{1}{3^6 \cdot 6!} + \dots \end{aligned}$$

Очевидно, что $|\frac{1}{729 \cdot 720}| < 0,001$, то есть для обозначенной точности нам достаточно взять первые четыре члена.

Теперь пусть $x = \frac{1}{4}$, тогда:

$$\begin{aligned} \cos \frac{1}{4} &= \sum_{n=0}^{\infty} (-1)^n \cdot \frac{1}{4^{2n} \cdot (2n)!} = \\ &= 1 - \frac{1}{4^2 \cdot 2!} + \frac{1}{4^4 \cdot 4!} - \frac{1}{4^6 \cdot 6!} + \dots \end{aligned}$$

Учитывая оценку $|\frac{1}{4096 \cdot 720}| < 0,001$, можно опустить члены ряда, начиная с пятого. В итоге имеем: $\cos \frac{1}{3} \cong \frac{495989}{524880}$ и $\cos \frac{1}{4} \cong \frac{2857439}{2949120}$, то есть $\frac{2857439}{2949120} > \frac{24}{25} > \frac{495989}{524880}$. Учитывая, что

функция $\cos x$ на отрезке $[0, \frac{\pi}{2}]$ убывает, можно сделать вывод, что неравенство (1) доказано.

4. Геометрический метод

Представим на тригонометрическом круге два вектора \vec{OA} и \vec{OB} , такие, что углы между осью Ox и ними были равны $\arcsin 0,8$ и $\arcsin 0,6$ соответственно (рис. 2). Скалярное произведение этих векторов можно записать так: $\vec{OA} \cdot \vec{OB} = |\vec{OA}| \cdot |\vec{OB}| \cdot \cos \gamma$, где угол $\gamma = \arcsin 0,8 - \arcsin 0,6$. В прямоугольной декартовой системе координат векторы \vec{OA} и \vec{OB} имеют координаты $(0,8; 0,6)$ и $(0,6; 0,8)$ соответственно. Используя координатную формулу скалярного произведения, получаем $\vec{OA} \cdot \vec{OB} = 0,8 \cdot 0,6 + 0,6 \cdot 0,8$. В итоге имеем равенство: $\cos \gamma = 0,6 \cdot 0,8 + 0,8 \cdot 0,6 = 0,96$. Заметим, что $\sin \gamma = 0,28$.

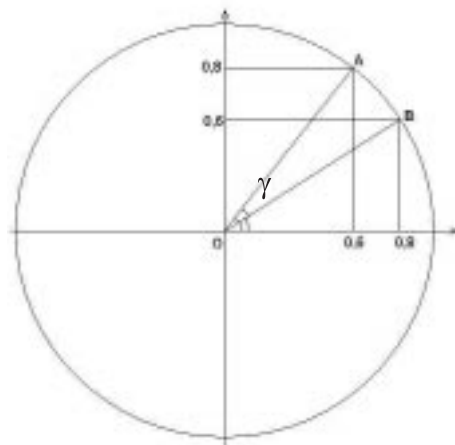


Рис. 2

Угол γ равен длине дуги PQ (рис. 3). Оценим величину угла γ снизу. Очевидно, что длина дуги PQ больше, чем длина хорды, стягивающей её концы. Рассмотрим PNQ , где $PN = \sin\gamma$ и $NQ = 1 - \cos\gamma$. По теореме Пифагора имеем:

$$\begin{aligned} PQ &= \sqrt{\sin^2\gamma + (1 - \cos\gamma)^2} = \\ &= \sqrt{0,28^2 + (1 - 0,96)^2} = \\ &= \sqrt{0,0784 + 0,0016} = \sqrt{0,08} = \frac{\sqrt{2}}{5}. \end{aligned}$$

Следовательно, $\gamma > \frac{\sqrt{2}}{5} > \frac{1}{4}$.

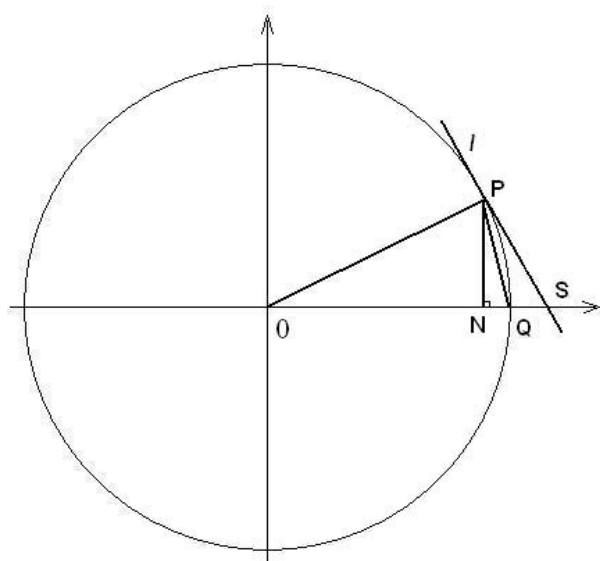


Рис. 3

Теперь оценим величину γ сверху. Прямая l – касательная к окружности в точке $P(0,96;0,28)$. Длина дуги PQ меньше, чем длина отрезка PS . Найдём его длину. Касательная l задаётся урав-

нением $y = y_0 + \frac{x_0}{\sqrt{1-x_0^2}} \cdot (x - x_0)$. Подставляя $x_0 = 0,96$ и $y_0 = 0,28$, получаем уравнение прямой l : $0,07y = 0,24x - 0,2108$. Точка S – это точка пересечения прямой l с осью Ox (рис. 3). Из

уравнения прямой l находим координаты точки $S - \frac{0,0527}{0,06}$ и 0 . Теперь выражаем расстояние от P до S :

$$\begin{aligned} PS &= \sqrt{\left(\frac{0,05527}{0,06} - 0,96\right)^2 + (0,28)^2} = \\ &= \sqrt{\left(\frac{0,0527 - 0,96 \cdot 0,06}{0,06}\right)^2 - 0,0784} = \\ &= \sqrt{\left(\frac{-0,0049}{0,06}\right)^2 + 0,0784} = \\ &= \sqrt{\frac{0,00002401 + 0,00028224}{0,0036}} = \\ &= \sqrt{\frac{0,00030625}{0,036}} = \frac{0,0175}{0,06} = \frac{7}{24}. \end{aligned}$$

Тогда получаем: $\gamma < \frac{7}{24} < \frac{1}{3}$.

Таким образом, полученные оценки для угла γ доказывают неравенство (1).

В заключение отметим, что представленный в настоящей статье материал полезен учителям математики при проведении внеклассных занятий по предмету и при подготовке ими элективных и факультативных курсов для учащихся. Кроме того, затронутые вопросы могут быть взяты школьниками в качестве дополнительных подходов к решению задач по установлению неравенств. Специалистам-математикам предложенные методы способны помочь при оценке соответствующих величин, возникающих в результате расчётов.

Примечания

1. *Островерхая, А. Д.* Применение теоремы Лагранжа и её следствий при решении задач [Текст] / А. Д. Островерхая // Математика в школе. 2001. № 9. С. 49–53.
2. *Вороной, А. Н.* Интеграл помогает доказывать неравенства [Текст] / А. Н. Вороной // Математика в школе. 2002. № 6. С. 66–70.

В. А. Онегов

**ПРИВЕДЕНИЕ ИСХОДНОЙ
ЛИНЕЙНОЙ СИСТЕМЫ К ВИДУ,
УДОБНОМУ ДЛЯ ПРИМЕНЕНИЯ МЕТОДА
ИТЕРАЦИЙ С ГАРАНТИРОВАННОЙ
СХОДИМОСТЬЮ**

В приложениях, особенно при решении уравнений математической физики методом сеток, метод итераций решения возникающих систем уравнений является предпочтительным. В то же время вид системы не всегда позволяет очевидным способом привести систему к виду, удобному для применения метода итераций, обеспечивающему сходимость итерационного процесса. Однако линейная алгебра дает основные принципы решения поставленной задачи. Кажущаяся простота метода итераций иногда приводит к поверхностному представлению о тривиальности решения возникающих вопросов. Это, конечно, не так. В работе приведены два приема решения сформулированной задачи с теоретическим обоснованием, практически примерами и комментариями.

Часто в приложениях возникает ситуация, в которой систему линейных алгебраических уравнений целесообразно решать методом итераций. Например, системы, возникающие при применении метода сеток для уравнений математической физики, имеют именно такой характер.

Однако в стандартном курсе «Численные методы» вопрос о приведении исходной системы к виду, удобному для применения метода итераций и гарантирующему сходимость, освещен недостаточно. Исследователь, встретившейся с подобной ситуацией, бывает не готов к ее решению. Наша статья имеет целью восполнить отмеченный пробел. Отметим, что возникающие при изложении этой темы проблемы весьма глубоки и опираются на теоретические положения линейной алгебры.

Пусть решается система линейных уравнений $AX = F$. (1)

Стационарные итерационные процессы строятся по следующей схеме. Выбирается неособенная матрица H , и невязка уравнения (1) $(F - AX)$ умножается на H , к результату прибавляется и вычитается вектор X . Это приводит к эквивалентной системе

$$X = X + H \cdot (F - AX). \quad (2)$$

Обозначив $B = E - HA$ и $G = HF$, систему (2) перепишем в виде

$$X = BX + G. \quad (3)$$

Итерационный процесс при выборе некоторого начального вектора $X^{(0)}$ осуществляется в соответствии с рекуррентными равенствами

$$X^{(k)} = BX^{(k-1)} + G, \quad k = 1, \dots \quad (4)$$

Основным вопросом является сходимость построенной последовательности итераций. Напомним центральную теорему о сходимости.

Необходимым и достаточным условием сходимости итерационного процесса (4) является то, что все собственные значения матрицы B по модулю были меньше единицы.

Исходя из свойств матрицы системы A , может быть подобрана матрица H , обеспечивающая выполнение условия теоремы. Вообще говоря, самым хорошим случаем был бы выбор в качестве матрицы H матрицы, близкой к A^{-1} . Трудоемкость нахождения такой матрицы H может оказаться соизмеримой с общей трудоемкостью самого итерационного процесса.

Конечно, важно иметь универсальный метод построения матрицы H , обладающий небольшой трудоемкостью и обеспечивающий сходимость итерационного процесса.

1. Рассмотрим один такой метод. Пусть в исходной системе матрица A симметрична и положительно определена. Симметричная положительно-определенная матрица обладает важным свойством – все ее собственные значения вещественны и положительны.

Выберем в качестве $H = \frac{2}{\mu} E$. Тогда

$$B = E - \frac{2}{\mu} A, \quad (5)$$

где

$$\mu = \|A\|_1 = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{i,j}|.$$

Чтобы установить сходимость итерационного процесса с матрицей (5), надо оценить по модулю собственные значения матрицы B . Поскольку все собственные значения матрицы A заключены в открытом интервале $(0; \mu)$, что следует из неравенства $|\lambda| < \|A\|_1$ (λ – собственные значения матрицы A). Собственные значения θ матрицы B представимы в виде

$$\theta = 1 - \frac{2}{\mu} \cdot \lambda.$$

Следовательно, все собственные значения θ матрицы B принадлежат открытому промежутку $(-1; 1)$. Напомним, что собственные значения матрицы B вещественны.

Таким образом, установили, что процесс итераций, определяемый формулой (4) при выбранной матрице B , сходится в соответствии с центральной теоремой сходимости итерационных процессов.

Может сложиться представление, что требование симметричности и положительной определенности матрицы A исходной системы (1) обременительно. Но это не так. Достаточно исходную систему (1) с произвольной неособенной

матрицей умножить на A^T и будем иметь дело с системой, матрица которой

$$A^* = A^T A$$

симметричная и положительно-определенная.

Пример 1. Применяя описанную схему, решим систему линейных уравнений

$$AX = F, \text{ при}$$

$$A = \begin{pmatrix} 1,00000 & 0,17536 & -0,25761 & 0,54327 \\ 0,47125 & 1,00000 & 0,67785 & -0,32112 \\ -0,11375 & 0,35361 & 1,00000 & -0,74981 \\ 0,55116 & 0,43595 & 0,36765 & 1,00000 \end{pmatrix},$$

$$F = \begin{pmatrix} 0,31358 \\ 0,52463 \\ 0,73375 \\ 0,91231 \end{pmatrix}.$$

Далее приведены необходимые промежуточные вычисления:

$$A^T = \begin{pmatrix} 1,00000 & 0,47125 & -0,11375 & 0,55116 \\ 0,17536 & 1,00000 & 0,35361 & 0,43595 \\ -0,25761 & 0,67785 & 1,00000 & 0,36765 \\ 0,54327 & -0,32112 & -0,74961 & 1,00000 \end{pmatrix},$$

$$A^T F = \begin{pmatrix} 0,98018 \\ 1,23680 \\ 1,34400 \\ 0,36417 \end{pmatrix},$$

$$A^* = A^T A =$$

$$= \begin{pmatrix} 1,53879 & 0,84667 & 0,15071 & 1,02837 \\ 0,84667 & 1,34584 & 1,14656 & -0,05497 \\ 0,15071 & 1,14656 & 1,66101 & -0,73958 \\ 1,02837 & -0,05497 & -0,73958 & 1,96018 \end{pmatrix}, \quad (6)$$

$$\mu = 3,783101,$$

$$B = \begin{pmatrix} 0,18649 & -0,44760 & -0,07968 & -0,54367 \\ -0,44760 & 0,28850 & -0,60615 & 0,02906 \\ -0,07968 & -0,60615 & 0,12188 & 0,39099 \\ -0,54367 & 0,02906 & 0,39099 & -0,03628 \end{pmatrix},$$

$$G = \begin{pmatrix} 0,1819 \\ 0,65386 \\ 0,71053 \\ 0,19253 \end{pmatrix}.$$

За начальное приближение принята правая часть G , то есть $X^{(0)} = G$. Промежуточные приближения предлагаемого итерационного процесса приведены в табл. 1.

Таблица 1

K	$x^{(k)}_1$	$x^{(k)}_2$	$x^{(k)}_3$	$x^{(k)}_4$
20	0,35518	-0,17964	1,06856	0,39870
40	0,45271	-0,34179	1,17068	0,39870
60	0,48353	-0,39187	1,20171	0,37469
80	0,49307	-0,40737	1,21131	0,37276
100	0,49602	-0,41217	1,21428	0,37216
120	0,49694	-0,41366	1,21520	0,37198
140	0,49722	-0,41412	1,21548	0,37192
160	0,49731	-0,41426	1,21557	0,37190
180	0,49734	-0,41430	1,21560	0,37190
200	0,49734	-0,41432	1,21561	0,37190
220	0,49735	-0,41432	1,21561	0,37190

Процесс итерации в изложенном варианте дает сходящийся результат. Правда, сходимость очень медленная, и это говорит о том, что модуль наибольшего по модулю собственного значения построенной матрицы B меньше единицы, но близок к ней, а норма матрицы B может быть больше единицы.

В нашем примере $\|B\|_I = 1,371312$.

Применение достаточного признака сходимости возможно только в одном случае – использовании третьей (сферической) нормы вектора и согласованной с ней третьей нормой матрицы. Напомним, что для X – вектора размерности n

$$\|X\|_{III} = \sqrt{\sum_{i=1}^n x_i^2}.$$

Согласованная с третьей нормой вектора норма квадратной матрицы A с вещественными элементами определяется следующим образом:

$$\|A\|_{III} = \sqrt{\Lambda_1},$$

где Λ_1 – наибольшее собственное значение матрицы $A^T A$.

Матрица $A^T A$ – симметричная положительно-определенная. У такой матрицы все собственные значения вещественные и положительные.

Было установлено, что у построенной матрицы $B = E - \frac{2}{\mu} A$ (A – симметричная положительно-определенная матрица) собственные значения θ имеют вид $\theta = 1 - \frac{2}{\mu} \cdot \lambda$, где λ – собственные значения матрицы A . Третья норма матрицы B будет такой: $\|B\|_{III} = \max_{\lambda} \left| 1 - \frac{2}{\mu} \cdot \lambda \right| < 1$.

Это вытекает из того, что собственные значения матрицы B одновременно являются собственными значениями транспонированной матрицы B^T . Таким образом, собственные значения матрицы $B^T B$ являются квадратами матрицы B , а корень квадратный от них – модуль собственных значений матрицы B .

Для анализа рассмотренного примера была решена полная проблема собственных значений для матрицы A^* (6). Было найдено:

$$\lambda_1 = 3,0157361, \lambda_2 = 2,9681834, \\ \lambda_3 = 0,4110713, \lambda_4 = 0,1077101.$$

Тогда собственными значениями матрицы B будут:

$$\theta_1 = 0,943057, \theta_2 = 0,782680, \\ \theta_3 = -0,569180, \theta_4 = -0,594320,$$

а третья норма матрицы $\|B\|_{III} = 0,943057$.

Доказано, что при выполнении достаточного условия сходимости процесса итераций (любая норма матрицы $\|B\| < 1$) имеет место оценка погрешности

$$\|X^* - X^{(k)}\| \leq \frac{\|B\|^k}{1 - \|B\|} \cdot \|G\|. \quad (7)$$

В нашем случае достаточное условие сходимости выполнено для третьей нормы матрицы, поэтому, используя полученную величину $\|B\|_{III}$ и вычисленное значение $\|G\|_{III} = 1,112638$, неравенство (7) переписывается в следующем виде:

$$\|X^* - X^{(k)}\| \leq \frac{0,943057^k}{1 - 0,943057} \cdot 1,112638 = 19,538496 \cdot 0,943057^k. \quad (8)$$

Неравенство (8) дает нам гарантированную оценку погрешности приближенного решения $X^{(k)}$, полученного в рассмотренном итерационном процессе. За двести шагов имеем

$$\|X^* - X^{(200)}\| \leq 19,538496 \cdot 0,943057^{200} = \\ = 19,538496 \cdot 0,000008 = 0,000158.$$

При установлении неравенства (8) для нахождения нормы $\|B\|_{III}$ нами была решена полная проблема собственных значений для матрицы A^* . На самом деле без этого, конечно, можно обойтись. Тем более, что решение полной проблемы собственных значений не менее трудоемко, чем решение системы линейных алгебраических уравнений. Поэтому в описанной ситуации возможны два выхода. Первый. Установив факт сходимости предложенного итерационного процесса, осуществить его как вычислительный процесс (на компьютере). Второй. Если требуется установление гарантированной оценки погрешности типа (7), (8), то в ходе реализации итерационного процесса найти наибольшее по модулю собственное число матрицы B . Это можно сделать практически без увеличения трудоемкости процесса итераций.

Кратко наметим, как это сделать. На каждом шаге k итерационного процесса находится вектор $X^{(k)} = (B^k + B^{k-1} + B^{k-2} + \dots + E) \cdot G$. При этом осуществляется одно умножение вектора $X^{(k-1)}$ на матрицу B и одно сложение (полученного произведения со свободным членом G). Если каждый раз вычислять вектор $r_k = X^{(k)} - X^{(k-1)}$, то на каждом итерационном шаге (при $k > 2$)

кроме вектора r_k имеется еще вектор r_{k-1} (сохраненный от предыдущего шага). Векторы r_k и r_{k-1} имеют вид

$$r_k = B^k G, r_{k-1} = B^{k-1} G, \quad (9)$$

а это именно те векторы, которые строятся в численном методе нахождения наибольшего по модулю собственного значения матрицы B . В качестве произвольного вектора, фигурирующего в этом методе, здесь выступает вектор G . Напомним, что за приближение к отыскиваемому собственному значению берется отношение компоненты вектора r_k к соответствующей компоненте вектора r_{k-1} . Нахождение наибольшего по модулю собственного значения, осуществленного внутри итерационного процесса, в нашем примере привело к сходимости к $\theta_1 = 0,943057$ за 80 шагов.

2. Рассмотрим приведение системы с матрицей специального вида. В прикладных задачах часто встречаются системы, в которых диагональные элементы преобладают над остальными элементами. В этом случае подготовку системы можно осуществить так.

Перепишем систему (1) в развернутом виде:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = f_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = f_2, \\ \dots, \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = f_n.$$

Так как на диагонали находятся доминирующие элементы, можно каждое уравнение разделить на соответствующий диагональный элемент $a_{i,i}$ и разрешить относительно x_i :

$$x_1 = \frac{f_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2 - \dots - \frac{a_{1n}}{a_{11}}x_n, \\ x_2 = \frac{f_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1 - \dots - \frac{a_{2n}}{a_{22}}x_n, \\ \dots, \\ x_n = \frac{f_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}}x_1 - \dots - \frac{a_{n,n-1}}{a_{nn}}x_{n-1}.$$

В матричном виде последняя система записывается в виде

$$X = BX + G, \quad (10)$$

где $\|B\|_I = \max_{1 \leq i \leq n} \frac{1}{|a_{i,i}|} \cdot \sum_{i \neq j} |a_{i,j}|$,

$$B = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \dots & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \dots & 0 \end{pmatrix}, H = D^{-1},$$

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}, G = \begin{pmatrix} f_1 \\ a_{11} \\ f_2 \\ a_{22} \\ \dots \\ f_n \\ a_{n,n} \end{pmatrix}. \quad (11)$$

Метод итераций, примененный к (10) при (11), носит название метода простой итерации.

Достаточное условие $\|B\|_1 < 1$ может быть выполнено только в случае

$$|a_{i,j}| > |a_{i,1}| + \dots + |a_{i,i-1}| + |a_{i,i+1}| + \dots + |a_{i,n}|,$$

что является весьма жестким требованием на матрицу A.

Из (10) и (11) следует, что необходимое и достаточное условие сходимости в данном случае состоит в выполнении неравенства $|\lambda| < 1$, где λ – любое собственное значение матрицы $B = E - D^{-1} \cdot A$. Если записать определители, участвующие в установлении условия сходимости, в форме

$$\begin{aligned} \text{Det}(B - \lambda E) &= \text{det}(E - D^{-1}A - \lambda E) = \\ &= \text{det}(D^{-1}) \cdot \text{det}(D - A - \lambda D) = \\ &= (-1)^n \cdot \text{det}(D^{-1}) \cdot \text{det}(A - D + \lambda D), \end{aligned}$$

то необходимое и достаточное условие сходимости метода простой итерации можно сформулировать так: все корни уравнения

$$\text{det}(A - D + \lambda D) = \begin{vmatrix} a_{11} \cdot \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} \cdot \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{n,n} \cdot \lambda \end{vmatrix} = 0 \quad (12)$$

должны быть по модулю меньше единицы.

Конечно, прямая проверка выполнимости условия (12) весьма трудоемка. Однако в случае симметричной матрицы A с положительными диагональными элементами необходимому и достаточному условию сходимости метода простой итерации можно придать легко проверяемую форму. Именно, если матрица A обладает названными свойствами, то необходимым и достаточным условием сходимости метода простой итерации является положительная определенность матриц A и 2D - A. (Подробное доказательство этого утверждения можно найти в кн.: Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры. М.: Физматгиз, 1960. 656 с.)

Пример 2. Пусть решается система $AX = F$, где

$$A = \begin{pmatrix} 2,8 & -1,4 & 0 & 0 \\ -1,4 & 2,6 & -1,2 & 0 \\ 0 & -1,6 & 2,3 & -1,1 \\ 0 & 0 & -1,1 & 2,2 \end{pmatrix}, F = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \quad (13)$$

Системы, имеющие структуру матрицы A вида (13) или подобного типа, часто возникают при решении уравнений математической физики конечно-разностными (сеток) методами.

Будем решать (13) методом простой итерации: $X^{(k)} = BX^{(k-1)} + F$,

$$B = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{7}{13} & 0 & \frac{6}{13} & 0 \\ 0 & \frac{12}{23} & 0 & \frac{11}{23} \\ 0 & 0 & \frac{1}{2} & 0 \end{pmatrix}, F = \begin{pmatrix} \frac{5}{14} \\ \frac{5}{13} \\ \frac{10}{23} \\ \frac{5}{11} \end{pmatrix}.$$

Ниже, пока без теоретического обоснования сходимости, приведена таблица приближений, построенных методом простой итерации (табл. 2).

Таблица 2

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
0	0,357143	0,384615	0,434783	0,454545
10	1,389947	2,120069	2,234214	1,543628
20	1,514513	2,321066	2,439823	1,671202
30	1,528982	2,344412	2,463705	1,686020
40	1,530662	2,347124	2,466479	1,687741
50	1,530857	2,347439	2,466801	1,687941
60	1,530880	2,347475	2,466839	1,687961
70	1,530883	2,347480	2,466843	1,687967

Таким образом, сходимость процесса простой итерации установлена практически. Видим, что скорость сходимости мала.

Установим факт сходимости из теоретических соображений. Вообще говоря, теоретическое обоснование сходимости надо делать до практической реализации метода. В прикладной математике, одной из основных составляющих которой является вычислительная математика (численные методы), часто теоретические исследования и численный эксперимент трудно отделить друг от друга.

Ранее было установлено, что сходимость простой итерации имеет место, если матрица A преобразованной системы симметрична, положительно определена и имеет положительные диагональные элементы. Кроме этого матрица 2D - A симметрична и положительно определена. Матрица D - диагональная, причем ее диагональные элементы совпадают с соответствующими диагональными элементами матрицы A. Таким образом, матрица 2D - A имеет такие же диагональные элементы, что и матрица A, а остальные (недиагональные) элементы отличаются от соответствующих элементов A знаком.

Проверим выполнение перечисленных условий для матриц A и 2D - A.

Матрица A симметричная, элементы ее главной диагонали положительны. Установим ее по-

ложительную определенность. Известно, что матрица положительно определена, если выполнены неравенства (критерий Сильвестера)

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} > 0, \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} > 0 \quad \text{и т. д. (14)}$$

Для нашей матрицы имеем

$$\begin{vmatrix} 2,8 & -1,4 \\ -1,4 & 2,6 \end{vmatrix} = 5,32 > 0, \\ \begin{vmatrix} 2,8 & -1,4 & 0 \\ -1,4 & 2,6 & -1,2 \\ 0 & -1,2 & 2,3 \end{vmatrix} = 8,204 > 0, \\ \begin{vmatrix} 2,8 & -1,4 & 0 & 0 \\ -1,4 & 2,6 & -1,2 & 0 \\ 0 & -1,2 & 2,3 & -1,1 \\ 0 & 0 & -1,1 & 2,2 \end{vmatrix} = 11,6116 > 0. \quad (15)$$

Положительная определенность матрица А установлена. Проверка выполнения неравенств типа (14) для матрицы 2D – А приводит к вычислению определителей (15), у которых знаки недиагональных элементов изменены на минус. Вычисления показывают, что значения этих определителей совпадают с определителями в (15). Положительная определенность 2D – А также установлена. Это гарантирует сходимость.

Первая норма матрицы В в рассматриваемом варианте метода простой итерации равна единице. Действительно,

$$\|B\|_1 = \max\{1; \frac{7}{13} + \frac{6}{13}; \frac{12}{23} + \frac{11}{23}; 1\} = 1.$$

Последнее обстоятельство объясняет, почему скорость сходимости последовательности приближений метода простой итерации в данном примере мала. Это связано с тем, что среди собственных значений матрицы В есть близкие по модулю к единице (хотя все $|\lambda| < 1$).

В заключение отметим, что и в методе простой итерации можно пользоваться третьей векторной нормой и третьей нормой матрицы В. Тогда появляется необходимость нахождения наибольшего по модулю собственного значения матрицы В, которое можно осуществить по описанной в первой части статьи схеме.

А. В. Панкратова

ОБ УТОЧНЕНИИ НЕРАВЕНСТВА КОШИ МЕТОДОМ MIHÁLY BENCZE

В статье доказывается уточнение неравенства Коши методом М. Bencze, демонстрируются применения установленного неравенства при решении задач.

В работе [1] М. Bencze описан следующий результат. Пусть $x \in (0; +\infty)$, $a_i > 0$, $i = 1, \dots,$

$$n, B(x) = \sqrt[n]{\prod_{i=1}^n (x + a_i)} - x. \text{ Справедлива следующая}$$

Теорема (М. Bencze). Функция $B(x)$ возрастает,

причем $G_n \leq B(x) \leq A_n$, где $G_n = \sqrt[n]{\prod_{i=1}^n a_i}$

есть среднее геометрическое, а $A_n = \frac{\sum_{i=1}^n a_i}{n}$ – среднее арифметическое чисел a_1, a_2, \dots, a_n .

Сформулированное утверждение естественно понимать как уточнение неравенства Коши для арифметико-геометрических средних положительных чисел.

Цель настоящей работы – обобщить теорему Mihály Bencze на случай взвешенных средних, а также рассмотреть некоторые приложения полученных результатов.

Теорема. Пусть $x \in (0; +\infty)$, $a_i > 0$, $i = 1, \dots, n$, p_1, \dots, p_n – положительные числа, удовлетворяющие условию $\sum_{i=1}^n p_i = 1$. Тогда функция

$$\bar{B}(x) = \prod_{i=1}^n (x + a_i)^{p_i} - x \text{ – неубывающая, причем} \\ \bar{G}_n \leq \bar{B}(x) \leq \bar{A}_n, \quad (1)$$

где $\bar{G}_n = \prod_{i=1}^n a_i^{p_i}$ – взвешенное среднее геометрическое, а $\bar{A}_n = \sum_{i=1}^n p_i a_i$ – взвешенное среднее арифметическое чисел a_1, a_2, \dots, a_n с набором весов p_1, \dots, p_n .

Доказательство. Установим, что функция $\bar{B}(x)$ не убывает. Для этого найдем ее производную и покажем, что она неотрицательна.

$$\bar{B}'(x) = \left(\frac{p_1}{x + a_1} + \dots + \frac{p_n}{x + a_n} \right) \cdot \prod_{i=1}^n (x + a_i)^{p_i} - 1.$$

ПАНКРАТОВА Лариса Валерьевна – старший преподаватель кафедры прикладной математики ВятГУ © Панкратова А. В., 2008

Неравенство $\overline{B}'(x) \geq 0$ равносильно соотношению $\left(\sum_{i=1}^n \frac{p_i}{x+a_i}\right)^{-1} \leq \prod_{i=1}^n (x+a_i)^{p_i}$, которое вытекает из монотонности взвешенного среднего степенного, так как представляет неравенство между средним гармоническим и средним геометрическим. Заметим, что в случае $\overline{B}'(x) = 0$ получаем соотношение $\overline{B}(x) = \text{const} = a$, которое описывается условием $a_1 = \dots = a_n = a$.

Итак, $\overline{B}(x)$ – неубывающая функция, следовательно, для нее справедливо соотношение

$$\lim_{x \rightarrow 0+} \overline{B}(x) \leq \overline{B}(x) \leq \lim_{x \rightarrow +\infty} \overline{B}(x). \quad (2)$$

Один из пределов в (2) очевиден: $\lim_{x \rightarrow 0+} \overline{B}(x) = \prod_{i=1}^n a_i^{p_i} = \overline{G}_n$. Вычисление же другого предполагает раскрытие неопределенности вида $\infty - \infty$. Представим выражение под знаком предела следующим образом:

$$\lim_{x \rightarrow +\infty} \overline{B}(x) = \lim_{x \rightarrow +\infty} \frac{\prod_{i=1}^n \left(1 + \frac{a_i}{x}\right)^{p_i} - 1}{\frac{1}{x}}.$$

Теперь воспользуемся правилом Лопиталья-Бернулли раскрытия неопределенности вида $\left(\frac{0}{0}\right)$. После сокращения числителя и знаменателя на $-\frac{1}{x^2}$ получим:

$$\lim_{x \rightarrow +\infty} \overline{B}(x) = \lim_{x \rightarrow +\infty} \left((p_1 a_1 + \dots + p_n a_n) \cdot \prod_{i=1}^n \left(1 + \frac{a_i}{x}\right)^{p_i} \right) = \overline{A}_n.$$

Теорема полностью доказана.

З а м е ч а н и е. Положив $x = 1$ в соотношении, установленном данной теоремой, получим,

что $\overline{B}(x)|_{x=1} = \prod_{i=1}^n (1+a_i)^{p_i} - 1$. Это позволяет левую часть доказанного неравенства рассматривать как обобщение неравенства Гюйгенса:

$\prod_{i=1}^n (1+a_i)^{p_i} \geq 1 + \prod_{i=1}^n a_i^{p_i}$, где $a_i > 0$, $i = 1, \dots, n$, p_1, \dots, p_n – положительные числа, удовлетворяющие условию $\sum_{i=1}^n p_i = 1$ (см., напр., [2]).

Рассмотрим теперь некоторые применения описанных результатов.

1. Сравните числа $\sqrt[3]{2730} - 12$ и $\sqrt[3]{3360} - 13$.
Р е ш е н и е. Воспользуемся свойством моно-

тонности функции $B(x) = \sqrt[n]{\prod_{i=1}^n (x+a_i)} - x$, установленным М. Bencze. Пусть $a_1 = 1$, $a_2 = 2$, $a_3 = 3$. Вычислим значения функции $B(x)$ для $x = 12$ и для $x = 13$. Получим, что

$$B(12) = \sqrt[3]{(12+1)(12+2)(12+3)} - 12 = \sqrt[3]{2730} - 12,$$

$$B(13) = \sqrt[3]{(13+1)(13+2)(13+3)} - 13 = \sqrt[3]{3360} - 13.$$

Так как числа a_1, a_2, a_3 различны, то $B(12) < B(13)$, следовательно, $\sqrt[3]{2730} - 12 < \sqrt[3]{3360} - 13$.

2. Сравните числа $\sqrt[3]{2700} - 10$ и $\sqrt[3]{1690} - 8$.

Эту задачу мы адресуем читателю, она аналогична предыдущей. Для решения достаточно подобрать нужную совокупность чисел $a_i > 0$, $i = 1, \dots, n$ с соответствующим набором весов p_1, \dots, p_n и воспользоваться свойством монотонности функции $\overline{B}(x)$, установленным выше.

3. [3, с. 207]. Произведение положительных чисел a_1, a_2, \dots, a_n равно 1. Докажите, что $(1+a_1)(1+a_2)\dots(1+a_n) \geq 2^n$.

Р е ш е н и е. Авторы предлагают реализовать индуктивное решение данной задачи, хотя и делают ссылку на возможность использования других подходов. Мы поступим следующим образом: вычислим значение функции

$B(x) = \sqrt[n]{\prod_{i=1}^n (x+a_i)} - x$ при $x = 1$ и оценим результат снизу по теореме М. Bencze. Получим, что $\sqrt[n]{(1+a_1)(1+a_2)\dots(1+a_n)} - 1 \geq \sqrt[n]{a_1 a_2 \dots a_n} = 1$, исходя из условий, наложенных на числа a_1, a_2, \dots, a_n . Очевидно, что приведенное соотношение немедленно влечет неравенство, данное в задаче.

Несложно видеть и различные пути обобщения и видоизменения данной задачи, исходя из неравенства (1).

4. Найдите значение предела

$$\lim_{n \rightarrow \infty} \frac{\sum_{m=1}^n \left(\prod_{k=1}^m \left(m + \frac{1}{k(k+1)} \right)^{1/m} - m \right)}{n}.$$

Р е ш е н и е. Рассмотрим последовательность $\{b_m\}$, для которой $b_m = \prod_{k=1}^m \left(m + \frac{1}{k(k+1)} \right)^{1/m} - m$. Заметим, что исходя из теоремы М. Bencze

$$\lim_{m \rightarrow \infty} b_m = \lim_{m \rightarrow \infty} B(x)|_{x=m} = \lim_{m \rightarrow \infty} \frac{\sum_{k=1}^m \frac{1}{k(k+1)}}{m} = \lim_{m \rightarrow \infty} \frac{1}{m} = 0.$$

Воспользуемся теперь известным утверждением о том, что если последовательность $\{b_m\}$ имеет предел (конечный или нет), то тот же предел имеет и последовательность $\{c_m\}$, для которой

$$c_m = \frac{b_1 + b_2 + \dots + b_m}{m} \quad (\text{доказательство приведенного}$$

утверждения можно найти, например, в [4, с. 79]).

Так что
$$\lim_{n \rightarrow \infty} \frac{\sum_{m=1}^n \left(\prod_{k=1}^m \left(m + \frac{1}{k(k+1)} \right)^{1/m} - m \right)}{n} = 0.$$

5. Имеет ли уравнение

$$\left(x + \frac{1}{\sin \frac{2\pi}{15}} \right) \left(x + \frac{1}{\sin \frac{4\pi}{15}} \right) \left(x + \frac{1}{\sin \frac{8\pi}{15}} \right) = \left(x + \frac{1}{3 \sin \frac{\pi}{15}} \right)^3$$

решения на множестве положительных чисел?

Решение. Перепишем данное уравнение в равносильном виде:

$$\left(x + \frac{1}{\sin \frac{2\pi}{15}} \right)^{1/3} \left(x + \frac{1}{\sin \frac{4\pi}{15}} \right)^{1/3} \left(x + \frac{1}{\sin \frac{8\pi}{15}} \right)^{1/3} - x = \frac{1}{3 \sin \frac{\pi}{15}}.$$

Применяя теперь теорему М. Bencze (а это возможно), получим:

$$\left(x + \frac{1}{\sin \frac{2\pi}{15}} \right)^{1/3} \left(x + \frac{1}{\sin \frac{4\pi}{15}} \right)^{1/3} \left(x + \frac{1}{\sin \frac{8\pi}{15}} \right)^{1/3} -$$

$$-x < \frac{1}{3 \sin \frac{2\pi}{15}} + \frac{1}{3 \sin \frac{4\pi}{15}} + \frac{1}{3 \sin \frac{8\pi}{15}}.$$

Заметим, что

$$\frac{1}{\sin \frac{2\pi}{15}} = \operatorname{ctg} \frac{\pi}{15} - \operatorname{ctg} \frac{2\pi}{15},$$

$$\frac{1}{\sin \frac{4\pi}{15}} = \operatorname{ctg} \frac{2\pi}{15} - \operatorname{ctg} \frac{4\pi}{15},$$

$$\frac{1}{\sin \frac{8\pi}{15}} = \operatorname{ctg} \frac{4\pi}{15} - \operatorname{ctg} \frac{8\pi}{15}.$$

Складывая эти равенства почленно, получим, что

$$\frac{1}{\sin \frac{2\pi}{15}} + \frac{1}{\sin \frac{4\pi}{15}} + \frac{1}{\sin \frac{8\pi}{15}} = \operatorname{ctg} \frac{\pi}{15} - \operatorname{ctg} \frac{8\pi}{15} =$$

$$= \frac{\sin \left(\frac{8\pi}{15} - \frac{\pi}{15} \right)}{\sin \frac{\pi}{15} \sin \frac{8\pi}{15}} = \frac{\sin \left(\pi - \frac{8\pi}{15} \right)}{\sin \frac{\pi}{15} \sin \frac{8\pi}{15}} = \frac{1}{\sin \frac{\pi}{15}}.$$

Возвращаясь к данному уравнению, заключаем, что оно не может иметь решений на указанном промежутке.

6. При каком значении параметра а функция

$$f(x) = (x+1)^{1/3} \cdot \left(x + \frac{1}{\sin^2 \frac{\pi a}{2}} \right)^{2/3} - x \quad \text{достигает}$$

своей точной верхней грани на множестве $(0; +\infty)$?

Решение. Для исследования функции $f(x)$ применим теорему, доказанную выше. Положим

$$a_1 = 1, \quad a_2 = \frac{1}{\sin^2 \frac{\pi a}{2}}, \quad p_1 = \frac{1}{3}, \quad p_2 = \frac{2}{3} \quad (\text{нетрудно}$$

видеть, что при допустимых значениях параметра a все условия используемой теоремы выполняются). Тогда из неравенства (1) следует, что

$$f(x) \leq \frac{1}{3} + \frac{2}{3 \sin^2 \frac{\pi a}{2}}, \quad \text{при этом равенство возможно}$$

только в том случае, когда $a_1 = a_2$. Получаем

уравнение $\sin^2 \frac{\pi a}{2} = 1$, откуда следует, что $a = 2k + 1, k \in \mathbb{Z}$. При этом $f(x) = 0$ для всех значений $x \in (0; +\infty)$.

Примечания

1. Bencze, M. New Proof and Refinement for the AM-GM Inequality [Text] / M. Bencze // Octogon. 2002. Vol. 10. № 1. P. 347–348.

2. Колпащикова, Н. В. Одно доказательство неравенства Коши для взвешенных средних и его применение [Текст] / Н. В. Колпащикова // Некоторые вопросы теории среднего степенного: сб. науч. ст. Киров: Изд-во ВГПУ, 1999. С. 62–67.

3. Ленинградские математические кружки [Текст]: пособие для внеклассной работы. Киров: Изд-во «АСА», 1994. 272 с.

4. Фихтенгольц, Г. М. Курс дифференциального и интегрального исчисления [Текст]: в 3 т. Т. 1 / Г. М. Фихтенгольц. М: ФИЗМАТЛИТ, Лаборатория Знаний, 2003. 680 с.

**ОБОБЩЕНИЕ ОДНОЙ ТЕОРЕМЫ
О СРЕДНЕМ ЗНАЧЕНИИ**

В статье воспроизводится одна малоизвестная теорема о среднем значении и формулируется ее обобщение в терминах односторонних производных.

В реферате 8Б3 реферативного журнала «Математика» за 1999 г. (№ 8) был анонсирован следующий результат из недоступного для нас иностранного журнала.

Теорема 1. Пусть функция $f(x)$ определена и непрерывна на отрезке $[a; b]$ и дифференцируема на интервале $(a; b)$. Тогда найдется точка $c \in (a; b)$, такая, что будет выполняться равенство $f'(c) \cdot \int_a^b f(x) dx = f(c) \cdot (f(b) - f(a))$.

Заметим, приведенный результат легко следует из классической теоремы Коши о среднем значении [1], покажем это.

Напомним теорему Коши.

Пусть функции $f(x)$ и $g(x)$ определены и непрерывны на отрезке $[a; b]$ и дифференцируемы на интервале $(a; b)$, тогда найдется такая точка $c \in (a; b)$, что будет выполняться равенство $f'(c) \cdot [g(b) - g(a)] = g'(c) \cdot [f(b) - f(a)]$.

Докажем теперь теорему 1. Для этого в условиях теоремы Коши в качестве функции $g(x)$ возьмем интеграл с переменным верхним пределом $\int_a^x f(p) dp$, где $x \in [a; b]$. Так как $g'(c) = f(c)$, то получаем требуемое соотношение

$$f'(c) \cdot \int_a^b f(x) dx = f(c) \cdot (f(b) - f(a)).$$

Следствие 1. В условиях теоремы 1 при $f'(x) \neq 0$ можно ввести оценку интеграла

$\int_a^b f(x) dx$ сверху и снизу:

$$\inf_{x \in [a; b]} \frac{f(x) \cdot (f(b) - f(a))}{f'(x)} \leq \int_a^b f(x) dx \leq \sup_{x \in [a; b]} \frac{f(x) \cdot (f(b) - f(a))}{f'(x)}.$$

Пример 1. Оцените значение интеграла

$$\int_{\pi/2}^{\pi} \frac{\sin x}{x} dx.$$

ПЛЕТНЕВ Константин Владимирович – студент III курса факультета информатики ВятГУ © Плетнев К. В., 2008

Воспользуемся следствием 1. Производная

$$f'(x) = \frac{x \cos x - \sin x}{x^2}.$$

Подставим данные в неравенство следствия 1:

$$\inf_{x \in [\pi/2; \pi]} \frac{2/\pi \cdot x \cdot \sin x}{\sin x - x \cdot \cos x} \leq \int_{\pi/2}^{\pi} \frac{\sin x}{x} dx \leq \sup_{x \in [\pi/2; \pi]} \frac{2/\pi \cdot x \cdot \sin x}{\sin x - x \cdot \cos x}.$$

Найдем \inf и \sup функции $F(x) = \frac{2/\pi \cdot x \cdot \sin x}{\sin x - x \cdot \cos x}$ на отрезке $[\pi/2; \pi]$. С помощью производной этой функции нетрудно показать, что точные грани $F(x)$ принимает на концах отрезка. Отсюда получаем нужную оценку:

$$0 \leq \int_{\pi/2}^{\pi} \frac{\sin x}{x} dx \leq 1.$$

Обобщим приведенный результат на случай правосторонних производных. Справедлива

Теорема 2. Пусть функция $f(x)$ определена и непрерывна на отрезке $[a; b]$ и имеет правостороннюю производную на интервале $(a; b)$. Тогда найдутся такие точки c и d , принадлежащие $(a; b)$, что выполняются следующие неравенства:

$$f'_+(c) \cdot \int_a^b f(x) dx \geq f(c) \cdot (f(b) - f(a)),$$

$$f'_+(d) \cdot \int_a^b f(x) dx \leq f(d) \cdot (f(b) - f(a)).$$

Доказательство. Применим следующую теорему Ismat Veg и Akbar Azam из [1].

Пусть функции $f(x)$ и $g(x)$ определены и непрерывны на отрезке $[a; b]$. Если $f(x)$ и $g(x)$ имеют правосторонние производные на интервале $(a; b)$, то существуют точки $c, d \in (a; b)$, такие, что выполняются следующие неравенства:

$$[f(b) - f(a)] \cdot g'_+(c) \leq [g(b) - g(a)] \cdot f'_+(c)$$

$$\text{и } [f(b) - f(a)] \cdot g'_+(d) \geq [g(b) - g(a)] \cdot f'_+(d).$$

В условиях вспомогательной теоремы в качестве функции $g(x)$ возьмем интеграл с переменным верхним пределом $\int_a^x f(p) dp$, где $x \in [a; b]$.

Так как $g'_+(c) = f(c)$ и $g'_+(d) = f(d)$, то получаем требуемые соотношения:

$$f(c) \cdot (f(b) - f(a)) \leq f'_+(c) \cdot \int_a^b f(x) dx,$$

$$f(d) \cdot (f(b) - f(a)) \geq f'_+(d) \cdot \int_a^b f(x) dx.$$

Замечание. Очевидно, теорема 2 и нижеприводимое следствие 2 также могут быть сформулированы и для случая левосторонних производных.

Следствие 2. Если в условиях теоремы 2 правосторонняя производная функции $f(x)$ на интервале $(a; b)$ сохраняет знак, то найдутся точки $c, d \in (a; b)$, такие, что будет иметь место оценка

$$\frac{f(c) \cdot (f(b) - f(a))}{f'_+(c)} \leq \int_a^b f(x) dx \leq \frac{f(d) \cdot (f(b) - f(a))}{f'_+(d)}$$

Приведем иллюстрацию теоремы 2.

Пример 2. Найдите хотя бы одну пару точек c и d , удовлетворяющих неравенствам теоремы 2, для функции $f(x) = |x|$, определенной и непрерывной на отрезке $[-2; 1]$ и имеющей правостороннюю производную на интервале $(-2; 1)$.

Все условия теоремы 2 выполнены, запишем

неравенства: $f'_+(c) \cdot \int_{-2}^1 |x| dx \geq |c| \cdot (-1)$,

$f'_+(d) \cdot \int_{-2}^1 |x| dx \leq |d| \cdot (-1)$. Правосторонняя производная равна: $f'_+(x) = \begin{cases} 1, x \geq 0 \\ -1, x < 0 \end{cases}$. Вычислим интеграл:

рал: $\int_{-2}^1 |x| dx = \int_0^1 x dx + \int_{-2}^0 (-x) dx = \frac{x^2}{2} \Big|_0^1 - \frac{x^2}{2} \Big|_{-2}^0 = 2,5$.

Получим следующие неравенства:

$f'_+(c) \cdot 2,5 \geq -|c|$, $f'_+(d) \cdot 2,5 \leq -|d|$. На рис. 1

видно, что неравенство $f'_+(d) \cdot 2,5 \leq -|d|$ выполняется на интервале $(-2; 0)$, а неравенство

$f'_+(c) \cdot 2,5 \geq -|c|$ выполняется на $[0; 1)$, соответственно можно выбрать любую пару точек из полученных интервалов. Ответ: $d = -1$ и $c = 0,5$.

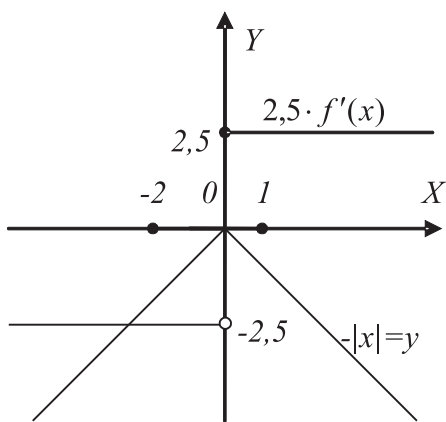


Рис. 1

Пример 3. Дана функция $f(x) = \begin{cases} -x+1, x \geq 0 \\ 1, x < 0 \end{cases}$,

определенная и непрерывная на отрезке $[-1; 2]$, и имеющая правостороннюю производную на интервале $(-1; 2)$. Найдите хотя бы одну пару точек c и d , удовлетворяющих неравенствам теоремы 2.

Все условия теоремы 2 выполнены, запишем

неравенства: $f'_+(c) \cdot \int_{-1}^2 f(x) dx \geq f(c) \cdot ((-2+1)-1)$,

$f'_+(d) \cdot \int_{-1}^2 f(x) dx \leq f(d) \cdot ((-2+1)-1)$. Правосторонняя производная функции $f(x)$ равна:

$f'_+(x) = \begin{cases} -1, x \geq 0 \\ 0, x < 0 \end{cases}$. Вычислим интеграл:

$\int_{-1}^2 f(x) dx = \int_0^2 (-x+1) dx + \int_{-1}^0 1 dx = -\frac{x^2}{2} \Big|_0^2 + x \Big|_{-1}^0 = -2+1+2 = 1$.

Получим следующие неравенства:

$f'_+(c) \geq -2 \cdot f(c)$, $f'_+(d) \leq -2 \cdot f(d)$. Из рис. 2 видно,

что неравенство $f'_+(d) \leq -2 \cdot f(d)$ выполняется на интервале $(0,5; 1)$, а неравенство

$f'_+(c) \geq -2 \cdot f(c)$ выполняется на $(-1; 0,5)$, соответственно можно выбрать любую пару точек из этих интервалов. В точке $x = 1,5$ выполнится равенство. Ответ: $c = -0,5$ и $d = 0,7$.

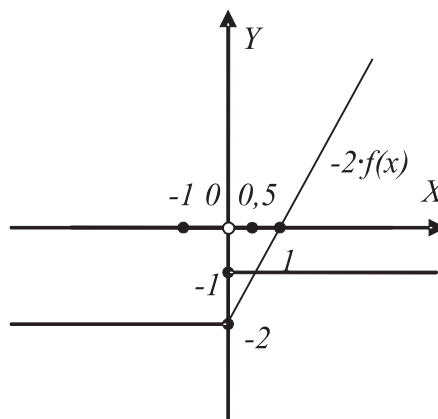


Рис. 2

Примечания

1. Ismat Beg, Akbar Azam. Mean value inequalities and some fundamental results of calculus [Text] / Ismat Beg, Akbar Azam // Australian mathematical society gazette. 1993. 20, № 3. С. 73–79.

М. Х. Прилуцкий, Е. А. Кумагина

**ПРИМЕНЕНИЕ УПРАВЛЯЕМОГО
ФРОНТАЛЬНОГО АЛГОРИТМА
ДЛЯ РЕШЕНИЯ ЗАДАЧИ
МНОГОРЕСУРСНОГО СЕТЕВОГО
ПЛАНИРОВАНИЯ**

В статье рассматривается задача многоресурсного сетевого планирования. Предлагается фронтальный алгоритм ее решения, основанный на принципе оптимальности динамического программирования.

1. Задача многоресурсного сетевого планирования

Изготовление сложных изделий требует выполнения совокупности взаимозависимых работ, связи между которыми описываются с помощью канонических сетевых моделей – ориентированных взвешенных графов без петель и контуров, элементам которых поставлены в соответствие некоторые характеристики. Каноничность сетевой модели означает, что никакая работа не может начать выполняться до тех пор, пока не завершатся все ей предшествующие по технологии изготовления работы.

При описании предметной области мы будем пользоваться понятиями: изделие – продукт трудовой деятельности, работа – составная часть изделия, ресурс – станок для выполнения работы.

Работы сетевых моделей мы будем характеризовать технологическими и организационными условиями. К технологическим условиям относятся: условия взаимозависимости выполнения работ, условия, связанные с длительностями и выполнением работ на станках (работа выполняется на станке без перерывов, на станке одновременно выполняется только одна работа). К организационным условиям относятся условия, связанные с моментами начала и окончания выполнения работ.

Общая проблема управления процессом изготовления сложных изделий заключается в определении места (станка) и сроков начала и окончания выполнения работ, таким образом, чтобы, не нарушая требования технологического и, возможно, организационного характера, обеспечить изготовление всей заданной совокупности изделий.

2. Математическая модель

Исходные параметры математической модели

$T = \{1, 2, \dots, T_0\}$ – множество тактов планирования.

ПРИЛУЦКИЙ Михаил Хаимович – доктор технических наук, профессор по кафедре ИАНИ ННГУ
КУМАГИНА Елена Александровна – кандидат технических наук, старший преподаватель кафедры ИАНИ ННГУ

© Прилуцкий М. Х., Кумагина Е. А., 2008

$P = \{1, 2, \dots, k\}$ – множество изделий.

$J(p)$ – множество работ изделия $p \in P$.

$J = \bigcup_{p \in P} J(p) = \{1, 2, \dots, n\}$ – множество всех работ.

$I = \{1, 2, \dots, m\}$ – множество станков.

Технологические параметры

$K(j)$ – множество работ, непосредственно предшествующих работе с номером $j, j \in J$.

$R = \|r_{ij}\|_{m \times n}$, где r_{ij} – время выполнения работы j на станке i .

Организационные параметры

q_j – ранний срок возможного начала выполнения работы $j, j \in J$.

D_j – директивный срок окончания выполнения работы $j, j \in J^D$, где J^D – множество работ, имеющих директивные сроки, $J^D \subseteq J$.

Варьируемые параметры математической модели

$x = (x_1, \dots, x_n)$ – вектор времен начала выполнения работ.

$y = (y_1, \dots, y_n)$ – вектор времен окончания выполнения работ.

$Z = \|z_{ijt}\|_{m \times n \times T_0}$, где

$$z_{ijt} = \begin{cases} 1, & \text{если на станке } i \\ & \text{выполняется работа } j \text{ в такт } t, \\ 0, & \text{в противном случае.} \end{cases}$$

Ограничения математической модели

Технологические ограничения

Естественные ограничения на введенные переменные:

$$\begin{aligned} x_j \in T, y_j \in T, j \in J, \\ z_{ijt} \in \{0,1\}, i \in I, j \in J, t \in T \end{aligned} \quad (1)$$

Взаимозависимость работ, определяющая каноничность сетевой модели, задается ограничениями:

$$x_j \geq 1 + \max_{l \in K(j)} y_l, j \in J. \quad (2)$$

Требования выполнения работ на станке без перерывов:

$$y_j = x_j + r_{ij}, i \in I, j \in J \quad (3)$$

Каждый станок одновременно выполняет одну работу:

$$\sum_{j \in J} z_{ijt} = 1, i \in I, t \in T \quad (4)$$

Каждая работа выполняется на одном станке:

$$\sum_{t \in T} z_{ijt} = r_{ij}, i \in I, j \in J. \quad (5)$$

Организационные ограничения

Ограничения на сроки начала работ:

$$x_j \geq q_j, j \in J. \quad (6)$$

Ограничения на сроки окончания работ:

$$y_j \leq D_j, j \in J^D. \quad (7)$$

Исходные параметры, варьируемые параметры и ограничения (1)-(7) представляют собой общую математическую модель сетевого планирования.

3. Постановка задачи

В рамках построенной общей математической модели ставятся различные оптимизационные задачи. Постановка задачи определяется двумя факторами: множеством допустимых решений и критерием (или группой критериев) оптимальности. Множество допустимых решений задается системой ограничений, которая выбирается из вышеописанных условий (1)-(7). Среди ограничений задачи обязательно должны присутствовать технологические ограничения и могут присутствовать ограничения организационного типа. Вместо ограничений организационного типа могут вводиться соответствующие критерии оптимальности.

Формально задача многоресурсного сетевого планирования ставится следующим образом. Требуется найти совокупность варьируемых параметров (1), удовлетворяющих системе ограничений (2)-(6), для которых достигается минимального значения обобщенный критерий оптимальности (8), связанный со штрафными санкциями, налагаемыми на систему за нарушения директивных сроков выполнения работ

$$F(x, y, z) = \sum_{j \in J} a_j \times \max(0, y_j - D_j) \quad (8),$$

где a_j – штрафные санкции за нарушение на один такт планирования работой j определенного для неё директивного срока, $j \in J^P$.

В дальнейшем решение задачи (1)-(8) мы будем называть расписанием выполнения работ, или просто расписанием. Поставленная задача относится к классу NP-трудных задач, т. е. для ее решения не существует эффективных алгоритмов определения оптимального решения [3].

4. Управляемый фронтальный алгоритм

Для решения поставленной задачи предлагается фронтальный подход, основанный на идеологии «жадных алгоритмов».

При таком подходе в определенные такты времени определяется «фронт работ». Это множество работ, которые могут быть выполнены, начиная с определенного такта. Возникает задача упорядочения работ из фронта работ. После установления порядка выполнения выбранная из фронта работ очередная работа включается в строящееся расписание, причем для ее выполнения используются все доступные к данному моменту времени ресурсы в максимально возможном объеме.

В работах [1, 2] предложены подходы к решению этой задачи. Они основаны на применении специальных перестановочных процедур (порядок выполнения работ определяется путем решения специальных задач о назначениях).

Предлагается развить идею фронтального подхода и принимать решение о выборе очеред-

ной работы, основываясь не на одном конкретном алгоритме, а на наилучшем в данной ситуации.

Будем моделировать процесс построения решения поставленной задачи системой без обратных связей. Пусть $\Phi(t) = \{j_1, j_2, \dots, j_k\}$ – «фронт работ», определенный к такту планирования t – это множество работ, любая из которых может начать выполняться с указанного момента времени, $j_l \in J$, $l = \overline{1, k}$, $t \in T$. Это те работы, для которых все им предшествующие работы уже выполнены, и ранние сроки начала этих работ не превышают указанного момента времени. Множество различных фронтов работ обозначим через W .

Фронт работ на заданный момент времени будем отождествлять с состоянием системы. Основная идея фронтального алгоритма заключается в преобразовании множества $\Phi(t)$ в вектор, который определяет порядок последовательного включения очередной работы в строящееся решение. Для этого на множестве работ из $\Phi(t)$ необходимо установить линейный порядок. На основании линейного порядка, используя «построитель» расписания, будем применять процедуру, которая «жадным» образом включает работы в строящееся расписание.

Пусть Q – множество различных стратегий преобразования множества $\Phi(t)$ в вектор. В качестве таких стратегий могут быть, например, стратегия, основанная на решении специальной задачи о назначениях [1], стратегия упорядочения работ по их резервам времени [4], стратегия, основанная на поиске перестановки, заданной глубины [4], стратегия, основанная на алгоритме Метрополиса [4] и др.

Выбор стратегии будем отождествлять с управлением, применяемым к системе, результатом которого будет часть расписания, построенного «построителем» расписаний. Предполагается, что применение той или иной стратегии требует использования различных временных ресурсов. Пусть τ_q – время, необходимое для реализации стратегии q , $q \in Q$.

Обозначим через $g(\Phi(t), q)$ – суммарное время простоя станков для части расписания, построенного после применения стратегии q к фронту работ $\Phi(t)$.

Рассматриваемая система функционирует следующим образом. В состоянии $\Phi(t)$ к системе применяется управление, задаваемое стратегией q , которое приносит системе «доход» τ_q , связанный с временем на реализацию стратегии q . Под воздействием управления q множество работ из фронта преобразуется в вектор, по которому, используя «построитель» расписания, в уже построенное к этому моменту времени расписание добавляются работы.

При этом система перейдет в новое состояние $\Phi(t')$, где t' – время завершения выполнения самой первой работы из части построенного расписания. Такой переход приносит системе «доход», задаваемый величиной $g(\Phi(t), q)$. При этом предполагается, что величины τ_q и $g(\Phi(t), q)$ измеряются в одних и тех же единицах и суммарный «доход» от перехода системы из состояния $\Phi(t)$ в состояние $\Phi(t')$ под воздействием управления q определяется величиной $\tau_q + g(\Phi(t), q)$.

В новом состоянии $\Phi(t')$ к системе применяется управление q' , $q' \in Q$ и т. д. Система функционирует конечное число тактов T_0 – до момента построения «полного» расписания, включающего в себя все работы из множества J .

Относительно рассматриваемых систем обычно ставят следующую задачу ([5]): при заданном состоянии системы и числе тактов, оставшихся до конца функционирования, найти оптимальную стратегию из некоторого класса стратегий по критерию максимизации суммарного «дохода». Будем задавать стратегию в виде функции $f(\Phi(t), T)$ определенной на множестве $W \times \{1, 2, \dots, T_0\}$ со значениями из Q . Функция $f(\Phi(t), T)$ в зависимости от состояния (фронта работ) и числа тактов, оставшихся до конца функционирования (в общем случае – от числа работ, которые еще не включены в строящееся расписание), определяет, с помощью какой стратегии необходимо установить линейный порядок на множестве работ из фронта работ.

Пусть S – множество различных стратегий. Обозначим через $\mu(\Phi(t), T, f(\Phi(t), T))$ суммарный «доход», который получит система, если она начинает функционировать в состоянии $\Phi(t)$, до конца функционирования осталось T тактов и к системе применяются управления, задаваемые функцией $f(\Phi(t), T)$. Тогда задача выбора оптимальной стратегии будет заключаться в определении такой функции $f_0(\Phi(t), T)$, для которой

$$\mu(\Phi(t), T, f_0(\Phi(t), T)) = \max_{f(\Phi(t), T) \in S} \mu(\Phi(t), T, f(\Phi(t), T)) .$$

Так как поведение системы в момент времени t не зависит от предыстории процесса и общий «доход» системы складывается из «доходов», полученных на каждом шаге, то к системе применим принцип оптимальности динамического программирования.

Обозначим через $v(\Phi(t), T)$ суммарный доход, который получит система, если она начинает свое функционирование из состояния $\Phi(t)$, до конца функционирования осталось T тактов, и к системе на каждом шаге применяются оптимальные управления. Тогда, применив к системе принцип оптимальности динамического программирования, построим рекуррентные соотношения

$$v(\Phi(t), T) = \min_{q \in Q} \{ \tau_q + g(\Phi(t), q) + v(\Phi(t'), T - 1) \},$$

из которых, задав граничные доходы $(\Phi(t), 0) = 0$, $\Phi(t) \in W$, можно находить оптимальную стратегию.

Примечания

1. Прилуцкий, М. Х. Задача упорядочения работ как задача о назначениях [Текст] / М. Х. Прилуцкий, Е. А. Кумагина // Вестник Нижегородского государственного университета. Математическое моделирование и оптимальное управление. Н. Новгород: Изд-во ННГУ, 1999. Вып. 21. С. 270–275.

2. Прилуцкий, М. Х. Перестановочные процедуры решения задач распределения ресурсов [Текст] / М. Х. Прилуцкий, Е. А. Кумагина // Прикладные задачи моделирования и оптимизации: межвуз. сб. науч. тр. Ч. II. Воронеж, 2000. С. 81–90.

3. Прилуцкий, М. Х. Задачи распределения разнородных ресурсов в сетевых канонических структурах [Текст] / М. Х. Прилуцкий, Е. А. Кумагина // Перспективные информационные технологии и интеллектуальные системы. 2000. № 4. С. 46–52.

4. Прилуцкий, М. Х. Многостадийные задачи теории расписаний с альтернативными вариантами выполнения работ [Текст] / М. Х. Прилуцкий, С. Е. Власов // Системы управления и информационные технологии. 2005. № 2. С. 44–48.

5. Казанцев, Э. Н. Об одном классе управляемых марковских цепей [Текст] / Э. Н. Казанцев, М. Х. Прилуцкий // Успехи математических наук. Т. 33. Вып. 6(204). М., 1978. С. 213–214.

НЕКОТОРЫЕ ПРИМЕНЕНИЯ НЕРАВЕНСТВА ГЮЙГЕНСА И ЕГО УТОЧНЕНИЯ

В данной работе рассматриваются некоторые применения уточнения неравенства Гюйгенса в вопросах доказательства неравенств и нахождения значений пределов специального вида. Показываются преимущества метода неравенства Гюйгенса перед стандартными методами.

Уточненное неравенство Гюйгенса фактически нигде не встречается в литературе. Оно было сформулировано и доказано в 2007 г. Т. А. Ананьиной в работе (Ананьина Т. А. Уточнение аналога неравенства Иенсена для логарифмически выпуклых функций и его применения // Вестник Вятского государственного гуманитарного университета. Информатика, математика, язык. Киров: Изд-во ВятГГУ, 2007. С. 135). Напомним его. Это есть неравенство

$$1 + \prod_{i=1}^n a_i^{p_i} \leq \prod_{i=1}^n \dots \prod_{i_k=1}^n \left(1 + \sqrt[k]{a_{i_1} \cdot \dots \cdot a_{i_k}} \right)^{p_{i_1} \cdot \dots \cdot p_{i_k}} \leq \prod_{i=1}^n (1 + a_i)^{p_i}, \quad (*)$$

где a_1, \dots, a_n – положительные числа, p_1, \dots, p_n – набор положительных весов, удовлетворяющий условию $\sum_{i=1}^n p_i = 1$, $1 < k < n$, в котором равенство возможно лишь при условии $a_1 = \dots = a_n$.

Если убрать среднюю часть неравенства (*), то получим весовое неравенство Гюйгенса:

$$1 + \prod_{i=1}^n a_i^{p_i} \leq \prod_{i=1}^n (1 + a_i)^{p_i}.$$

Цель работы – рассмотреть некоторые применения уточнения (*) неравенства Гюйгенса. Перейдём к рассмотрению упомянутых в аннотации вопросов.

Задача 1. Докажите неравенство

$$2 + \prod_{n=1}^{\infty} (b^{\frac{1}{n(n+1)}} - 1)^{\frac{1}{n(n+1)}} < \prod_{n=1}^{\infty} (b^{\frac{1}{n}} + 1)^{\frac{1}{n(n+1)}}, \quad (b > 1).$$

Решение. Составим ряд из показателей степеней $\sum_{n=1}^{\infty} \frac{1}{n(n+1)}$, где n -я частичная сумма ряда

$$\sigma_n = \frac{n}{n+1} \xrightarrow{n \rightarrow \infty} 1.$$

С помощью уточненного неравенства Гюйгенса произведем оценку левой части исходного неравенства сверху и правой – снизу:

$$\begin{aligned} \lim_{n \rightarrow \infty} \left(2 + \prod_{k=1}^n (b^{\frac{1}{k(k+1)}} - 1)^{\frac{1}{k(k+1)}} \right) &< \lim_{n \rightarrow \infty} \left(1 + \prod_{k=1}^n (b^{\frac{1}{k(k+1)}} + 1)^{\frac{1}{k(k+1)}} \right) = \\ &= \lim_{n \rightarrow \infty} \left(1 + \prod_{k=1}^n b^{\frac{1}{k^2(k+1)^2}} \right) = 1 + \lim_{n \rightarrow \infty} \prod_{k=1}^n b^{\frac{1}{k^2(k+1)^2}} = 1 + \lim_{n \rightarrow \infty} b^{\sum_{k=1}^n \frac{1}{k^2(k+1)^2}} = 1 + b^{\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2(k+1)^2}} \\ \lim_{n \rightarrow \infty} \left(\prod_{k=1}^n (b^{\frac{1}{k}} + 1)^{\frac{1}{k(k+1)}} \right) &> \lim_{n \rightarrow \infty} \left(1 + \prod_{k=1}^n (b^{\frac{1}{k}} + 1 - 1)^{\frac{1}{k(k+1)}} \right) = 1 + \lim_{n \rightarrow \infty} \prod_{k=1}^n b^{\frac{1}{k^2(k+1)}} = \\ &= 1 + \lim_{n \rightarrow \infty} b^{\sum_{k=1}^n \frac{1}{k^2(k+1)}} = 1 + b^{\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2(k+1)}} \end{aligned}$$

Рассмотрим значения пределов в полученных оценках:

$$\begin{aligned} \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2(k+1)^2} &= -2 + \lim_{n \rightarrow \infty} \sum_{k=1}^n \left(\frac{1}{k^2} + \frac{1}{(k+1)^2} \right) = -2 + \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} + \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{(k+1)^2} = \\ &= -2 + \frac{\pi^2}{6} + \left(\frac{\pi^2}{6} - 1 \right) = \frac{\pi^2}{3} - 3 \end{aligned}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2(k+1)} = -1 + \lim_{n \rightarrow \infty} \left(\frac{1}{n+1} + \sum_{k=1}^n \frac{1}{k^2} \right) = -1 + \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} - 1$$

Подставив полученный результат в исходное неравенство, мы докажем его справедливость:

$$\lim_{n \rightarrow \infty} \left(2 + \prod_{k=1}^n \left(b^{\frac{1}{k(k+1)}} - 1 \right) \right) < 1 + b^{\frac{\pi^2}{3} - 3} < 1 + b^{\frac{\pi^2}{6} - 1} < \lim_{n \rightarrow \infty} \left(\prod_{k=1}^n \left(b^{\frac{1}{k(k+1)}} + 1 \right) \right).$$

Сформулируем еще одну подобную задачу.

Задача 2. Докажите неравенство

$$\prod_{n=1}^{\infty} \left(36^{\frac{(n(n+1))^2}{(2n-1)(2n+1)^2}} - 1 \right)^{\frac{2n+1}{(n(n+1))^2}} < \prod_{n=1}^{\infty} \left(64^{\frac{n(n+1)}{3(2n+1)}} + 1 \right)^{\frac{2n+1}{(n(n+1))^2}}.$$

Данную задачу адресуем читателю для самостоятельного осмысления.

Задача 3. Вычислите значение произведения $\prod_{j=1}^{\infty} \prod_{k=1}^{\infty} 4^{\frac{1}{2^{j+k}}}$.

Решение. Будем решать данную задачу методом уточненного неравенства Гюйгенса.

$$\text{Имеем: } 1 + \prod_{i=1}^{\infty} 3^{2^i} \leq \prod_{j=1}^{\infty} \prod_{k=1}^{\infty} 4^{\frac{1}{2^{j+k}}}, \text{ так как } a_1 = a_2 = \dots = a_n = \dots, \text{ то } 1 + \prod_{i=1}^{\infty} 3^{2^i} = \prod_{j=1}^{\infty} \prod_{k=1}^{\infty} 4^{\frac{1}{2^{j+k}}}.$$

$$\text{Вычислим левую часть: } \lim_{i \rightarrow \infty} \left(1 + \prod_{n=1}^i 3^{2^n} \right) = 1 + \lim_{i \rightarrow \infty} 3^{\sum_{n=1}^i 2^n} = 4.$$

$$\text{Ответ: } \prod_{j=1}^{\infty} \prod_{k=1}^{\infty} 4^{\frac{1}{2^{j+k}}} = 4.$$

Теперь рассмотрим задачу вычисления предела, определяя сложность вычислений для нее.

Задача 4. Вычислить значение предела

$$\lim_{n \rightarrow \infty} \prod_{i=1}^n \prod_{j=1}^n \prod_{m=1}^n \prod_{k=1}^n \left(1 + \sqrt[4]{a_i \cdot a_j \cdot a_m \cdot a_k} \right)^{\frac{1}{i^4}},$$

$$\text{где } a_i = \frac{1}{1+i^2}, a_j = \frac{1}{1+j^2}, a_m = \frac{1}{1+m^2}, a_k = \frac{1}{1+k^2}.$$

Решение. Будем решать данную задачу с помощью вычислительной техники. Составим алгоритм вычисления. Каждое вложенное произведение дает цикл с параметром, изменяющимся от 1 до n , где $n \rightarrow \infty$.

```

for i:=1 to n do
  for j:=1 to n do
    for m:=1 to n do
      for k:=1 to n do <Вычисление значения>
  
```

Временную сложность <Вычисление значения> возьмем за единицу, тогда временная сложность всего алгоритма будет $T(n) = \Theta(n^4)$.

Приведем таблицу, отражающую количество требуемых операций и временные затраты при решении данной задачи:

	$n \rightarrow \infty$			
	100	1 000	10 000	100 000
n^4	10^8	10^{12}	10^{16}	10^{20}
$t, \text{ с}$	10^{-2}	10^2	10^6	10^{10}
$t, \text{ лет}$	$0,317 \cdot 10^{-9}$	$0,317 \cdot 10^{-5}$	3,17	317

(10^{10} операций/с, 1 год $\approx 3,155 \cdot 10^7$ с.)

Проанализировав значения временной оценки алгоритма, видим, что решение поставленной задачи занимает слишком длительный период времени.

Рассмотрим решение задачи методом уточненного неравенства Гюйгенса. Применим неравенство для оценки выражения под знаком предела сверху и снизу:

$$1 + \prod_{i=1}^n a_i^{\frac{1}{n}} \leq \prod_{i=1}^n \prod_{j=1}^n \prod_{m=1}^n \prod_{k=1}^n \left(1 + \sqrt[4]{a_i \cdot a_j \cdot a_m \cdot a_k} \right)^{\frac{1}{4}} \leq \prod_{i=1}^n (1 + a_i)^{\frac{1}{n}}.$$

Осуществим предельный переход в двойном неравенстве и найдем значения пределов левой и правой частей неравенства:

$$\lim_{n \rightarrow \infty} \left(1 + \prod_{i=1}^n a_i^{\frac{1}{n}} \right) = \lim_{n \rightarrow \infty} \left(1 + \prod_{i=1}^n \left(\frac{1}{1+i^2} \right)^{\frac{1}{n}} \right) = 1 + \lim_{n \rightarrow \infty} \prod_{i=1}^n \left(\frac{1}{1+i^2} \right)^{\frac{1}{n}} = 1$$

$$\lim_{n \rightarrow \infty} \prod_{i=1}^n \left(1 + \frac{1}{a_i} \right)^{\frac{1}{n}} = \lim_{n \rightarrow \infty} \left(\prod_{i=1}^n \left(1 + \frac{1}{1+i^2} \right)^{\frac{1}{n}} \right) = \lim_{n \rightarrow \infty} \prod_{i=1}^n \left(\frac{2+i^2}{1+i^2} \right)^{\frac{1}{n}} = 1$$

Таким образом, имеем $\lim_{n \rightarrow \infty} \prod_{i=1}^n \prod_{j=1}^n \prod_{m=1}^n \prod_{k=1}^n \left(1 + \sqrt[4]{a_i \cdot a_j \cdot a_m \cdot a_k} \right)^{\frac{1}{4}} = 1$, где

$$a_i = \frac{1}{1+i^2}, \quad a_j = \frac{1}{1+j^2}, \quad a_m = \frac{1}{1+m^2}, \quad a_k = \frac{1}{1+k^2}.$$

Ответ: Значение предела равно единице.

Используя вычислительную технику для задачи нахождения предложенного к вычислению предела, мы можем получить лишь приближенное значение. Математический пакет MathCad Professional не дает практического ответа к данной задаче. Решая же методом уточненного неравенства Гюйгенса, мы получаем точный результат.

Приведенные выше задачи позволяют смотреть на уточненное неравенство Гюйгенса (*) как на метод исследования соответствующих вопросов. Мы видим, что во многих случаях этот метод может давать более простое решение задачи.

РАЗВИТИЕ ПОИСКОВОЙ АКТИВНОСТИ ШКОЛЬНИКОВ ПРИ РАБОТЕ С НЕЗАВЕРШЕННЫМИ ЗАДАЧНЫМИ СИТУАЦИЯМИ

На основе анализа психолого-педагогической литературы в статье показывается роль незавершенных задачных ситуаций для усиления поисковой активности школьников. Под такой ситуацией понимается субъективное отношение человека к некоторой задаче, сложившееся к определенному моменту ее решения как естественным образом, так и под воздействием искусственно созданных внешних обстоятельств. Это отношение создается путем специального «прерывания» процесса совместного поиска и зависит как от особенностей организации управления процессом решения задачи, так и степени соответствия трудности задачи уровню математической подготовки школьников.

Обучение решению математических задач традиционно занимает одно из ведущих мест в методических исследованиях. В соответствии с этими исследованиями выделяют несколько этапов работы над задачей. Среди них наибольшую трудность представляет этап поиска решения задачи. Он включает в себя, в первую очередь, отыскание плана решения, реализуемого, как правило, в ходе аналитико-синтетического поиска. Ведущую роль здесь, как известно, играет соотношение требования с данными, а также с имеющимися у учащихся предварительными знаниями о рассматриваемых в условии понятиях и их свойствах. Часто, особенно в решении алгебраических задач, этот процесс заканчивается установлением «типа» задачи, актуализации известного способа получения подзадачи. Нередко поиск решения пронизывает весь процесс решения задачи и содержит несколько переплетающихся между собой циклов, включающих следующие компоненты:

- многосторонний анализ ситуации;
- предварительная прикидка плана решения;
- целенаправленные попытки реализации плана;
- констатация неудачи.

Поиск решения можно считать законченным, когда решение или полностью найдено, или для его завершения осталось выполнить ряд очевидных действий, результаты которых уже не вызывают у решающего каких бы то ни было сомнений. Другими словами, поиск решения задачи заканчивается отысканием не просто какого-либо плана, а плана, заведомо приводящего к цели. Данный этап присутствует в работе над любой задачей, хотя в ряде случаев носит настолько

свернутый характер, что может почти не осознаваться решающим задачу. При этом учителю надо иметь в виду, что свернутость действия (поиска решения) достигается в процессе эффективного формирования у учащегося различных приемов аналитико-синтетического поиска.

Сравнительный анализ психологической и методической литературы свидетельствует, что поиск плана решения задачи является базисом и ориентировочной основой всего процесса решения. Такое понимание сущности рассматриваемого этапа предполагает, что весь процесс решения задачи можно считать с достаточной степенью достоверности адекватным поиску плана ее решения.

Поиск плана решения задачи является не одномоментным действием, а сложной многосторонней работой учащихся. Обучение школьников такому поиску применительно к обучению математике получило широкое развитие под влиянием работ Д. Пойа, в которых с помощью системы советов и указаний предлагалось побудить учащихся к самостоятельному нахождению решения. В дальнейших исследованиях было установлено, что обучение поиску должно представлять собой процесс усвоения общих и специальных приемов решения различных классов задач, а также одновременное формирование необходимых для их решения приемов творческого мышления. При этом обучение поиску целесообразно организовывать с учетом общих и специальных закономерностей его реализации.

Между тем учебная практика свидетельствует о довольно низком уровне поисковой активности школьников, неумении школьников вести целенаправленный поиск плана решения задачи, формализме в знаниях, стремлении учеников лишь запомнить и впоследствии воспроизвести приведенные учителем рассуждения. Наибольшие затруднения возникают в организации поиска плана решения нестандартных задач, требующего привлечения эвристических рассуждений, которые выходят за рамки тривиального перебора возможных альтернатив.

Вполне очевидно, что указанные трудности обусловлены особенностями организации учебно-познавательного процесса, методами и средствами учебной работы, которыми пользуется учитель в своей работе. В частности, к таким особенностям, негативно отражающимся на поисковых возможностях школьников, относятся две характерные стратегии работы учителя. Одна из них предполагает совместное пошаговое движение учителя и учеников к решению, когда сам процесс поиска происходит как последовательное и закономерное появление новых следствий и вспомогательных задач, подкрепляемое время от времени наводящими указаниями учителя.

Развивающий эффект такой работы может проявляться в основном лишь на начальных этапах обучения. Во втором случае учитель просто предлагает условие задачи, дополнительно указывая основополагающий прием, лежащий в основе всего решения. Сам поиск при данном варианте носит во многом спонтанный характер.

Между тем психологи утверждают, что для того чтобы та или иная задачная ситуация непосредственно побуждала учащегося к поисковой активности, она должна содержать в себе некоторую субъективную незавершенность. Под незавершенной задачной ситуацией понимается субъективное отношение человека к некоторой задаче, сложившееся к определенному моменту ее решения как естественным образом, так и под воздействием искусственно созданных внешних обстоятельств. Это отношение создается путем специального «прерывания» процесса совместного поиска и зависит как от особенностей организации управления процессом решения задачи, так и степени соответствия трудности задачи уровню математической подготовки школьников.

В общем случае искусственное «прерывание» задания приобретает смысл тогда, когда человек максимально погружен в работу. Такие моменты называются «точками наиболее сильного контакта учащегося с заданием». В такие моменты контакта с заданием его завершение, как правило, видится решающему относительно близким и достижимым. Ученик уже увидел, как следует сделать то, что требуется, уже «погрузился в работу всей личностью», но еще не вполне прогнозирует, как будет выглядеть итоговый результат.

Найти точки наиболее тесного соприкосновения с заданием на практике достаточно сложно. Как правило, прерывание осуществляется именно в тот момент, когда ученики, полностью погружившись в процесс решения, сталкиваются с необходимостью актуализации существенно иного видения ситуации на основе применения того или иного эвристического приема; намечают возможности для такого применения, но до конца еще не могут дать точный прогноз относительно будущих перспектив избранного подхода. В процессе обсуждения возможных вариантов такого прогноза учитель отслеживает характер «движения мысли» учеников, которые в этом случае не просто комментируют уже выполненные действия и операции, а непосредственно реагируют на складывающуюся ситуацию и тем самым одновременно раскрывают субъективный характер ее динамического восприятия и осмысления.

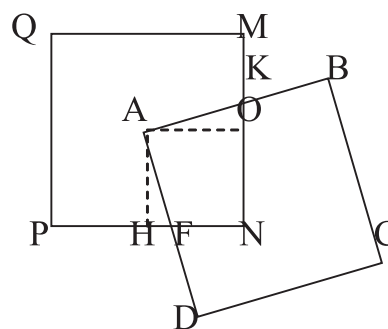
Рассмотрим небольшой пример.

Задача. Вершина A квадрата $ABCD$ расположена в центре квадрата $MNPQ$, а сторона AB отсекает одну треть от MN . Найдите площадь общей части двух квадратов, если $MN = 1$.

Осмысление условия задачи, проводимое учениками совместно с учителем, приводит к целесообразности подбора равновеликой четырехугольной фигуре $AKNF$ фигуры, площадь которой легко можно подсчитать.

Дальнейшая работа с участием учителя в данном случае может заключаться в попытках разбиения искомого четырехугольника на наиболее «простые» составляющие либо его совмещения с известной фигурой с использованием известных видов движения. Наличие своеобразного центра конфигурации наталкивает на возможность рассмотрения в качестве такого вида поворота. Действительно, если применить при решении данной задачи поворот треугольника AKO на 90° , то можно попытаться доказать равновеликость четырехугольника $AKNF$ квадрату $AONH$.

Здесь ситуация незавершенности и соответствующий ей мотивационный импульс достигает пика, обеспечивая почти автоматическое ее завершение (сразу или несколько позднее) даже в том случае, когда учитель искусственно начинает «уводить» ребят на другую проблему, не давая возможности «добраться» до конкретного результата (площадь общей части квадратов равна $1/4$).



Таким образом, в рассмотренном случае именно частичное разрешение задачной ситуации, заключающееся в переосмыслении рассматриваемой фигуры (общей части двух исходных квадратов) в рамках возможных альтернативных геометрических конфигураций, смогло обеспечить подключение механизма незавершенного действия.

А. Н. Соколова

ОБ ИСПОЛЬЗОВАНИИ РЕСУРСОВ ИНТЕРНЕТ В УЧЕБНОЙ И НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ СТУДЕНТОВ-МАТЕМАТИКОВ

В статье дается характеристика популярных русскоязычных ресурсов Интернет, посвященных математике, делается попытка их оценки по степени достоверности данных и приводятся возможности использования подобных ресурсов в учебной и научно-исследовательской работе студентов математических специальностей.

Сегодня не так как прежде важна сумма приобретаемых знаний в той или иной области деятельности человека. Современные знания постоянно изменяются и часто быстро устаревают, поэтому важно, чтобы специалист в любой профессиональной сфере умел самостоятельно учиться, работать с информацией, самостоятельно пополнять и совершенствовать свои знания и умения, поскольку именно этим ему придется заниматься в течение всей профессиональной деятельности. Современное общество заинтересовано в том, чтобы его граждане были способны самостоятельно и активно действовать, принимать решения, быстро адаптироваться к изменяющимся условиям жизни.

Как отмечено в [1], вуз есть «социальный институт, выполняющий заказ общества». Он «должен выполнять задачу подготовки специалиста, способного:

- самостоятельно критически мыслить, уметь видеть возникающие проблемы и находить пути их решения;
- грамотно работать с информацией (уметь собирать факты, анализировать, обобщать их, сопоставлять с аналогичными или альтернативными вариантами решения, устанавливая статистические закономерности, делая аргументированные выводы, умело применяя их для решения новых проблем);
- быть коммуникабельным, контактным в различных средах общения;
- самостоятельно развивать собственный интеллект, нравственность» [2].

Решение отмечаемых задач стоит перед высшей школой и перед образовательными учреждениями общего среднего образования. Таким образом, процесс обучения современного человека становится непрерывным. Целесообразным решением проблемы непрерывного образования является использование новых информационно-коммуникативных образовательных технологий.

СОКОЛОВА Анна Николаевна – аспирант кафедры информатики и методики обучения информатике ВятГУ
© Соколова А. Н., 2008

Еще недавно чуть ли не единственным источником информации для специалистов и людей, получающих то или иное образование, являлись опубликованные статьи и книги, но в век высоких технологий все больше людей отдают предпочтение поиску информации в глобальной сети. Нередко имеет место ситуация, когда студенты ищут нужную информацию не в библиотеках, а в Интернете, не обращая порой должного внимания на достоверность источника. Однако следует помнить, что за большим объемом *не всегда* стоит высокое качество информации. Доверяясь непроверенным сведениям, студент рискует получить неверный или необоснованный результат в своей работе. Особенно остро эта проблема стоит у студентов специальностей, связанных с информационными технологиями.

Приведем пример. На «Математическом форуме» (<http://mathematics.forum24.ru/?1-1-0-00000070-000-0-0-1198365745>) размещена тема, посвященная решению следующей задачи: «Функция $y = f(x)$ определена на всей числовой прямой и является четной периодической функцией с периодом, равным 6. На отрезке $[0;3]$ функция задана формулой $f(x) = 2 + 2x - x^2$. Определите количество нулей этой функции на отрезке $[-5;4]$ ».

В следующем сообщении приводится решение: «Ответ: 4.

Подробности в уме...

Скажу только, что функция $f(x) = 2 + 2x - x^2$ имеет на отрезке $[0;3]$ один корень. А общая функция (функция нашей задачи) достигает локальных максимумов в точках 1, -1 , $-5,5$. А расстояние от точки максимума до корня равно примерно 1,73 (естественно, зависит от того, в какую сторону откладывать)».

При этом никаких пояснений и ссылок на соответствующие утверждения классического анализа не делается. С одной стороны, ответ можно принять как формально правильный, но, очевидно, что представленное решение является неполным, практически отсутствует.

Формальное использование подобных результатов не формирует у изучающих математику необходимых навыков, поскольку студент не вникает в суть задачи, не проводит глубокий анализ и не делает самостоятельных выводов.

Цель данной статьи – охарактеризовать ресурсы Интернет на русском языке, посвященные математике, и попытаться оценить их по степени достоверности данных.

Конкуренцию традиционным составляют электронные библиотеки. Как правило, в них собраны уже опубликованные книги, представленные в различных форматах файлов. Для удобства пользователей содержимое ресурса классифицировано по рубрикам, часто имеется возможность

поиска по автору, году издания, ключевым словам. Поскольку содержание книги в печатном и электронном вариантах совпадает, то эту информацию вполне можно считать достоверной. Однако при составлении библиографического списка использованную книгу необходимо обозначить как электронный ресурс и обязательно указать режим доступа.

Помимо электронных библиотек огромные массивы данных содержат разнообразные интернет-энциклопедии. Рассмотрим «Википедию», «свободную энциклопедию, которую может редактировать каждый» (http://ru.wikipedia.org/wiki/Заглавная_страница). На момент написания данной статьи в «Википедии» было размещено 215256 статей на русском языке, и с каждым днем это число увеличивается.

Нас, прежде всего, интересует математический портал «Википедии»: <http://ru.wikipedia.org/wiki/Портал:Математика>. Здесь статьи разделены на категории «Введение» и «Ключевые статьи по разделам математики». Важной особенностью «Википедии» является возможность создавать и редактировать существующие статьи. С одной стороны, для студентов написание статей для «Википедии» может стать первым шагом к научно-исследовательской деятельности, но с другой – свобода размещения информации вызывает опасения, что в энциклопедию могут попасть ложные сведения и непроверенные факты.

Для иллюстрации степени достоверности информации, размещаемой в «Википедии», приведем выдержку из ее правил:

«Википедия – не место для публикации оригинальных исследований (таких, например, как “новые” теории). Для этой цели больше подходит Викитека (<http://ru.wikipedia.org/wiki/Викитека>).

Такая позиция Википедии – это не попытка ущемить чьи-то права и интересы. Это просто констатация факта: Википедия – не первичный, а вторичный источник информации, то есть такой, который собирает, анализирует, оценивает, интерпретирует и синтезирует информацию из первичных источников. В Википедии можно писать оригинальные обзоры и обобщения, но не новые заявления, утверждения или выводы.

...Заслуживающим доверие источником информации может служить известный научный рецензируемый журнал, академические издания. Самиздат и книги, выпущенные на средства автора, такими источниками не считаются» (http://ru.wikipedia.org/wiki/Википедия:Об_оригинальных_исследованиях).

Таким образом, поскольку не допускается публикация непроверяемых сведений, в качестве источника информации «Википедию» можно считать вполне приемлемой.

В отношении остальных интернет-энциклопедий вывод о достоверности содержащейся в них информации можно сделать, ознакомившись с правилами публикации статей. Также важным фактором является наличие раздела правил, в котором оговариваются авторские права и условия использования материалов энциклопедии. Чем меньше требований и ограничений накладывается на статьи, тем выше вероятность получить непроверенные сведения.

Теперь обратим внимание на математические сайты, рекомендованные Федеральным порталом «Российское образование» (<http://www.edu.ru/>). Поиск производится в рубрике каталога (с учетом действующих фильтров). Рубрики соответствуют разделам математики (http://www.edu.ru/modules.php?op=modload&name=Web_Links&file=index&l_op=viewlink&cid=1314). Результатом поиска является ссылка на ресурс, занесенный в базу данных, с краткой аннотацией его содержания. В роли такого ресурса может выступать сайт библиотеки или университета, электронный учебник, портал и тому подобное.

Особо отметим портал <http://ega-math.narod.ru/>. Автором собраны книги, статьи и заметки об интересных задачах из различных разделов математики, а также о жизни и творчестве выдающихся ученых-математиков. Очень полезным для начинающих авторов математических статей будет очерк П. Р. Халмша «Как писать математические тексты» (<http://ega-math.narod.ru/Halmos.htm>).

Большой популярностью среди студентов-математиков пользуется сайт www.exponenta.ru. В разделах, посвященных математическим пакетам Mathcad, Matlab, Mathematica, Maple, Statistica, находятся электронные учебники, справочники и статьи. С примерами применения математических пакетов в образовательном процессе можно ознакомиться в разделе «Методические разработки». Кроме того, с сайта можно скачать демо-версии популярных математических пакетов, электронные книги и свободно распространяемые программы. Exponenta.ru часто проводит конкурсы студенческих научных работ и методических разработок преподавателей. Подробную информацию о текущих конкурсах можно узнать здесь: http://www.exponenta.ru/educat/competit/competit_konk.asp#2

Веб-сайт <http://eqworld.ipmnet.ru/> содержит обширную информацию о различных классах обыкновенных дифференциальных уравнений, дифференциальных уравнений с частными производными, интегральных, функциональных и других уравнений. Основная цель создания веб-сайта EqWorld – помочь максимально широкому кругу научных работников, преподавателей вузов, инженеров, аспирантов и студентов луч-

ше ориентироваться в многочисленных уравнениях из различных разделов математики и их решениях (<http://eqworld.ipmnet.ru/ru/about/features.htm>).

По адресу <http://www.math.ru/> находятся книги, видеолекции, занимательные математические факты, различные по уровню и тематике задачи, истории из жизни математиков – «все то, что поможет окунуться в удивительный и увлекательный мир математики» (<http://www.math.ru/>). Учредителями портала Math.Ru являются отделение математических наук Российской академии наук и Московский центр непрерывного математического образования.

В современных условиях многие академические научные журналы имеют собственные страницы в Интернете. Как правило, такие ресурсы содержат информацию о журнале, условия и стоимость подписки, архив номеров.

Научно-популярный физико-математический журнал «Квант» издается с января 1970 г. Сейчас старые номера журнала «Квант» практически недоступны читателям. Очень ограниченное число библиотек имеет полное собрание вышедших журналов. Сайт <http://kvant.mcsme.ru/> призван открыть путь к богатому архиву журнала с 1970 по 2003 г.

Сибирский математический журнал (СМЖ) – научно-теоретическое периодическое издание, выпускается с 1960 г. Журнал публикует научные статьи по всем основным разделам математики (функциональный анализ, дифференциальные уравнения, алгебра и логика, геометрия и топология, теория вероятностей и математическая статистика, теория условно-корректных задач математической физики, вычислительные методы линейной алгебры и др.) Каждый годовой том содержит 6 выпусков по 240 страниц. С 1993 г. издается Институтом математики им. С. Л. Соболева Сибирского отделения РАН. Журнал выходит на русском языке, переводится на английский язык и печатается в США. На сайте <http://a-server.math.nsc.ru/publishing/smj/index.php> доступны архивы номеров журнала с 1993 по 2008 г.

Информационная система Math-Net.Ru (<http://www.mathnet.ru/>) – это общероссийский математический портал, предоставляющий российским и зарубежным математикам различные возможности в поиске информации о математической жизни в России. Данный сайт может быть полезен довольно широкой аудитории: студентам, аспирантам, математикам-профессионалам, исследователям.

На главной странице приведен небольшой список известных российских математических журналов: Дискретная математика (<http://dma.mi.ras.ru/>), Известия Российской академии наук. Серия мате-

матическая (<http://www.mathnet.ru/izv>), Математические заметки (<http://www.mathnet.ru/mz>), Математические труды (<http://www.springerlink.com/content/1055-1344>), Математический сборник (<http://www.mathnet.ru/msb>), Современная математика. Фундаментальные направления (<http://www.mathnet.ru/cmfd>), Современные проблемы математики (<http://www.mi.ras.ru/spm>), Фундаментальная и прикладная математика (<http://mech.math.msu.su/fpm>), Функциональный анализ и его приложения (<http://www.mi.ras.ru/~faa>), Lobachevskii Journal of Mathematics (<http://ljm.ksu.ru>) и др. Для каждого журнала приведено краткое описание научных направлений, состав редакционной коллегии, указана контактная информация, имеется возможность поиска статьи по различным параметрам. Важным преимуществом статей, размещенных в системе Math-Net.Ru, является возможность скачать их полный текст в формате pdf-файла.

Один из разделов электронной библиотеки Математического института им. В. А. Стеклова РАН называется «Математические ресурсы России» (http://libserv.mi.ras.ru/res_main.html), в Поморском государственном университете создана информационная система «Русские математические ресурсы «Internet»» (<http://www.pomorsu.ru/Departments/Math/math/russian.html>).

Работы в этом направлении ведутся также в рамках интеграционных проектов, объединяющих несколько организаций математического профиля. Одним из таких проектов является междисциплинарный интеграционный проект СО РАН «Древовидный каталог математических Интернет-ресурсов» [3]. Работа над проектом начата в 2003 г. Результатом явилось создание информационной системы «MathTree – каталог математических Интернет ресурсов» (<http://www.mathtree.ru/>). Одним из основных отличий данной коллекции от других каталогов является то, что каталог пополняется и модерируется ведущими специалистами в различных областях математики, что гарантирует определенную тщательность отбора ресурсов. На сайте представлен древовидный каталог математических ресурсов, в котором все разделы математики разбиты на 7 больших ветвей. Каждая из ветвей разделяется на несколько менее широких ветвей и так далее. На разных уровнях каталога хранятся данные, связанные с конкретным разделом математики. За основу систематизации математических ресурсов взята международная классификация разделов математики, принятая в реферативных журналах *Mathematical Reviews* и *Zentralblatt*. В них для разделения статей по темам используются математическая предметная классификация (*Mathematics Subject Classification*) и универсальная десятичная классификация (УДК) [4]. В на-

стоящее время в каталоге находится: ресурсов – 4759, разделов – 274.

К сожалению, далеко не всегда удастся получить доступ к полному тексту статьи. В таких случаях результат, описанный в статье, представлен в виде короткой аннотации в 1–2 предложениях. К примеру, по такой схеме организован доступ к содержанию журнала «Математические труды», издаваемого Институтом математики им. С. Л. Соболева СО РАН. Тем не менее для исследователя часто бывает достаточно аннотации, содержащей основной результат, чтобы возникли новые вопросы, появились новые мысли относительно методов исследования.

Часто бывает так: чтобы получить ответ на интересующий вопрос, студенты обращаются на форумы, например, такие, как <http://allmatematika.ru/>. Безусловным достоинством подобных ресурсов является возможность общения и обмена опытом между начинающими и опытными математиками-исследователями. Однако есть риск получить неправильный или неполный ответ. Именно поэтому к сведениям, полученным на разнообразных форумах, необходимо относиться очень критично и подвергать тщательной проверке каждый факт, который приводит автор сообщения. (Соответствующую иллюстрацию мы привели выше.)

Очень солидную подборку ссылок на различные ресурсы, посвященные математике, содержит проект «Избранные математические сайты» <http://www.imath.kiev.ua/~golub/rmsites.html>.

В рамках одной статьи невозможно проанализировать все существующие источники, к тому же их количество увеличивается с каждым днем. Мы ограничились рассмотрением ресурсов, которые находятся в первой двадцатке результатов поиска по запросу «математический сайт» в популярных русскоязычных поисковых системах: nigma.ru, www.yandex.ru, www.rambler.ru, www.google.ru, www.gogo.ru.

Часть ресурсов глобальной сети содержит полезную, достоверную и интересную информацию, которая может быть использована в учебном процессе и исследовательской деятельности студентов. Кроме того, поиск в сети Интернет можно использовать для повышения мотивации у обучаемых. При анализе содержания различных математических сайтов у студента сформировывается осознание того, что развитие математики не остановилось, а динамично продолжается, и

классические теории находят все новые обобщения и приложения в современной жизни.

Например, при изучении курса дифференциального исчисления студентам может быть дано задание: выяснить актуальные проблемы данного раздела анализа. Для его выполнения потребуется обращаться не к классическим учебникам математического анализа, а к статьям в академических научных, научно-популярных, научно-методических журналах и информации на специализированных сайтах. Навыки поиска и анализа нового материала важны для будущих специалистов, а особенно – для будущих учителей математики. В период информатизации образования в России перед учителем стоит задача адаптироваться к новым условиям, уметь грамотно использовать средства информационных технологий в обучении, научить школьников самостоятельно искать, оценивать и использовать информацию по открытым проблемам современной математики. Успешное решение указанной задачи возможно только при условии, что учитель сам имеет соответствующую подготовку.

Особо значимым и ценным является понимание возможности сделать вклад в развитие современной математики. Таким образом, отношение к информации, размещенной в глобальной сети, не должно быть потребительским, а, напротив, открытые проблемы математики, решением которых занимаются пользователи Интернета, должны стимулировать у студентов интерес к научно-исследовательской работе.

Примечания

1. Петрова, А. И. Внедрение коммуникативных личностно-ориентированных образовательных технологий в учебный процесс [Текст] / А. И. Петрова // Математика в образовании: сб. статей. Вып. 3 / под ред. И. С. Емельяновой. Чебоксары: Изд-во Чуваш. ун-та, 2007. С. 96–101.

2. Там же.

3. Клименко, О. А. MathTree – каталог математических Интернет-ресурсов [Текст] / О. А. Клименко, В. Э. Филиппов // Материалы XIII конференции представителей региональных научно-образовательных сетей “RELARN-2006”. Барнаул, 2006. С. 31–35.

4. Де Роббио, Антонелла. Математическая предметная классификация (МПК) ATION и похожие классификационные системы в мире электронной информации [Электронный ресурс] / Антонелла Де Роббио, Дарио Магуоло, Альберто Марини: ГПНТБ России. Код доступа: <http://www.gpntb.ru/win/inter-events/crimea2001/tom/sec4/Doc23.HTML>

М. В. Таранова

ДВА ПОДХОДА
В ОРГАНИЗАЦИИ ПОЗНАНИЯ

В статье рассматриваются два подхода к проблеме методов познания. Ставится вопрос об использовании системно-структурных средств обучения.

Система образования сегодня заметно отличается от той, которая казалась незыблемой 15–20 лет назад. Наметились существенные изменения, которые приняли характер устойчивых тенденций. К одной из них можно отнести профиллизацию образования, основная цель которой заключается в ориентации воспитания и развития каждого обучаемого сообразно его интересам и возможностям: как человека творческого, как носителя и продолжателя науки и культуры. В связи с этим возникает вопрос о том, какими средствами познания необходимо вооружить обучаемого, чтобы в последующей познавательной деятельности он бы мог осуществлять самостоятельную мыслительную деятельность. Обобщая различные подходы к вышеобозначенной проблеме, мы выявили, что в теории методов познания существует два принципиально разных подхода к организации исследования. Одно из направлений называют «объектно-натуралистическим», а другое – методологическим, или теоретико-мыслительным, или эпистемологическим в узком смысле этого слова [1]. В первом варианте направленность исследования определяют с точки зрения того объекта, на который направлена деятельность исследователя или проектировщика. «Всякий исследователь, принимающий натуралистический подход, независимо от того, в какой науке он работает, исходит из того, что ему уже дан объект рассмотрения, что он сам как исследователь противостоит этому объекту и применяет к нему определенный набор процедур и операций, которые дают ему, исследователю, “знания об объекте”» [2]. Исследователь-натуралист никогда не задает вопросов, откуда взялся «объект» и каким образом он до момента исследования получился, поскольку для него природа самого начала состоит из объектов, или, точнее, из объектов созерцания, которые и становятся затем объектами специального научного исследования: «Главный недостаток всего предшествующего материализма – включая и фейербаховский – заключается в том, что предмет, действительность, чувственность берется только в фор-

ме объекта, или в форме созерцания, а не как человеческая чувственная деятельность, практика, не субъективно», – писал в своих работах К. Маркс [3]. То же самое происходит и с учеником, когда ему предлагается «решить задачу», «решить задачу с параметром», или «исследовать решение заданного уравнения или неравенства». Натуралистически организованное сознание ученика не замечает сложнейших структур мышления и деятельности и того обстоятельства, что объект мыслительной деятельности включен в нее и является функциональным и морфологическим элементом этой деятельности. Сознание юного исследователя в этой ситуации, и это в лучшем случае, видит только два морфологических аспекта деятельности: ее объект и субъект, сознание же различает и разделяет только эти два фокуса, между ними проводит границу, сводит все свои мысли к объекту исследования, а затем полагает между объектом и субъектом определенное отношение, связь особого рода – познавательно-исследовательскую. К примеру, учащемуся часто очень сложно решать задачи с параметром, в каждом отдельном случае перед ним объект как данность, совсем ему не знакомый, или знакомый отдаленно, который необходимо исследовать.

В методологическом же варианте направленность исследования определяется не по объекту, который осваивается деятельностью и мышлением, а по специфике самих процедур деятельности и мышления [4]. То есть подход к исследованию характеризуется не извне косвенно, не типом объекта, на который он направлен, а изнутри и непосредственно. Если, скажем, мы будем характеризовать такой подход в рамках теории мышления, то он выступит как особый стиль и способ мышления, особый аппарат мыслительной работы. Если же мы будем характеризовать подход в рамках теории научного исследования, то он выступит как особая система средств и методов научного исследования и т. д. Деятельностный, или системодетельностный, подход в создании организационной структуры мышления исходит не из оппозиции «субъект – объект», или, в более специфических терминах, «исследователь – исследуемый объект», а из самих систем деятельности и мышления, из тех средств и методов, той техники и технологии, тех процедур и операций, которые составляют структуру исследовательской мыслительности и задают основные формы ее организации [5].

Различие двух подходов к организации работы человеческого сознания не означает, что эти две точки зрения уместно сравнивать в терминах «лучше – хуже» или в терминах «более значима – менее значима», каждый из подходов имеет законное право на существование. Натурали-

ТАРАНОВА Марина Владимировна – кандидат педагогических наук, доцент по кафедре высшей математики Новосибирского государственного педагогического университета
© Таранова М. В., 2008

стический подход логически обоснован практикой его использования на протяжении уже более чем четырех столетий, и именно ему наука обязана своими основными успехами. Исследователь-«натуралист», при обилии входящих в сознание разнообразных элементов, в предметно-теоретической форме фиксирует только объект исследования, его сознание сосредоточено только на исследуемом объекте, только его он замечает и видит – и в этом величайшая простота и сила натуралистического подхода, его бесспорное практическое преимущество. Однако при всей простоте и очевидности натуралистического подхода в организации сознания существует ряд проблем, которые этим же подходом и порождаются. К примеру, так называемая проблема «правильного» истолкования понятий, поиска «подлинного смысла» отдельных высказываний и учений в герменевтике. Теоретическая герменевтика – искусство и теория истолкования текстов – выдвинула ряд проблем, принципов, идей, которые характеризуют процесс понимания, наиболее важной из них является идея *герменевтического круга* [6]. «Герменевтический круг возникает в результате столкновения двух тенденций. С одной стороны, для понимания и правильного истолкования текста, рассуждения или теории нужно сначала понять отдельные элементы, из которых складывается целое. Но, с другой стороны, только целостное представление обеспечивает понимание подлинного смысла отдельных частей и поэтому понимание целого должно предшествовать пониманию деталей», – отмечает в своих исследованиях В. В. Мадер [7]. Действительно, сущность отдельного элемента заключается не в нем самом, а проявляется в той роли, которую этот элемент играет в объемлющей его системе. То есть понимание части предполагает понимание целого, а понимание целого само основано на понимании его частей. Что является первичным – отдельные первичные понятия, которые рассматриваются сами по себе, или структура целого? Предположение же того, что основой построения должны быть первичные понятия, не очень приемлемо, хотя бы потому, что смысл первичных понятий определяется соответствующей системой знаний. Элементы этой системы сами по себе никаким наличным бытием не обладают, у них нет каких-либо внутренних свойств, которые бы характеризовали их внутренние свойства. На самом деле свойства предметов – это всего лишь проявление их взаимодействия с другими предметами. Таким образом, определить понятия и свойства предметов самих по себе не представляется возможным. К этому пониманию пришли многие ученые и философы. Психология творчества давно осмыслила и при-

няла на вооружение вывод о том, что продуктивное, творческое мышление начинается не с деталей, а с внезапного усмотрения общего принципа, общей идеи, возникающей в форме «озарения». Ж. Пиаже, отмечал, что «основой общего здания математики долгое время считались некоторые объекты, рассматриваемые... изолированно друг от друга [8]. «...Однако, – продолжает Ж. Пиаже, – на самом деле эти объекты являются лишь элементами объемлющей системы, вне которой эти элементы не имеют ни значения, ни существования» [9]. Такое же понимание мы находим в работах В. В. Давыдова. Он подчеркивал, что необходимость усмотрения структуры целого является одним из проявлений принципа восхождения от общего к частному.

Содержание системодетельностного подхода показывает, что использование логики *его* идей в обучении оказывает наиболее эффективное воздействие на воспитание мышления будущего проектировщика, конструктора. По существу, умение решать ту или иную исследовательскую задачу – это часть той культуры мышления, которая присуща исследователю, конструктору. Поэтому-то перед учителем профильного физико-математического класса, прежде всего, должна стоять задача воспитания особого стиля мышления, которое выражено в пытливости ума, в целенаправленном выборе средств мыслительной деятельности школьника. Понимание целей обучения и воспитания как планированное обучение средствами мыслительной деятельности предполагает необходимость выделения тех средств, которые были бы целесообразными и наиболее оптимальными для решения учебных и исследовательских задач.

Примечания

1. *Щедровицкий, Г. П.* Избранные труды [Текст] / Г. П. Щедровицкий. М.: Шк. культ. полит., 1995. С. 71.
2. Там же. С. 144.
3. *Маркс, К.* Тезисы о Фейербахе [Текст] / К. Маркс // Маркс К., Энгельс Ф. Соч. 2-е изд. Т. 3. М., 1955.
4. *Щедровицкий, Г. П.* Указ. соч. С. 72.
5. Там же. С. 145.
6. *Мадер, В. В.* Герменевтика: История и современность [Текст] / В. В. Мадер. М.: Мысль, 1985; *Мадер, В. В.* Введение в методологию математики (Гносеологические, методологические и мировоззренческие аспекты математики. Математика и теория познания) [Текст] / В. В. Мадер. М.: Интерпракс, 1995. 464 с.
7. *Мадер, В. В.* Герменевтика: История и современность. С. 17.
8. *Пиаже, Ж.* Структуры математические и операторные структуры мышления [Текст] / Ж. Пиаже // Преподавание математики. М.: Учпедгиз, 1960. С. 11.
9. Там же. С. 13.

ЯЗЫК

Т. Б. Агалакова, Ю. В. Казаковцева

КОНЦЕПТ GOOD/ДОБРО: ОБЩЕЕ И СПЕЦИФИЧЕСКОЕ

В статье представлены результаты анализа структуры концепта «добро», вербализованного в английском языке словом *good*, а в русском языке – словом *добро*; общие и специфические признаки концепта.

Современный этап в развитии российской лингвистической мысли характеризуется повышенным интересом к исследованиям в области когнитивной лингвистики и, в частности, ее направления – лингвистической концептологии. В задачи последней входит установление состава, корпуса, репертуара фундаментальных национально-культурных единиц – концептов. «Концепт – это оперативная единица памяти, ментального лексикона, концептуальной системы и языка мозга, всей картины мира, квант знания. Самые важные концепты выражены в языке» [1]. Вербализованными оказываются те из них, которые фиксируются в национальной концептосфере и создают национальную картину мира.

Понятия концептосферы и национальной картины мира являются важнейшими в когнитивной лингвистике. «Концептосфера – это чисто мыслительная сфера, состоящая из концептов, существующих в виде мыслительных картинок, схем, понятий, фреймов, сценариев, гештальтов... наиболее абстрактных сущностей, обобщенных разнобразными признаками внешнего мира» [2]. Часть концептосферы, получившая выражение в системе языковых знаков – слов, фразеологизмов, синтаксических структур, является семантическим пространством языка.

Структура концепта может быть представлена в виде поля. Под лексико-семантическим полем (далее ЛСП) в системе языка понимается совокупность семантически соотносительных классов слов разных частей речи [3]. В качестве предмета нашего исследования мы выбрали ЛСП

good/добро. Было установлено, что количественный состав поля в обоих языках достаточно объемный. Сюда входят лексемы с разными грамматическими и лексическими характеристиками, с разной частотностью употребления и валентностью. Особое место в парадигме ЛСП занимают имена существительные и прилагательные. Они позволяют выражать дополнительные оттенки в значении, связанные с оценочностью и экспрессией.

Все отобранные лексемы можно условно распределить по следующим группам:

1) слова, характеризующие нравственно-этические чувства (доброта, милосердие, сострадание, добросердечный, добропорядочный, *virtuous*, *polite*, *compassion* и т. п.);

2) слова, выражающие оценку качества или действия (добрый, хороший, одобрить, *kind*, *care* и т. п.);

3) слова, характеризующие материальную сторону действительности (добротный, качественный, удобный, *satisfactory* и т. п.);

4) слова, описывающие эмоционально-позитивные переживания (счастье, любовь, *love*, *happiness*).

Анализ фактического материала показал, что как для носителей русского языка, так и английского языка *добро* предстает, в первую очередь, в виде лучших человеческих качеств. Несмотря на субъективность в толковании данного понятия, можно восстановить некий эталон доброго, положительного человека. Доминантным является признак «высокая нравственность, духовность».

Обращает на себя внимание выделение представителями английской лингвокультуры таких признаков, как *pleasant* «приятный, доставляющий удовольствие»; *enjoyable*, *amusing* «доставляющий удовольствие, наслаждение». Носители русского языка выделяют признаки «радость», «нежность», психологически близкие признакам «счастье», «любовь», «*happiness*», «*love*».

Для носителей английского языка одной из характеристик добра является божественное добро. Бог выступает как воплощение принципов добра и означает высшую добродетель (*Good God! Good Heavens*).

Примером, отражающим специфику ЛСП «добро» в русском языке, могут служить стилистически маркированные лексемы, английские эквиваленты которых не сопровождаются соот-

АГАЛАКОВА Татьяна Борисовна – кандидат филологических наук, доцент по кафедре германских языков ВятГУ

КАЗАКОВЦЕВА Ю. В. – студентка V курса факультета информатики ВятГУ

© Агалакова Т. Б., Казаковцева Ю. В., 2008

ветствующими пометами. Среди них – слова, относящиеся к сниженной или разговорной лексике: добряк, душа-человек, славный, недурной и др. Нам также встретились примеры устаревших слов с пометой «возвышенное», «книжное», например: добродетель, благой, добросердечие, доброжелатель; фразеологизмы типа золотое сердце, ангельская душа, святой человек.

Таким образом, можно констатировать, что концепт «добро», будучи универсальным, имеет наряду с общими признаками и национально-специфические.

Примечания

1. Кубрякова, Е. С. Концепт [Текст] / Е. С. Кубрякова, В. З. Демьянков, Ю. Г. Панкратц, Л. Г. Лузина // Краткий словарь когнитивных терминов. М., 1996. С. 90–94.

2. Попова, З. Д. Очерки по когнитивной лингвистике [Текст] / З. Д. Попова, И. А. Стернин. Воронеж, 2001.

3. Васильев, Л. М. Теория семантических полей [Текст] / Л. М. Васильев // Вопросы языкознания. 1971. № 5. С. 104.

Т. Б. Агалакова, А. М. Шиляева

ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ ПОРТФОЛИО В ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

В статье представлены результаты опытно-экспериментальной работы по использованию языкового портфолио в организации самостоятельного индивидуального чтения студентов I, II и III курсов специальности «Информатика с доп. спец. иностранный язык».

На современном этапе перед учебными заведениями поставлена задача формирования у обучающихся потребности к самообразованию. Современный студент, будущий специалист должен уметь организовать свою познавательную деятельность и быть способным к постоянному профессиональному совершенствованию. Следовательно, в процессе обучения в вузе должны не только закладываться основы профессиональной деятельности обучающихся, но и формироваться навыки самообразовательной работы.

В действующих в российских вузах учебных планах количество часов, отведенное на изучение дисциплины, распределяется поровну между аудиторной и внеаудиторной работой. Переход

российских вузов к двухуровневой системе подготовки специалистов, а также появление такой формы организации учебного процесса, как дистанционное обучение, ведет к повышению удельного веса самостоятельной работы студентов. В этих условиях перед преподавателем стоит ответственная задача – организовать активную работу студентов для достижения поставленной цели.

Существуют разные способы организации самостоятельной работы обучающихся. В последние годы большим успехом как у школьных учителей, так и у преподавателей вузов пользуется технология портфолио. В университетском образовании данная технология служит способом отслеживания профессионально-личностного становления студентов, объективации их познавательного опыта и организации самостоятельной научной работы. Это особый, оригинальный путь сбора, хранения и обработки информации. Направления использования портфолио в вузе многообразны. Это могут быть портфели конференций, портфели научной работы, портфели личных достижений, портфели индивидуального творчества и др. [1]

Данная технология не только помогает организовать самостоятельную работу студентов и создает условия для активного участия в процессе построения знаний, но и дает возможность каждому студенту сохранить свой стиль деятельности, интегрируя опыт самостоятельной работы с опытом аудиторных занятий [2]. К достоинствам использования портфолио также следует отнести следующие факторы:

- 1) всесторонняя подготовка обучающегося к непрерывному процессу образования, саморазвития и самосовершенствования;
- 2) осознание личной ответственности за результаты обучения;
- 3) повышение мотивации в дальнейшем совершенствовании приобретенных навыков;
- 4) формирование способности объективной самооценки [3];
- 5) возможность фиксировать изменение и рост обучающихся за определенный период времени [4].

Одним из достоинств портфолио, по мнению Е. О. Галицких, является и то, что «информация в портфеле подвижна, легко переконструируется; соответственно меняются названия файлов. Портфель можно периодически «чистить», освобождая от лишних документов, или кодировать, сжимать информацию в модель, схему, вывод, перспективу» [5].

Вместе с наличием объективных преимуществ использование технологии портфолио сопровождается и рядом трудностей, а именно: психологическая неподготовленность обучающихся пе-

АГАЛАКОВА Татьяна Борисовна – кандидат филологических наук, доцент по кафедре германских языков ВятГУ

ШИЛЯЕВА Анна Михайловна – старший преподаватель кафедры германских языков ВятГУ
© Агалакова Т. Б., Шиляева А. М., 2008

рейти от привычной работы с тетрадями к новому способу организации материала в папке, большие затраты времени на данный вид работы, несформированное умение провести рефлексию собственного труда [6]. По мнению Е. С. Полат, умению адекватно оценивать собственные достижения и возможности «необходимо так же учить детей, как мы учим их знаниям, умениям, навыкам, самостоятельному мышлению» [7].

Технология портфолио активно используется в практике обучения иностранным языкам. На занятиях по языку наибольшее распространение получили *Демонстрационный портфель* (Show Case) и *Обучающий портфель* (Language Learning Portfolio). Первый включает образцы лучших самостоятельных работ обучающегося, выполненные в течение определенного периода времени, и демонстрирует степень сформированности умений в различных видах иноязычного общения. Второй вид портфеля содержит материалы и рекомендации для самостоятельной работы обучающихся, средства самостоятельной диагностики и оценки владения языком [8].

В данной статье нам хотелось бы поделиться опытом использования технологии портфолио в организации самостоятельного индивидуального чтения студентов. В течение семестра студенты должны были прочитать одно или два художественных произведения на английском языке. Результатом самостоятельной работы стало оформление портфеля читателя (Reader's Portfolio), который включал следующий материал (files):

1) титульная страница, где указываются имя студента, факультет, группа, дисциплина, автор и название книги;

2) инструкции по оформлению и ведению портфолио, список рекомендуемой литературы по жанрам;

3) сведения об авторе произведения;

4) характеристика главных героев (имя, возраст, внешность, образование, семейное положение, работа, черты характера, взаимоотношения с другими героями книги);

5) краткое содержание прочитанного;

6) новая лексика, примеры ее употребления в тексте и собственные примеры;

7) отрывок из книги (12 строк) и его литературный перевод на русский язык;

8) описание студента себя как читателя, впечатления от прочитанной книги;

9) краткие пояснения студента о том, что получилось, а что нет; какие выводы он может сде-

лать из результатов проделанной работы (рефлексия).

Подводя итоги работы, студенты отмечают определенные трудности, с которыми они столкнулись в ходе оформления портфолио. По их мнению, это «очень кропотливая работа», требующая умственного напряжения, усидчивости, терпения и, наконец, «уйму времени». Одновременно это и хорошая практика письма и составления пересказа прочитанного. Выражение собственных мыслей, критическая оценка содержания произведения помогли студентам глубже его понять. Прочитанные и проанализированные в портфолио книги «особенно ценны» для них, так как стали не просто развлечением, но и новым опытом, почвой для размышлений, основой для учебной деятельности. По словам студентов, система портфолио «заставила» их работать в полную силу.

В целом опытно-экспериментальная работа показала, что данная технология стимулирует активность и самостоятельность студентов, способствует развитию творческих способностей и критического мышления, совершенствует навыки владения иностранным языком.

Примечания

1. *Гальскова, Н. Д.* Российский языковой портфель для начальной школы [Текст] : руководство для учителей и родителей / Н. Д. Гальскова, З. Н. Никитенко. М.: МГЛУ, 2003. С. 115.

2. *Макарова, Е. Е.* Использование технологии портфолио при организации самостоятельной работы студентов [Текст] / Е. Е. Макарова // Совершенствование преподавания иностранных языков в школе и вузе. Вып. 12. Киров: Изд-во ВятГГУ. 2007. С. 63.

3. *Иванова, Н. В.* Языковой портфель как инструмент формирования социо-, кросскультурной компетенции учащегося [Электронный ресурс] / Н. В. Иванова. Режим доступа: http://portal.krsnet.ru/razdels/uchitelja/rmo/metod/ang/yaz_portf.htm. С. 4.

4. *Федотова, Е. Г.* Портфолио как система альтернативного оценивания в практике зарубежной школы [Текст] / Е. Г. Федотова // Школьные технологии. 2005. № 3. С. 151.

5. *Гальскова, Н. Д.* Указ. соч. С. 115.

6. *Сычева, А. В.* Языковой портфель для филологов [Текст] / А. В. Сычева // Классическое лингвистическое образование в современном мультикультурном пространстве: м-лы междунар. науч. конф. 22–23 апреля 2004 г. М.; Пятигорск, 2004. С. 402.

7. *Полат, Е. С.* Портфель ученика [Текст] / Е. С. Полат // Иностранные языки в школе. 2000. № 5. С. 5.

8. *Щукин, А. Н.* Обучение иностранным языкам: теория и практика [Текст] : учебное пособие для преподавателей и студентов / А. Н. Щукин. М.: Филоматис, 2004. С. 267–268.

А. И. Бардовская

НЕКОТОРЫЕ ЗАКОНОМЕРНОСТИ НОМИНАЦИИ СИНЕСТЕТИЧЕСКИХ ОЩУЩЕНИЙ В ПРОИЗВЕДЕНИЯХ РУССКИХ ПИСАТЕЛЕЙ

В статье обсуждаются различные современные теории синестезии и на примере анализа ряда синестетических образований, свойственных художественному тексту, показывается целесообразность их комплексного рассмотрения при изучении синестетической метафоры.

Одной из актуальных проблем современного лингвистического исследования является проблема *синестетической метафоры*, определяемой, в самом общем виде, как использование слов, связанных с одной сферой сенсориума, для описания ощущений и восприятий, связанных с другой сферой чувств. Возврат к антропоцентризму, при котором человек рассматривается как главный фактор функционирования языка, обуславливает развитие представлений об интересующем нас явлении как не только существенным образом характеризующем язык, но и имеющем отношение к развитию и функционированию мышления. Таким образом, в центре внимания оказывается вопрос о синестетической метафоре как универсалии не только собственно лингвистической, но и лингвопсихической, лингвоментальной, изучение которой требует выхода за пределы замкнутой языковой системы и поиска интеграции данных различных наук о природе интермодальности. С другой стороны, признается, что наблюдения за случаями проявлений синестезии в речи представляют несомненную ценность в поисках закономерностей психофизиологического механизма взаимодействия органов чувств и способны пролить свет на его природу [1].

Необходимо подчеркнуть очевидную ценность материала художественной практики как плодотворной почвы изучения интересующего нас явления: как характерное свойство человеческой психики синестезия ярче всего проявляется при восприятии искусства (отметим, что об «эстетической» природе синестезии говорит сама этимология этого термина). Об этом писал еще А. Белый, полагавший, что отсутствие цветного слуха в художнике пера и кисти – изъян. Обосновывая потенциал материала художественной практики в изучении синестезии, Б. М. Галеев отмечает, что искусство как форма художественной коммуникации обращается, прежде всего, к

тем синестезиям, которые обладают определенной общезначимостью [2]. Механизмы синестезии в этом случае позволяют говорить о разных способах представления знаний, помогают понять, почему одни и те же значения могут реализовываться в графических, вербальных, музыкальных, пластических формах. По мысли В. Ф. Петренко, «это уровень глубинной семантики, отражающий те когнитивные структуры, на языке которых “говорят” метафора, аналогия, поэзия» [3]. Язык художественной литературы предоставляет в этом отношении богатые возможности для изучения закономерностей и природы синестезии.

В нашем исследовании [4] мы исходили из предположения об универсальном характере синестетической метафоры и направлений кроссмодальных переносов, обусловленных общезначимыми механизмами взаимодействия органов чувств, что, в целом, нашло подтверждение при анализе 700 основанных на синестезии языковых номинаций, собранных из произведений более 20 английских и русских писателей (Т. Hardy, R. L. Stevenson, O. Wilde, H. Wells, C. Dickens, М. Горького, И. Бунина, В. Набокова, М. Булгакова, И. Шмелева и ряда других). Вместе с тем нельзя не обратить внимание и на некоторые различия в моделях синестетического ассоциирования у англоязычных и русскоязычных писателей, а также на специфику синестезии применительно к идиостилю отдельных авторов. Следовательно, обращаясь к проблеме основанных на синестезии языковых выражений, есть основания говорить о вероятной значимости в их образовании и действии не только универсальных процессов, являющихся принадлежностью человека как вида *Homo Sapiens* и протекающих на уровне организма, но и ассоциативных «штампов», обусловленных конкретной культурой (на что указывают, в частности, ученые, занимающиеся вопросами нейролингвистического программирования [5]), а также особенностей творческой деятельности индивида, способного «уловить» сигналы, предоставляемые ему органами чувств, служащие импульсом для создания новых, фантазийных межчувственных соответствий, находящихся языковое выражение.

Корпус исследованных синестетических метафор ограничивается номинациями с температурным компонентом значения. Выбирая именно эту группу межчувственных ассоциаций, мы обратили внимание на замечание ряда авторов (например, [6]) об особой значимости температурных перцептивных рядов в процессе коммуникации. Слово, несущее значение температурной модальности, обладает особым зарядом, оно способно вызвать у людей соответствующие ощущения и оказывать сильное воздействие. Изобилие «стер-

тых» синестетических метафор с температурным компонентом значения [7], частое использование температурных синестезий в специальных подъязыках искусствоведения [8], музыковедения [9], парфюмерии [10] дает возможность говорить о том, что привлечение температурных ассоциаций к описанию ощущений и восприятий различных модальностей отображает закономерности восприятия и воображения и заслуживает отдельного рассмотрения. Отметим также, что, хотя результаты проведенного исследования не идут в противоречие с выводами авторов, которые предлагали рассматривать синестетическую метафору как неделимое целое, без строгого выделения «синестезирующего» и «синестезируемого» элементов (например, [11]), нами все-таки предпринимается попытка определить роль температурной номинации в составе синестетического целого, что позволяет сделать некоторые выводы относительно значимости температурных ощущений в иерархии чувств и их коррелятов в сфере эмоций и ощущений других модальностей.

Итак, каковы особенности синестетических образований с температурным компонентом значения (далее – ТС), свойственных русскоязычному художественному тексту? Их специфика наблюдается уже при анализе грамматической структуры собранных ТС. Если процентное соотношение моделей русских и английских ТС примерно совпадает (наиболее частотна из них словосочетательная – более 50%, далее следуют случаи реализации синестезии на уровне предложения, слова и микроконтекста), то соотношение частей речи, представляющих температурную номинацию, существенно отличается. Ведущим способом передачи температурного ощущения как в английских, так и русских ТС является прилагательное, однако в первом случае их количество достигает 76%, а во втором 58%, далее следуют существительные – 18 и 28% соответственно, наречия – 2 и 11%. Совпадение здесь наблюдается лишь в одинаково незначительном количестве случаев выражения температурного ощущения причастиями (по 2%) и глаголами (2% и 1%). Убедительные выводы, касающиеся структурно-грамматического своеобразия ТС, свойственных русскоязычному художественному тексту, несомненно, требуют дальнейшего рассмотрения вопроса, сопоставления с другими разновидностями аналогичных явлений, однако даже столь беглый взгляд позволяет говорить об их разнообразии и о немалой значимости температурной ассоциации, получающей различные грамматические реализации.

Анализ содержательной стороны ТС показал, что наиболее часто температурные ассоциации привлекаются к описанию ощущений внешней, экстерорецептивной сферы чувств (что состави-

ло порядка 60% случаев как в англоязычном, так и русскоязычном материале). Это иллюстрирует закономерность, отмечавшуюся в работах [12] и объясняемую тем, что перенос значений чаще происходит от низших, более примитивных сфер сенсорiums к более дифференцируемым (зрению и слуху). Вместе с тем обращает на себя внимание подавляющее количество случаев совмещения номинаций температурных ощущений и ощущений и восприятий зрительной модальности в англоязычном материале (около 50% от общего количества собранных номинаций) и лишь около 30% аналогичных ассоциаций у русских писателей, в результате чего наиболее частотными моделями синестетического ассоциирования на материале русского языка становятся ТЕМПЕРАТУРА ↔ ЗРЕНИЕ и ТЕМПЕРАТУРА ↔ СЛУХ, а на материале английского – ТЕМПЕРАТУРА ↔ ЗРЕНИЕ и ТЕМПЕРАТУРА ↔ ОБОНЯНИЕ.

Ведущей субмодальностью зрительного восприятия в исследованных ТС оказывается цветное восприятие. Однако в русскоязычном материале подобная модель не может быть названа абсолютно доминирующей (у английских писателей число подобных метафор составляет более 50% от общего числа температурно-зрительных номинаций): количественно ей незначительно уступают описания цвето-световых и световых ощущений в температурных терминах. Здесь мы встречаем как простые синестезии (например, *теплая зелень; голубоватый леденящий свет; жарко блестеть*), так и комплексные, включающие в себя номинации ощущений и восприятий различных модальностей (например, *вода была сиреневого цвета, такого мягкого и теплого – ЦВЕТ ↔ ТЕМПЕРАТУРА ↔ ТАКТИЛЬНОСТЬ*). Присутствие номинации, соотносимой с субмодальностью «пластика», встречается редко как в рассмотренных английских, так и русских примерах, однако более типично для последних (например, *плавные, холодные линии*), в то время как субмодальности «форма» и «пространство» оказываются более характерными для материала английского языка и единичны среди русских ТС (*столбы зноя; холодная высота помещения*).

Группа интермодальных ассоциаций, описывающих собственно температурное ощущение, количественно не занимает ведущих позиций в собранных английских и русских ТС (их общее число составило по 10%), однако заслуживает отдельного интереса. По нашим наблюдениям, спектр периферийных ощущений в синестезиях такого типа весьма широк, что отчасти противоречит положению о примитивности, недифференцированности сферы температурного восприятия; кроме того, серьезные различия в наиболее час-

тотных моделях ассоциирования такого типа на исследованном материале английского и русского языков позволяют сделать предположение о возможной обусловленности их появления и функционирования факторами культуры. В русскоязычном художественном тексте наиболее часто (около 30% случаев) для описания температурных ощущений привлекаются номинации ощущений зрительной (например, *прозрачный холодок*) и болевой (например, *жалящий жар*) модальностей, характеризующиеся простой структурой (отметим, что в англоязычном материале подобные примеры мало распространены). Случаи комплексного привлечения разномодальных ассоциаций в такого типа русских ТС (комплексные синестетические образования, описывающие температурное ощущение, составляют большинство в корпусе собранных английских примеров – 40% случаев) несколько уступают первым двум по распространенности (например, *влажный, пахучий холодок* – ТАКТИЛЬНОСТЬ ↔ ОБОНЯНИЕ ↔ ТЕМПЕРАТУРА). Далее следуют незначительно представленные «эмоциональные» (например, *варварский мороз*), «тактильные» (например, *жесткий мороз*) и «слуховые» (например, *тихий холодок*) описания температурных ощущений (являющиеся, за исключением слуховых, весьма частотными в англоязычном материале) и единичные авторские случаи привлечения вкусовых (*холодок, подобный холодку мятной лепешки*), пластических (*неподвижный мороз*) и гравитационных (*день был легко-морозный*) ассоциаций.

Следует отметить, что, предпринимая попытку выделения моделей синестетического ассоциирования, нам зачастую приходилось несколько огрублять, упрощать собранный материал. Синестетические образования, свойственные художественному тексту, отличаются яркостью, эффектностью, эмоциональностью; они обладают сложным, комплексным характером, проявляющимся в случаях реализации синестезии на уровне синтаксических цепочек различной протяженности. Так, вряд ли стоит говорить о возможности разбиения на оптимальные с точки зрения выделения моделей синестетического переноса двучленные сочетания такого полимодального отрыва, как *«скука, холодная и нудная, дышит отовсюду: от земли, прикрытой грязным снегом, от серых сугробов на крышах, от мясного кифлича зданий; скука поднимается из труб серым дымом и ползет в серенькое, низкое, пыльное небо; скукой дымятся лошади, дышат люди. Она имеет свой запах – тяжелый и тупой запах пота, жира, конопляного масла, подовых тирогов и дыма; этот запах жмет голову, как теплая, тесная шапка, и, просачиваясь в грудь, вызывает странное опьянение, темное желание*

закрывать глаза, отчаянно заорать, и бежать куда-то, и удариться головой с разбега о первую стену» (М. Горький, В людях), в котором обладающий отрицательным эмоциональным зарядом синестетический образ *скуки, холодной и нудной*, сопровождается дополнительными разномодальными характеристиками, усиливающими эмоциональный эффект: обонятельными (*тяжелый и тупой запах*), болевыми (*жмет голову, просачивается в грудь*), двигательными (*бежать куда-то и удариться головой о первую стену*). «Всеобъемлющий» характер скуки, ее неизбежность подчеркивают пространственно-зрительные образы: *скукой дышат люди, она поднимается из труб серым дымом, скукой дымятся лошади*. Анализ эмоционального вектора подобных развернутых синестетических образований применительно к идиостилю ряда писателей позволил нам сделать вывод о большой значимости синестетических факторов в функционировании синестетических номинаций различных моделей в художественном тексте. Дальнейшая разработка вопроса роли эмоций в функционировании синестетических образований в художественном тексте видится нам перспективной в плане изучения не только синестетической метафоры как специфического языкового явления, но и в связи с такими глобальными проблемами, как проблемы взаимосвязи разума и эмоций, разума и тела.

Примечания

1. Cytowic, R. E. Synesthesia: Phenomenology and Neuropsychology [Electronic resource] / R. E. Cytowic. Режим доступа: <http://psyche.cs.monash.edu.au>
2. Галеев, Б. М. Синестезия – чудо поэтического мышления [Текст] / Б. М. Галеев // Научный Татарстан. 1999. № 3. С. 19–23.
3. Петренко, В. Ф. Введение в экспериментальную психосемантику. Исследование форм репрезентации в обыденном сознании [Текст] / В. Ф. Петренко. М.: МГУ, 1983. С. 98.
4. Бардовская, А. И. Средства номинации синестетических сообщений (на материале английских и русских художественных текстов) [Текст] : автореф. дис. ... канд. филол. наук / А. И. Бардовская. Тверь, 2005. 19 с.
5. Гордон, Д. Терапевтические метафоры [Электронный ресурс] / Д. Гордон. Режим доступа: <http://lib.baikal.net/>
6. Старобинский, Ж. Шкала температур: Язык тела в «Госпоже Бовари» [Текст] / Ж. Старобинский // Старобинский Ж. Поэзия и знание: История литературы и культуры. М.: Языки славянской культуры, 2002. Т. 1. С. 432–463.
7. Кундик, О. И. Специфика синестезийных словосочетаний в русском и английском языках [Текст] : дис. ... канд. филол. наук / О. И. Кундик. Саратов, 1997. 164 с.
8. Елина, Е. А. Вербальные интерпретации произведений изобразительного искусства [Текст] / Е. А. Елина. Саратов: Саратов. гос. соц.-эконом. ун-т, 2002. 256 с.

9. Сабанадзе, М. Я. Синестезия в подязыке музыкального языка: (на материале английского языка) [Текст] : автореф. дис. ... канд. филол. наук / М. Я. Сабанадзе. Л., 1987. 21 с.

10. Рудницка, Э. Вселенная духов [Текст] / Э. Рудницка // Ароматы и запахи в культуре. М.: Новое литературное обозрение, 2003. Кн. 2. С. 546–558.

11. Павлович, Н. В. Язык образов [Текст] / Н. В. Павлович. М.: «Азбуковник», 2004. 527 с.

12. Потебня, А. А. Эстетика и поэтика [Текст] / А. А. Потебня. М.: Искусство, 1976. 614 с.; Ульманн, С. Семантические универсалии [Текст] / С. Ульманн // Новое в зарубежной лингвистике. М.: Прогресс, 1970. Вып. V. С. 250–299.

Е. В. Кипрская

ОСНОВНЫЕ СФЕРЫ ИСПОЛЬЗОВАНИЯ ЭВФЕМИЗМОВ И ИХ ОТГРАНИЧЕНИЕ ОТ СХОДНЫХ ЛЕКСИЧЕСКИХ ЯВЛЕНИЙ

В статье рассматривается особенность функционирования эвфемизмов в современной речи в русском и английском языках, а также делается попытка провести четкую границу между различными похожими лексическими средствами языка.

В настоящее время принято считать, что социопсихические факторы определяют основные понятийные сферы, охватываемые эвфемией. Данным сферам соответствуют определенные лексические разряды эвфемизмов, группируемые в классификации. Типологические классификации разрабатывались в той или иной мере всеми исследователями эвфемизмов. В частности, Л. Блумфильд различает имена богов и всего, что вызывает уважение, «непристойные формы» и «зловещие языковые формы, которые обозначают что-то болезненное или опасное» [1: 261].

Значительное распространение получила классификация эвфемизмов, основанная на функциональном принципе. Так, Б. А. Ларин предложил классификацию по сферам применения эвфемизмов, в которой выделял три группы:

1) общеупотребительные эвфемизмы национального литературного языка;

2) классовые и профессиональные эвфемизмы;

3) семейно-бытовые [6].

В классификации эвфемизмов по порождающим мотивам, предложенной А. С. Куркиевым, выделяется четыре группы:

1) эвфемизмы, возникшие на основе суеверий (*болеть – нездоров, хворает*);

2) из чувства страха и неудовольствия (*убить – прибить, укокошить*);

3) на основе сочувствия и жалости (*больной – не все дома*);

4) порождаемое вежливостью (*старый – в летах, преклонный возраст*) [5].

Если в основу классификации Б. А. Ларина положена социальная природа явления, то А. С. Куркиев выделяет классы по порождающим мотивам.

Детальную классификацию предложил А. М. Кацев [3], в ней 10 лексических разрядов эвфемизмов:

1) эвфемизмы, обозначающие сверхъестественные силы *Lord – God*. Большое разнообразие и частое использование этого разряда эвфемизмов в современном английском языке объясняется влиянием религиозно-нравственного фактора. Запреты церкви и соответствующая языковая политика государства способствовали возникновению и закреплению в языковой практике большого числа эвфемизмов, возникших как реакция на различные запреты [3; 6]. Например, *goodness gracious, my Goodness* вместо *God*; *dickens* вместо *devil*;

2) эвфемизмы, обозначающие понятия «смерть», «болезнь» и связанные с ними действия предметы и качества: *to go to the Lord – to die; to bite the dust – to die in the battle*. В настоящее время происходит расширение данной лексической группы, ее пополнение новыми синонимическими рядами из-за большого количества локальных военных конфликтов, мировых войн и терроризма [4];

3) эвфемизмы, обозначающие умственные и физические недостатки. В подавляющем большинстве примеров эвфемистической замене подвергаются слова, обозначающие умственную неполноценность. Реже табуируются прямые обозначения понятий, относящихся к физическим недостаткам. Наиболее употребительны следующие эвфемизмы: *barmy (balmy), dense, deranged, dotty, soft, mentally defeated, not (quite) right in the head (silly или mad); Asylum, Home (mad-house)*. Синонимические ряды *mad* и *foolish* постоянно пополняются новыми эвфемизмами: *bonkers, kooky, mutton (mad); to flip out (to lose one's mind); homo insipiens (fool)*, о чем свидетельствуют словари табуизированной лексики и словари синонимов [2];

4) эвфемизмы, обозначающие пороки людей. Наибольшей распространенностью отличаются эвфемистические наименования понятий «пьяный» и «пьянство». Значительно реже употребляются эвфемизмы, означающие понятие «лгать», эпизодически – «трусить» и др.;

КИПРСКАЯ Екатерина Викторовна – кандидат филологических наук, доцент по кафедре германских языков ВятГУ

© Кипрская Е. В., 2008

5) эвфемизмы, обозначающие преступления и их последствия. Это – сфера одиозного и страшного. К числу наиболее распространенных эвфемизмов относят *to pinch (to steal); to do somebody in (to kill somebody); to put somebody away, to lock somebody up, to run somebody in (to arrest somebody); copper (policeman)*;

6) обозначение бедности, связанных с нею действий и качеств. Данный разряд не отличается большим разнообразием лексического материала. К наиболее употребительным принадлежат эвфемизмы *to be in a low water, to be on the rocks, to be in reduced circumstances, to get the bird (rocket, bullet)*. Отмечается распространенность слэнгизмов *to get the rocket (bullet)* и др.;

7) эвфемизмы, относящиеся к сфере сексуальных отношений, т. е. сфере постыдного. В этом лексическом разряде обращает на себя внимание значительное число и высокая частота встречаемости эвфемизмов. Эвфемия в данном лексическом пласте имеет свои особенности. Необходимо учитывать различия в плане социальной отнесенности с точки зрения стилевой принадлежности лексического материала. В настоящее время существуют противоборствующие тенденции в речи американской молодежи – с одной стороны, «боязнь показаться аффектированным и изнеженным» (*sisy*), а с другой – «боязнь употреблять слова с неприличным значением [1: 31]. Этой тенденцией можно объяснить преобладание слэнга в эвфемистической лексике современного молодого поколения, интенсивно используются ситуативные эвфемизмы типа *ballet, exploit, fun (sexual intercourse), brasstitute (prostitute)*. К числу неологизмов относят *preggers (pregnant), to liaison* от *liaison (sex intercourse)* [2];

8) обозначения некоторых частей тела;

9) обозначения некоторых предметов туалета;

10) обозначения некоторых физиологических функций.

Последние три лексических разряда, по мнению А. М. Кацева [3], составляют сравнительно небольшой по объему и распространенности материал, поэтому они рассматриваются в совокупности. К числу наиболее распространенных лексических единиц данного разряда принадлежат прежде всего, заменители слова *lavatory – Ladies, Gentleman, Gents, W.C., the smallest room*, а также заменители *naked – in the altogether, in a birthday suit* [2].

Отметим, что в понятийных сферах процессы табуирования и эвфемии протекают регулярно, данными разрядами не ограничивается лексический материал, который связан с эвфемией.

Важным представляется тот факт, что предложенные понятийные сферы выделяются и исследователями других языков [7]. Это позволяет

говорить о языковых универсалиях в плане эвфемии.

Однако наибольшее распространение получили тематические классификации эвфемизмов. Так, З. Я. Тураева, Г. Г. Почепцов выделяют три основные тематические группы:

1) эвфемизмы, используемые по принципу вежливости (эвфемистические выражения, направленные на смягчение различных типов дискриминации: возрастной; дискриминации, связанной с физическими или умственными недостатками, имущественной; и новая группа, включающая в себя эвфемизмы, смягчающие расовую и этническую дискриминацию);

2) эвфемизмы, основанные на принципе табуирования (включены эвфемизмы, основанные на суеверном страхе перед болезнью, смертью и т. д.);

3) эвфемизмы, камуфлирующие негативные явления в экономической, политической и социальной сторонах жизни (включены эвфемизмы, прикрывающие различные формы военной агрессии; смягчающие различные непопулярные меры в социально-экономической сфере и связанные с преступной деятельностью);

4) эвфемизмы, направленные на повышение социального статуса отдельных профессий [9; 7].

Значительное количество исследований посвящено анализу отдельных тематических групп и категорий эвфемизмов. Мы считаем возможным выделить некоторые тенденции, характеризующие современное состояние эвфемии. Так, многие из «классических» групп эвфемизмов, в первую очередь касающиеся социально-экономических сторон жизни и личной сферы, пополнились большим количеством эвфемизмов-неологизмов. Например, «слово *the poor* (бедные) в печати было заменено на *the needy, the ill-provided* (необеспеченные), затем на *the deprived* (лишенные благ), затем на *the socially deprived* (социально обездоленные), *the underprivileged* (малопривилегированные), а позже на *the disadvantages* (попавшие в менее благоприятные обстоятельства) и, наконец, на *low-income people* (малообеспеченные)» [10: 113]. В личной сфере период жизни от 65 лет и далее стали называть *third age*. Избегается употребление слова *old* (старый), вместо него распространены такие эвфемизмы, как *senior, mature* и *seasoned* [7]. В связи с изменениями в экономической жизни общества, общим повышением уровня жизни людей некоторые физические недостатки стали заменять эвфемизмами: *fat* (толстый) заменяется на *big – boned, differently sized*, вместо *bald* (лысый) используется эвфемизм *hair-disadvantaged*. Появились неологизмы в синонимических рядах эвфемизмов, обозначавших стандартные физические недостатки: вместо *deaf* (глухой) – *aurally inconvenienced*, вмес-

то *blind* (слепой) – *unseeing*. Умственно отстающих людей называют *learning disable, special, mentally challenged people*.

Тематическая группа, вызванная желанием завуалировать страх перед сверхъестественными силами, также претерпела некоторые изменения, связанные с реалиями наших дней. Например, *moonchild* (человек рожденный под созвездием Рака) заменило слово *cancer*, вызывающее ассоциации с болезнью. В данном случае используется метафора, в которой полностью отсутствуют семы, связанные с болезнью и смертью. *Hospice*, первоначально обозначающее приют или гостиницу, теперь обозначает больницу для неизлечимых больных, слова, где опущены семы 'болезнь', 'смерть'. Эвфемизмы *therapeutic terminal episode, therapeutic misadventure* или *negative patient care outcome*, которые являются описательными перифразами, полностью заменили глагол *to die*.

Усилилась тенденция к образованию новых эвфемизмов, поднимающих социальный статус отдельных профессий. К примеру, парикмахера стали называть *hairstylist* или *beautician*. Служащие кладбища традиционно были известны как *undertakers*, в последние годы их стали называть *morticians*, а затем *funeral directors*. Слово *garbage collector* (сборщик мусора) заменяется на *sanitation engineer*, а эвфемизм *environmental hygienist* – не что иное, как название дворника (*a janitor*).

Значительные изменения произошли в группе эвфемизмов, связанных с преступностью. Эвфемизм *correctional facilities* (исправительное учреждение) сменил слово *prison* (тюрьма), *correctional officers* или *custodial officers* (офицеры охраны) заменяют прежнее выражение *prison guards* (тюремные надзиратели). Самых заключенных теперь называют *clients of correctional system* (клиенты системы исправительных учреждений), *guests* (гости) или *people enjoying temporarily hospitality from the state* (люди, пользующиеся временным гостеприимством у государства).

Появились новые классификационные группы эвфемизмов, связанные с явлением политической корректности, в которой процесс эвфемизации является одним из способов камуфлирования действительности. Это, в первую очередь, эвфемизмы, смягчающие расовую и этническую дискриминацию. Такие слова являются, в основном, названиями различных расовых и национальных групп, подчеркивающими их самобытность и равноправный статус. Слово *black* (чернокожий) заменяется эвфемизмом *member of the African diaspora* (представитель африканской диаспоры). *Indian* (индеец) – словосочетанием *indigenous person* (местный житель).

Следующая группа – это эвфемизмы, служащие прикрытием агрессивных военных действий.

Значительная часть таких слов проникла в язык еще в 60-е гг. XX в. во время войны во Вьетнаме. Слова *involvement* (вовлечение) и *conflict* (конфликт) заменили *aggression* (агрессия) и *war* (война), уничтожение вооруженного сопротивления стали называть *pacification* (умиротворение), вместо слова *bomb* (бомба) употреблялось слово *device* (устройство), а вместо *bombing* (бомбардировка) – *limited air strike* (ограниченный воздушный удар) или *air support* (воздушная поддержка). Эвфемизм *strategic hamlet* (стратегическая деревушка) заменил *concentration camp* (концентрационный лагерь). Наконец, убитых стали называть *body-count* или *inooperative combat personnel*.

Значительное место занимают эвфемизмы, связанные с негативными последствиями в социально-экономической сфере. В течение XX в. *economic crisis* (экономический кризис) было вытеснено *slump* (падение), *slump* было заменено на *depression* (депрессия), что вызвало новый эвфемизм *recession* (отступление). Более современные термины в значении 'экономический кризис' включают *period of economic adjustment* (период стабилизации экономики), *period of negative economic growth* (период отрицательного экономического роста), *meaningful downturn* (значительный спад). Чтобы смягчить негативный эффект слов *firing laying off* (увольнение), возникли эвфемизмы *downsizing, rightsizing, redundancy elimination*. Название самой социальной структуры – капитализм – приобрело за последние десятилетия ряд новых заменителей несомненно эвфемистического характера: *the system of free enterprising* (система свободного предпринимательства), *open society* (открытое общество), *economic humanism* (экономический гуманизм).

Интерпретация эвфемизмов в современной лингвистической литературе зависит от понимания того, какие именно экстралингвистические сферы, по каким мотивам подвергаются завуалированию и как это осуществляется лингвистически. Природа эвфемизмов до настоящего времени остается невыясненной, что приводит к стиранию границ между эвфемизмами и другими видами перефразирования. Существует значительное количество исследований, в рамках которых эвфемизмы рассматривают как один из видов косвенных наименований. Мы сочли возможным выделить 4 группы косвенных наименований:

1) замененные слова и выражения, вызванные дипломатической тайной, политическими соображениями, юридическими нормами. В качестве примеров можно привести такие выражения, как *одна соседняя держава* или *N-ский город*. В эту же категорию входят и слова *жители* или *обыvatели* вместо *граждане, государство* вместо *отечество*, приводимые Е. И. Шейгал как при-

меры наложения табу, декретированного сверху. Данные слова не являются даже косвенными наименованиями [3; 10];

2) косвенные наименования по эстетическим мотивам. Примерами могут служить *рыцари педали* вместо *велосипедисты*, *saloon* вместо *bar* или *lady* вместо *woman*. В подобных случаях с эвфемией нередко отождествляют декоративность, манерность речи [6];

3) формы вежливости типа местоимения *Вы* вместо *ты* в русском языке, глаголы *предлагаю* или *рекомендую* вместо *приказываю*, вежливые фразы, выражающие отказ или отрицание;

4) случаи криптологии, приводимые А. А. Реформатским [8].

Мы считаем, что необходимо четко отграничивать различные виды косвенных наименований от эвфемизмов. Хотя как косвенные наименования, так и эвфемизмы способствуют эффекту смягчения, эвфемизмы, в отличие от косвенных наименований, выступают как заменители страшного, постыдного и одиозного, т. е. различаются не только мотивы, но и сферы завуалирования.

В большинстве исследований эвфемизмы сравниваются с различными видами тропов (метафоры, синекдохи, метонимии), литотой и гиперболой. При этом принято выделять следующие сходные признаки:

- эвфемизмы связаны с метафорическим переносом (в широком понимании), так как эвфемистический перенос наименования базируется на характерных признаках метафоры;

- имплицитное наличие литоты или гиперболы в значении слова может привести к созданию эвфемизма. Эвфемизм и ирония стилистически близки, только эвфемизмы возникают благодаря сближению значений слов, ирония – благодаря противопоставлению, противоположности. Эвфемизмы, таким образом, поддерживают «ценностные» нормы общения, ирония их нарушает [4; 6].

Главное отличие эвфемизма от других видов иносказания (аллегории, гиперболы, литоты, мейозиса) – в его функции: к эвфемизмам прибегают тогда, когда хотят завуалировать, закамуфлировать некий смысл, который говорящий почему-либо считает неудобным обозначать прямо [9]. Целью тропов поэтического языка является образное представление действительности, а целью эвфемизации речи – прикрытие неприглядных сторон жизни общества [6].

А. А. Халанская предлагает выделять основные тропеистические и перифрастические качества эвфемизмов. К основным сущностным перифрастическим признакам эвфемизмов относятся косвенность, описательность, иносказательность вторичной номинации, благодаря которой

реализуется их завуалирующая, смягчающая и укрывающая функции» [9: 23].

Существует мнение, что эвфемизмы, в зависимости от ситуации, могут нести в себе больше или меньше информации по сравнению с объективным высказыванием. Подобная ситуация складывается в результате того, что неподходящие семы изымаются, а вместо них используются более пригодные. При этом сохраняется инвариант, который способствует распознаванию значения эвфемизма, но к нему могут примыкать случайные семы, которые препятствуют его идентификации вследствие деформирования смысла [1].

В связи с этим можно сделать вывод, что эвфемизмы представляют собой универсальное лингвистическое явление, которое обычно обусловливается социально-историческими и морально-этическими нормами, национальными и языковыми традициями того или иного общества.

Сущность эвфемизмов, их эволюция связаны с установлением общественного и языкового конвенционализма. Социально-культурные, морально-этические и эстетические нормы общества формируют абстрактное понятие о положительности (отрицательности) различных предметов и явлений, что, в свою очередь, способствует корректности (некорректности) существующих наименований в соответствующих коммуникативных ситуациях. Определение социальной роли эвфемизма в конструировании подобных ситуаций актуально в плане изучения взаимодействия языка и общества.

Природа эвфемизмов остается невыясненной, что приводит к смешению эвфемизмов с другими видами перефразирования: литотой, гиперболой, различными видами косвенных наименований, вследствие чего не разработаны четкие критерии идентификации эвфемизмов и отсутствует общепринятая дефиниция самого термина; существуют многочисленные точки зрения на отграничение современных эвфемизмов от древних слов-табу.

В лингвистической литературе прослеживается устойчивая тенденция к появлению новых тематических групп эвфемизмов вследствие возникновения новых сфер в социально-политической жизни общества, подлежащих завуалированию.

Примечания

1. Блумфильд, Л. Язык [Текст] / Л. Блумфильд. М.: Просвещение, 1968. 432 с.

2. Дейк, Т. А. ван. Язык. Познание. Коммуникация [Текст] / Т. А. ван Дейк; пер. с англ. М.: Прогресс, 1989. 312 с.

3. Кацев, А. М. Языковые табу и эвфемия [Текст] / А. М. Кацев. Л.: Наука, 1988. 159 с.

4. Крысин, Л. П. Эвфемизмы в современной русской речи [Текст] / Л. П. Крысин // Русистика. 1994. № 1–2. С. 28–49.

5. Куркиев, А. С. О классификации эвфемистических названий в русском языке. Классификация эвфемизмов по порождающим мотивам [Текст] / А. С. Куркиев. Грозный, 1977. 88 с.

6. Ларин, Б. А. Об эвфемизмах [Текст] / Б. А. Ларин // Проблемы языкознания. Л., 1961. С. 23–48.

7. Почепцов, Г. Г. Языкрвая ментальность: Способ представления мира [Текст] / Г. Г. Почепцов // Вопросы языкознания. 1990. № 6. С. 110–122.

8. Реформатский, А. А. Введение в языковедение [Текст] / А. А. Реформатский. М.: Просвещение, 1991. 364 с.

9. Тураева, З. Я. Лингвистика текста [Текст] / З. Я. Тураева. М.: Просвещение, 1986. 127 с.

10. Шейгал, Е. И. Семиотика политического дискурса [Текст] / Е. И. Шейгал. Волгоград: Перемена, 2000. 368 с.

Н. С. Кожина

ПЕРСУАЗИВНОЕ СООБЩЕНИЕ И ЕГО КАНАЛ В СОВРЕМЕННОЙ КОММУНИКАЦИИ

В статье рассматриваются различные принципы построения персуазивного сообщения, анализируется связь его содержания с целью высказывания. Приводится описание наиболее распространенных техник, методов и средств речевого воздействия, а также отмечается значимость коммуникативного канала в современной коммуникации.

Сообщение речевого воздействия (далее – РВ) как важный компонент коммуникации подвергается глубокому и разностороннему анализу со стороны исследователей. Этот фактор является связующим звеном между продуцентом и реципиентом. Первый вкладывает в сообщение свою интенцию и знания о последнем, облекая их в такую форму и задействуя такие средства, которые помогли бы добиться желаемого результата. Соответственно, персуазивное сообщение исследуется на предмет используемых реципиентом стратегий, тактик, техник и средств РВ.

Стратегии РВ представляют собой «тип или линию поведения одного из коммуникантов в конкретной ситуации общения, которые соотносятся с планом достижения преимущественно глобальных (а иногда и локальных) коммуникативных целей в рамках всего сценария функционально-семантической репрезентации интерактивного типа» [16: 19]. Хотя понятие коммуникативных стратегий в современной лингвистике обнаруживает весьма широкую вариативность [12: 138], мы считаем целесообразным принять дефи-

ницию стратегии как комплекса речевых действий, направленных на реализацию определенного намерения [8].

В прагмалингвистическом исследовании Г. Г. Матвеевой отмечается, что речевое действие отправителя текста характеризуется осознанностью и продуманностью, стремлением выбрать удачную лингвистическую единицу для успешного воздействия на получателя текста [13]. Автор подразделяет стратегии РВ на две большие группы: 1) эмотивно-ориентированные стратегии, направленные на продуцента; 2) конативно-ориентированные речевые стратегии, направленные на реципиента. В каждой группе выделяется по три основные стратегии, которые реализуются в виде определенных речевых планов-вариантов (см. рисунок).

Следуя классическому аристотелевскому учению, многие исследователи выделяют три основные стратегии РВ: 1) рациональные, построенные с применением методов индукции и дедукции; 2) этические, связанные с личностью, характером, качествами говорящего; 3) эмоциональные, содержащие обращение к потребностям, ценностям, чувствам аудитории [24].

Родоначальники теории достижения согласия (compliance gaining) Дж. Марвел и Д. Шмит рассматривают РВ с точки зрения правил социального обмена, согласно которым действия коммуниканта направлены на получение чего-либо от другого коммуниканта в обмен на что-либо. Такая модель РВ изначально связана с силовой позицией продуцента, успешность коммуникативных действий которого зависит от того, какой стимул он может предложить реципиенту в обмен на достижение согласия. Дж. Марвел и Д. Шмит выделили пять основных стимулирующих стратегий, включающих награду, наказание, «экспертизу» (демонстрацию знаний о последствиях действий реципиента), общечеловеческие обязательства, личные обязательства [22: 360].

Исследовательская группа, состоящая из У. Шенк-Хамлина, Р. Вайзмана и Г. Н. Георгакаракоса, переработала теорию Марвела – Шмита и предложила свою модель достижения согласия [23; 26], включающую четыре основных элемента. Первым элементом этой модели является степень откровенности продуцента сообщения относительно своих коммуникативных целей. В зависимости от степени откровенности стратегии достижения согласия могут быть прямыми, косвенными или маскирующими. Второй элемент модели – мотивация, основанная на применении санкций (например, наказания или поощрения) или объяснении причин попытки воздействия. Третий элемент включает наличие или отсутствие логического обоснования коммуникативных дей-

КОЖИНА Наталья Сергеевна – старший преподаватель кафедры германских языков ВятГГУ
© Кожина Н. С., 2008

ствий продуцента. Четвертая составляющая этой модели связана с распределением контроля в коммуникативной ситуации. Так, от выбора типа мотивации зависит, кем контролируется коммуникативная ситуация: продуцентом сообщения или реципиентом. Контроль будет на стороне продуцента при использовании стратегии обещания поощрения или наказания. Если же продуцент в качестве мотивации взывает к чувству вины или ответственности реципиента, то контроль над ситуацией переходит к последнему.

Интересным представляется еще один подход к проблеме РВ, также разработанный в рамках теории достижения согласия. Группа исследователей под руководством А. Уилиса разработала классификацию стратегий РВ в зависимости от типа власти, используемого коммуникантами при попытке достижения согласия [25]. Власть обеспечивает доступ к влиянию на других людей. Она является результатом межличностного восприятия, поскольку у коммуниканта ровно столько

власти, сколько ему приписывается другими участниками коммуникативного процесса.

А. Уилисом и его сторонниками были выделены три основные типа власти. Первый тип власти – манипулятивный. Он связан с наличием у продуцента сообщения возможности манипулировать последствиями коммуникативного действия реципиента. Второй тип – отношенческий. Он базируется на взаимоотношениях коммуникантов. Вне зависимости от рода отношений всегда существует большая вероятность неравного распределения контроля между коммуникантами. Соответственно, наибольшей эффективности РВ продуцент сообщения может добиться при условии, если на его стороне больше контроля в отношениях с реципиентом. Третий тип власти – ценностный, он основан на способности продуцента определять и воплощать ценности. Чем выше эта способность продуцента, тем легче ему определять нормы поведения реципиента и оказывать на него влияние.



Все перечисленные выше направления теории достижения согласия описывают ряд стратегий РВ, выведенных из наиболее типичных моделей поведения коммуникантов (табл. 1). Близкими по тематике и заслуживающими внимания являются отечественные исследования речевой стратегии согласия/несогласия [17] и стратегии одобрения/неодобрения [6]. На наш взгляд, такое разнообразие подходов отражает сложность и неоднозначность феномена РВ, а в совокупности разные классификации позволяют составить наиболее полное представление о стратегиях РВ, что в итоге способствует повышению эффективности их применения.

Подробное описание стратегий и тактик РВ предлагает О. А. Михалева [14]. Автор выделяет три основные группы стратегий:

1) *стратегия на понижение* выражает отношение говорящего, направленное на овладение коммуникативной инициативой и дискредитацию оппонента;

2) *стратегия на повышение* реализуется при стремлении продуцента максимально увеличить значимость собственного статуса, продемонстрировать силу, уверенность, самодостаточность, компетентность;

3) *стратегия театральности* выделяется автором с опорой на работы [7; 18] и свидетельствует о значимости фактора аудитории. Ранее использование театральной метафоры для описания процесса коммуникации было представлено в работе К. Берка [19].

Каждая из перечисленных выше стратегий реализуется при помощи определенных тактик (см. табл. 2).

Теория нарратива, разработанная У. Фишером [20], описывает еще один интересный подход к построению сообщения РВ. Нарративом может считаться любой имеющий смысл вербальный или невербальный отчет о последовательности собы-

тий. Эффективность нарратива как средства РВ обеспечивается наличием связности и ценностной ориентации. В нарративе важны три вида связности: 1) внутренняя, т. е. гармоничное сочетание частей нарратива друг с другом; 2) внешняя, которая обеспечивает конгруэнтность данного нарратива другим нарративам; 3) характерная, показывающая, насколько правдоподобны действующие лица нарратива и какие идеи они пропагандируют. Как уже упоминалось, нарратив должен обладать ценностной ориентацией, т. е. демонстрировать ценности, значимые для реципиента и принимаемые им как аргументы в пользу положений продуцента. У. Фишер выделяет следующие закономерности функционирования ценностей в нарративе: (1) мораль нарратива ориентирована на ценности, которыми руководствуются персонажи в своих поступках; (2) ценности напрямую связаны с позитивными последствиями в объективной реальности; (3) ценности соотносятся с опытом реципиента; (4) следование ценностям представлено в нарративе как составляющая идеального человеческого поведения.

У. Фишер считает, что использование нарратива при создании персуазивного сообщения имеет ряд преимуществ по сравнению с аргументами, подчиняющимися законам формальной логики [20; 21]. Во-первых, нарратив намного ближе к реальному опыту реципиента, так как при его создании и восприятии задействованы все познавательные процессы психики. Во-вторых, нарратив основан на культурном знании, он понятен любому представителю данной культуры и не требует от него специальных профессиональных знаний. В-третьих, нарратив стимулирует процесс идентификации у коммуникантов и поэтому воздействует на них на подсознательном уровне. В силу всех вышеперечисленных причин представляется возможным сделать вывод, что

Таблица 1

ПАРАМЕТР	СТРАТЕГИЯ
Выраженность коммуникативного намерения	1) прямая; 2) косвенная; 3) маскирующая
Мотивация	• применение санкций; • объяснение причин
Наличие логического обоснования	1) рациональная; 2) эмоциональная
Контроль над коммуникативной ситуацией	• в зависимости от власти реципиента: ○ зывание к чувству вины или ответственности; • в зависимости от власти продуцента: ○ манипулятивная; ○ отношенческая; ○ ценностная

использование нарратива особенно эффективно при формировании общественного мнения по спорным моральным вопросам.

Политический дискурс традиционно является одной из основных сфер применения различных приемов, повышающих эффективность РВ. Исследуя сущность феномена формирования общественного мнения, С. Г. Кара-Мурза [9] выделил следующие наиболее распространенные методы манипуляции сознанием:

- внушение;
- перенос частного факта в сферу общего;

- использование слухов в неясной ситуации;
- метод под названием «нужны трупы»;
- метод «страшилок»;
- замалчивание одних фактов и «выпячивание» других;
- мистификация (создание лжесобытий);
- фрагментация;
- многократные повторы.

Как видно из приведенного списка, некоторые из перечисленных методов уже обсуждались нами выше. Представляется важным пояснить, что повтор как средство РВ используется довольно

Таблица 2

СТРАТЕГИЯ	ТАКТИКА	КРАТКОЕ ОПИСАНИЕ ТАКТИКИ
I. На понижение	1. Анализ-«минус»	Фактический разбор ситуации, предполагающий выражение <i>отрицательного</i> отношения к описываемому
	2. Обвинения	Приписывание кому-либо какой-либо вины, раскрытие чьих-либо неблагоприятных действий, намерений, качеств
	3. Обличения	Приведение фактов и аргументов, делающих явной чью-то виновность, преступность
	4. Оскорбления	Унижение, сопровождающееся экспликацией эмоционального составляющего компонента вместо приведения доказательств
	5. Угрозы	Запугивание, обещание причинить адресату неприятность, вербализация негативных последствий, ожидающих противоположную сторону
II. На повышение	1. Анализ-«плюс»	Фактический разбор ситуации, предполагающий выражение <i>положительного</i> отношения к описываемому
	2. Презентации	Представление кого-либо в привлекательном виде
	3. Неявной саморепрезентации	Выраженное косвенно представление говорящим себя в выгодном свете
	4. Отвода критики	Приведение с целью доказательства невиновности аргументов/фактов, с помощью которых можно объяснить какие-либо действия и поступки
	5. Самооправдания	Отрицательное суждение об объекте критики и его причастности к тому, что критикуется
III. Театральности	1. Побуждения	Призыв к действию, принятию точки зрения
	2. Кооперации	Обращение к адресату, с помощью которого говорящий конструирует образ последнего, апеллируя к его идеалам и ценностям. Избираются языковые средства, активизирующие в сознании слушателя те ментальные структуры, с помощью которых коммуникант позиционирует свою принадлежность к определенной группе
	3. Размежевания	Выявление различий в позициях и мнениях, основное средство реализации – местоимения «мы – они» и соответствующие формы глаголов
	4. Информирования	Приведение фактов и данных без выражения отношения говорящего
	5. Обещания	Добровольное обязательство сделать что-либо, использование формы будущего времени глаголов совершенного вида выражает уверенное обещание
	6. Прогнозирования	Предсказание дальнейшего течения событий с опорой на имеющиеся данные, использование глаголов совершенного вида имплицитно выражает мысль о достигнутом положительном результате
	7. Предупреждения	Предваряющее извещение о возможных событиях, действиях
	8. Иронизирования	Осуществление воздействия за счет контраста между сказанным и подразумеваемым
	9. Провокации	Подстрекательство кого-либо к таким действиям, которые могут повлечь за собой тяжелые для него последствия

часто, его главной функцией является выделение основной мысли текста. Е. В. Афанасенко рассматривает две группы повторов: 1) тавтологический, при котором наблюдается совпадение плана содержания и плана выражения; 2) семантической эквивалентности, предполагающей семантическое сходство при различиях внешней формы [1].

По мнению В. А. Даулетовой, политический дискурс – благодатная почва для применения техник нейролингвистического программирования, к которым можно отнести следующие: 1) умышленное использование пресуппозиций через вводные слова («очевидно», «ясно», «конечно»), модальные операторы («требуется», «должен»), конструкции с «если»; 2) перегрузку сознания реципиента информацией или воздействие на разные каналы получения и переработки информации; 3) метафоризацию [5: 8–9].

Следует отметить, что метафора как средство РВ активно исследуется на протяжении многих лет. В последнее время особенно актуальными стали исследования функционирования метафоры в политическом дискурсе (см., например, [3; 10]). В число наиболее популярных средств РВ входят также такие стилистические средства, как сравнение, эпитет, фразеологизм, каламбур, ирония, синонимия, антонимия, оксюморон, гипербола, литота [2; 4]. Считается, что частота употребления и предпочтение того или иного средства РВ связано с общей тенденцией языка к упрощению, стремлением довести мысль до аудитории путем экономии времени и усилий.

С фактором персуазивного сообщения тесно связано понятие канала, по которому это сообщение идет от продуцента к реципиенту. Отмечается, что регуляция деятельности индивида может быть непосредственной (прямой) и опосредованной [11: 32]. Прямое РВ предполагает непосредственную передачу сообщения оратором аудитории при личном контакте, обычно это происходит в форме митинга, беседы, лекции, публичного выступления. Прямое РВ характерно также для бытовой коммуникации, в которой коммуниканты общаются «лицом к лицу».

Опосредованное РВ исключает личный контакт между коммуникантами. Оно, например, осуществляется через искусство, театр, литературу. XX век ознаменовался появлением новых средств связи, таких, как телефон и телеграф. Эти средства выполняли функцию технического канала, передающего объективную информацию от источника к получателю. Позднее появились такие каналы передачи сообщения, как СМИ и Интернет. В итоге картина мира современного человека формируется вследствие его включенности в социальное взаимодействие, обеспечиваемое масс-медиа. И. В. Рогозина сделала под-

робное описание феномена медиа-коммуникации и пришла к выводу, что СМИ «осуществляют социально-преобразующую репрезентацию реальности» [15: 61].

Исследования персуазивного сообщения разнообразны и многочисленны. Их целью всегда являлось определение и описание текстовых компонентов и средств, обеспечивающих эффективность коммуникации. В последнее время в связи с изменением характера коммуникации и роли ее канала, в частности усиления влияния СМИ, стали активно анализироваться тексты масс-медиа как факторы структуризации реальности в сознании членов социума.

Примечания

1. Афанасенко, Е. В. Семантический повтор в политическом дискурсе (на материале русского и английского языков) [Текст] : автореф. дис. ... канд. филол. наук / Е. В. Афанасенко. Саратов, 2006. 18 с.
2. Баранов, А. Н. Словарь русских политических метафор [Текст] / А. Н. Баранов, Ю. Н. Караулов. М.: Редакция САМ «Помовский и партнеры», 1994. 351 с.
3. Будаев, Э. В. Метафора в политическом интердискурсе [Текст] : монография / Э. В. Будаев, А. П. Чудинов. Екатеринбург: Урал. гос. пед. ун-т, 2006. 208 с.
4. Голуб, И. Б. Стилистика русского языка [Текст] : учеб. пособие / И. Б. Голуб. М.: Айрис-Пресс, 2006. 448 с.
5. Даулетова, В. А. Вербальные средства создания автоимиджа [Текст] : автореф. дис. ... канд. филол. наук / В. А. Даулетова. Волгоград, 2004. 22 с.
6. Дворник, Т. В. Одобрение/неодобрение как категория субъект-объектных отношений [Текст] / Т. В. Дворник // Единичи языка: функционально-коммуникативный аспект. Ч. 2. Ростов н/Д: РГПУ, 2002. С. 96–97.
7. Желтухина, М. Р. Комическое в политическом дискурсе конца XX века. Русские и немецкие политики [Текст] : монография / М. Р. Желтухина. М.: Ин-т языкознания РАН; Волгоград: Изд-во ВФ МУПК, 2000. 264 с.
8. Иссерс, О. С. Коммуникативные стратегии и тактики русской речи [Текст] : монография / О. С. Иссерс. Омск: Изд-во Омск. гос. ун-та, 1999. 285 с.
9. Кара-Мурза, С. Г. Манипуляция сознанием [Текст] / С. Г. Кара-Мурза. М.: Алгоритм, 2004. 528 с.
10. Каслова, А. А. Метафорическое моделирование президентских выборов в России и США (2000 г.) [Текст] : автореф. дис. ... канд. филол. наук / А. А. Каслова. Екатеринбург, 2003. 21 с.
11. Леонтьев, А. А. Язык, речь, речевая деятельность [Текст] / А. А. Леонтьев. М.: Едиториал, УРСС, 2003. 216 с.
12. Макаров, М. А. Интерпретативный анализ дискурса в малой группе [Текст] / М. А. Макаров. Тверь: Твер. гос. ун-т, 1998. 200 с.
13. Матвеева, Г. Г. К вопросу о речевых стратегиях скрытого воздействия отправителя текста на его получателя [Электронный ресурс] / Г. Г. Матвеева. Режим доступа: <http://pi.sfedu.ru/projects/deutch/note44.html>
14. Михалева, О. А. Политический дискурс как сфера реализации манипулятивного воздействия

[Текст] : монография / О. А. Михалева. Иркутск: Иркут. ун-т, 2005. 320 с.

15. *Рогозина, И. В.* Медиа-картина мира: когнитивно-семиотический аспект [Текст] : монография / И. В. Рогозина; под ред. В. А. Пищальниковой. М.; Барнаул: Изд-во АлтГТУ, 2003. 289 с.

16. *Романов, А. А.* Системный анализ регулятивных средств диалогического общения [Текст] / А. А. Романов. М.: Ин-т языкознания АН СССР, 1988. 182 с.

17. *Чуриков, М. П.* Стратегия речевого поведения «согласие-несогласие» для воздействия говорящего на слушающего [Текст] / М. П. Чуриков // Актуальные проблемы филологии и методики преподавания. Ч. 1. Ростов н/Д: РГПУ, 2001. С. 204–207.

18. *Шейгал, Е. И.* Театральность политического дискурса [Электронный ресурс] / Е. И. Шейгал. Режим доступа: <http://www.philology.ru/linguistics1/sheygal-00.htm>

19. *Burke, K.* Language as symbolic action: essays on life, literature, and method [Text] / K. Burke. Berkely: University of California Press, 1966. 514 p.

20. *Fisher, W. R.* Human communication as narration: toward a philosophy of reason, value, and action [Text] / W. R. Fisher. Columbia, S.C.: University of South Carolina Press, 1987. 201 p.

21. *Littlejohn, S. W.* Theories of human communication [Text] / S. W. Littlejohn. 7th ed. Belmont, CA: Wadsworth Publishing Company, 2002. 378 p.

22. *Marwell, G.* Dimensions of compliance-gaining strategies: a dimensional analysis [Text] / G. Marwell, D. R. Schmitt // Sociometry. 1967. № 30. P. 350–364.

23. *Schenck-Hamlin, W. J.* A model of properties of compliance-gaining strategies [Text] / W. J. Schenck-Hamlin, R. L. Wiseman, G. N. Georgacarakos // Communication Quarterly. 1982. № 30. P. 92–100.

24. *Weida, S.* Using Rhetorical Strategies for Persuasion [Electronic resource] / S. Weida. Режим доступа: <http://owl.english.purdue.edu/owl/resource/588/04>

25. *Wheeless, L. R.* Compliance-gaining and power in persuasion [Text] / L. R. Wheeless, R. Barraclough, R. Stewart; Bostrom R. N. (ed.). Communication Yearbook 7. Beverly Hills: Sage, 1983. P. 105–145.

26. *Wiseman, R. L.* A multidimensional scaling validation of an inductively-derived set of compliance-gaining strategies [Text] / R. L. Wiseman, W. J. Schenck-Hamlin // Communication Monographs. 1981. № 48. P. 251–270.

Н. В. Новикова

СТИЛИСТИЧЕСКИЕ ФУНКЦИИ АРХАИЗМОВ В НОВЕЛЛАХ Э. ПО

Проблема экспрессивных возможностей архаизмов нуждается в переосмыслении и доработке. Связь архаизмов различных языковых уровней с содержанием произведения и творческим замыслом автора более сложна и неоднозначна, чем это может показаться на первый взгляд. Традиционно-формальный подход к устаревшим языковым явлениям опровергается самими языковыми фактами на примере произведений американского прозаика и поэта Э. По.

С точки зрения стилистики архаизмы являются выразительным средством, выполняющим то или иное стилистическое задание. В устной речи архаизмы практически не используются, но получают широкое применение в языке художественных произведений.

В лингвистической литературе сложилось два подхода к описанию экспрессивных возможностей архаизмов. Во-первых, архаизмы традиционно рассматриваются как средство создания колорита эпохи в исторической беллетристике [1: 12]; назначение архаизмов в исторических произведениях – создавать у читателя эффект присутствия при исторических событиях. Во-вторых, архаизмы могут рассматриваться как разновидность возвышенной лексики, используемой в поэзии, публицистической и ораторской речи [2: 257; 3: 161]. И в том и в другом случае (в поэзии и официальной речи) архаизмы являются выразительными средствами, но именно в поэтических произведениях функция архаизмов как средства создания торжественного, возвышенного тона выступает наиболее отчетливо. Некоторые исследователи считают поэтому архаизмы разновидностью поэтизмы, цель которых – придать тексту стилистическую окраску, сделать его отличным от текстов обыденной речи [2: 261; 4: 104].

Появление архаизмов в поэзии закономерно: оно объясняется их более приподнятой стилистической тональностью сравнительно с общими словами, во-первых, благодаря известной необычности архаичных слов и конструкций, неупотребительности их в обыденной речи, во-вторых, в результате неизбежных ассоциаций с малознакомой опозитизированной древностью и, в-третьих, по причине принадлежности их к традиционному поэтическому словарю, соответствия их особым нормам поэтического языка, которые касаются и выбора лексики, и морфологических форм, и синтаксических конструкций.

НОВИКОВА Наталья Вячеславовна – ассистент кафедры германских языков ВятГГУ
© Новикова Н. В., 2008

Архаизму присуща особая выразительность, он стилистически окрашен на фоне других, нейтральных современных слов. Иногда эта окраска настолько сильна, что затеняет, оттесняет на второй план значение слова или оборота, т. е. форма оказывается важнее содержания, и архаизм выступает как условный знак, примета «былых времен» [5: 122].

Введение в текст архаизма может создавать «иллюзию переноса во времени (в другую эпоху)» [5: 120] (функция исторической стилизации) либо переносить читателя в особый надвременной и надпространственный мир – так, как это происходит в произведениях Э. По.

Исследователи творчества Э. По отмечают принцип неопределенности в его поэтике: «В творениях поэта, как правило, отсутствуют пространственные, временные и иные характеристики, придающие содержанию определенность, конкретность и тем самым единичность» [6: 573]. Особенности мировоззрения Э. По наложили отпечаток на его творчество: «Ощущение угасания, безнадежности, бесцельности... окрасило все мироощущение писателя, легло в основу созданного его воображением *вневременного и внепространственного мира*, в котором бьется в трагическом надрыве охваченная ужасом душа человека» [6: 574].

Характерной особенностью этого писателя является расширение сферы использования устаревших языковых элементов, перенос стилистических функций архаизмов из поэзии в прозу. В прозе появление архаичных грамматических форм, которые сочетаются с лексикой возвышенной стилистической тональности, всегда связано у Э. По с тематикой конкретных произведений. Так, новеллы «Лигейя», «Элеонора», «Морелла» объединены известным сходством сюжета и образов: во всех названных рассказах вновь и вновь возникает излюбленная тема Э. По: «смерть молодой прекрасной женщины и горе ее возлюбленного» [7: 360]. Необходимо отметить, что в начале XIX в. этот мотив вообще характерен для тогдашней американской литературы. Не следует, однако, забывать, что для По «смерть молодой прекрасной женщины» – событие глубоко личное: в раннем детстве он потерял мать; когда ему было 19, умерла его приемная мать Френсис Аллан; от туберкулеза умерла Джейн Стенард, в которую поэт был влюблен в ранней юности; в 1847 г., умерла его жена Виргиния.

Шарль Бодлер, открывший Эдгара По французскому читателю, писал: «В рассказах По никогда не говорится о любви. “Элеонора” и “Лигейя” собственно говоря вовсе не представляют любовных историй; главная идея, которая пронизывает оба произведения, совершенно иная... Его портреты женщин окружены, так сказать,

ореолом; они сверкают среди неземной дымки, они написаны восторженным обожателем... Его женщины, все лучезарные и болезненные, умирают от каких-то странных болезней, говорят голосом, подобным музыке... своими странными стремлениями, познаниями, неизлечимой грустью они сильно напоминают личность их творца» [8: 788].

Использование архаизмов в речи героинь вполне функционально: таким образом персонажам придаются черты утонченности и аристократизма, подчеркивается их принадлежность к «аристократии духа». Возвышенности образов соответствует торжественная архаичность лексики и грамматических форм; мы наблюдаем здесь своего рода стилистическое согласование. Стилистическая цель архаизмов в этих новеллах – поднять тему и возвышенно-прекрасный образ героини над повседневностью, обыденностью.

Проведенный нами анализ примеров из произведений Э. По показывает, что введение архаизма в ткань повествования всегда обусловлено у Э. По логикой развития сюжетной линии: не случайно архаизмы появляются в его рассказах в самые драматичные, кульминационные моменты повествования, связанные с сильными душевными переживаниями персонажей.

Таким образом, архаизмы – это своего рода «вестники из иных миров, сообщающие тексту дополнительное измерение: они либо отсылают слушающего (читателя) к прошлым эпохам, либо (благодаря своей «чуждости») заставляют его активнее воспринимать текст, мобилизуют его эстетические начала» [5: 124]. Последняя функция архаизма реализуется в анализируемых новеллах, полностью согласуясь с эстетической концепцией американского поэта и новеллиста, для которого целью произведения является эмоционально-психологическое воздействие на сознание читателя. Цель поэзии, по Э. По, – «вызвать у читателя эмоциональный подъем, при котором только и возможно мимолетное “прозрение” Высшей Красоты» [6: 575].

В отношении Э. По особенно справедлива идея известного литературоведа и лингвиста Ю. Н. Тынянова о необходимости изучения архаизмов в контексте отдельного произведения, творчества автора в целом и в контексте всей современной автору литературы. «Функция архаизмов, – пишет ученый, – например, целиком зависит от системы, в которой они употреблены. В системе Ломоносова они имеют, например, функцию так называемого “высокого” словоупотребления, так как в этой системе доминирующую роль в данном случае играет лексическая окраска (архаизмы употребляются по лексическим ассоциациям с церковным языком). В системе Тютчева функции архаизмов другие, они в ряде случаев абст-

рактны: фонтан – водомет» [9: 273]. Невозможно определить функцию архаизма, т. е. соотносить его с «высоким» или «ироническим» словоупотреблением вне семантической и интонационной системы конкретного произведения. У Э. По один и тот же архаичный элемент может быть употреблен в абстрактной, «высокой» или иронической функции.

Характерный случай вкрапления архаизмов встречается в новелле «Падение дома Ашеров», где в ткань повествования включаются отрывки, написанные более архаизированным языком, чем остальной текст рассказа:

*“...Who entereth herein, a conqueror hath bin;
Who slayeth the dragon, the shield he shall win.*

*And Ethelred uplifted his mace, and struck upon
the head of the dragon, which fell before him, and
gave up his pesty breath, with a shriek so horrid
and harsh, and withal so piercing, that Ethelred
had fain to close his ears with his hands against the
dreadful noise of it, the like whereof was never
before heard”* [7: 123].

Архаичные явления прослеживаются в этом отрывке на всех языковых уровнях – фонетическом (*hath bin*), лексическом (*withal, fain*), грамматическом (*slayeth, entereth, hath bin*), синтаксическом (*the like whereof was never before heard*), инверсия, устаревшее построение фраз). Слово “*slayeth*” является не только грамматическим архаизмом, это также семантический архаизм, синонимичный современному “to kill”. В рассказе Э. По ссылается на книгу «Безрассудное свидание», якобы принадлежащую перу некоего сэра Ланселота Кеннинга. Поскольку искусствоведами книга с таким названием не была обнаружена, существует предположение, что приводимые По отрывки принадлежат ему самому [8: 754]. Однако попытка свести роль архаичных явлений в этих отрывках к простой стилизации под старину была бы ошибочна. Архаизмы у такого внимательного стилиста, как Э. По, всегда многофункциональны и не поддаются однозначной интерпретации.

Для иллюстрации этого положения обратимся к работе М. Сонме, исследовавшей роль архаизмов в произведениях классика английской романтической литературы С. Колриджа. М. Сонме считает, что использование архаизмов является гораздо более разнообразной и распространенной практикой, чем это обычно признается. Архаизм у М. Сонме – достаточно объемное понятие: «...всё, что относится к старине, может быть использовано и воспринято в качестве архаизма: манера повествования, детали жизнеописания, жанр произведения, организация текста и т. д.» [10: 29]. При этом роль внеязыковых архаизмов оказывается даже более значительной, чем у традиционно выделяемых лингвистических

архаизмов. Последние рассматриваются ею как формальные архаизмы, роль которых сводится к исторической стилизации, приданию тексту привкуса старины.

М. Сонме, исследовавшая архаичный язык «Баллады о старом моряке» С. Колриджа, приходит к следующему выводу: «В игре, в которую играют тексты, архаизмам отводится очень важная роль... архаизмы способны выступать в качестве своеобразных “туннелей”, через которые текст/читатель вводится в иную временную зону; архаизмы являются средством создания “временной интертекстуальности”, через которую ‘текст-сейчас’ и ‘текст-тогда’ соединяются и переплетаются, прочитываясь одновременно, но воспринимаясь как два разных текста» [Ibid.: 29]. Таким образом, отмечает М. Сонме, текст может в одно и то же время интерпретироваться как новейший литературный эксперимент и как предание «старины глубокой».

Проблема экспрессивных возможностей архаизма не пользуется популярностью у исследователей: некоторые ученые склонны нивелировать значение архаизма для раскрытия творческого замысла писателя, недооценивать связь архаизированного языка и содержания произведения. Архаизм слишком часто понимается как средство, ориентированное исключительно на поверхностный уровень читательского восприятия. Обращение автора к архаизмам воспринимается как искусственность языка, вычурность, притворство. Тексты Э. По предостерегают против упрощенно-формального понимания такого сложного выразительного средства, как архаизм. «Архаизм поверхностен (*genuinely superficial*), но эта поверхностность не обязательно предполагает отсутствие творческой глубины» [Ibid.: 29]. Эта мысль М. Сонме, иллюстрирующая творческое кредо английских романтиков, в полной мере относится и к произведениям Э. По.

Примечания

1. Григорьева, А. Д. Об основном словарном фонде и словарном составе русского языка [Текст] / А. Д. Григорьева. Киев: Рад. шк., 1954. 71 с.
2. Ахманова, О. С. Очерки по общей и русской лексикологии [Текст] / О. С. Ахманова. М.: Едиториал УРСС, 2004. 296 с.
3. Петрищева, Е. Ф. Стилистически окрашенная лексика русского языка [Текст] / Е. Ф. Петрищева. М.: Наука, 1984. 222 с.
4. Стилистика английского языка [Текст] / А. Н. Мороховский, О. П. Воробьева, З. В. Тимошенко. Киев: Выща шк., 1991. 289 с.
5. Норман, Б. Ю. Игра на гранях языка [Текст] / Б. Ю. Норман. М.: Флинта: Наука, 2006. 344 с.
6. Ковалев, Ю. В. Эдгар По [Текст] / Ю. В. Ковалев // История всемирной литературы. Т. 6. М., 1989. С. 571–577.
7. По, Э. Избранное [Текст] : сб. / Э. По; на англ. яз.; сост. Е. К. Нестерова. М.: Радуга, 1983.

8. По, Э. А. Полное собрание рассказов [Текст] / Эдгар Аллан По; под ред. А. А. Елистратова, А. Н. Ни-колюкина. М.: Наука, 1970.

9. Тынянов, Ю. Н. О литературной эволюции [Текст] / Ю. Н. Тынянов // Поэтика. История лите-ратуры. Кино. М., 1977. С. 270–281.

10. Sonmez, Margaret J.-M. Archaisms in 'The Rime of the Ancient Mariner' [Text] / Margaret J.-M. Sonmez. Cardiff Corvey: Reading the Romantic Text, Issue 9. 2002. December.

А. А. Свицова

WHEN IN ROME, DO AS ROMANS DO

В нашей статье рассматриваются основные подходы к таким понятиям, как культура, столкнове-ние культур, культурная специфика. Особо отмеча-ется, что задача исследователя состоит в интерпре-тации денотативного или образно-мотивированного аспектов значения языковых знаков в категориях культуры, т. е. в соотношении единиц системы язы-ка с единицами культуры, а главным условием эф-фективности межкультурного общения является взаимопонимание и диалог культур.

Пословица, вынесенная нами в заглавие ста-тьи, по мнению С. Г. Тер-Минасовой [1], являет-ся наиболее иллюстративной в ситуации, когда необходимо избежать конфликта культур. Соот-ношение между цивилизацией и культурой, куль-турой и человеком, культурами и языками раз-личных этносов являются наиболее животрепещу-щими вопросами нашего времени, и вопросы меж-культурного общения волнуют как политиков, так и культурологов, филологов, социологов.

В настоящее время общепринятым является мнение о том что как в культуре, так и в языке каждого народа присутствует универсальное (об-щечеловеческое) и национально-специфическое. Языковая картина мира включает универсальные черты, общие для всего человечества, черты куль-турно-специфичные – некоторую культурную сердцевину, единую для всех членов социальной группы или общности, а также индивидуальные черты, присущие определенной личности [2]. Универсальные значения, одинаково осмысляе-мые всеми людьми в мире или представителями определенных цивилизационных типов, создают почву для межкультурной коммуникации, без них межкультурное взаимопонимание было бы в принципе невозможно. В то же время в любой культуре имеются присущие только ей культур-ные значения, закрепленные в языке, моральных нормах, убеждениях, особенностях поведения и т. п.

СВИЦОВА Анна Альбертовна – кандидат филологи-ческих наук, доцент, и. о. зав. кафедрой германских языков ВятГУ
© Свицова А. А., 2008

Нации равны в том смысле, что являются но-сителями общечеловеческих ценностей, которые создаются первоначально в национальных фор-мах. Язык, будучи одним из основных признаков нации, выражает культуру народа, который на нём говорит, т. е. национальную культуру. Д. О. Добровольский выделяет два принципиаль-но различных понимания национальной специ-фики. В первом случае национально-культурная специфика некоторого явления данного языка оп-ределяется относительно некоторого другого язы-ка. Во втором случае речь идет о представлениях носителей языка о национальной маркированно-сти тех или иных единиц своего языка вне сопо-ставлении с другим языком [3].

Культурная специфика лексических единиц наиболее очевидна. Однако они не исчерпывают всего набора средств, по которым проводится национально-культурная дифференциация язы-ковых картин мира. Это объясняет, почему усво-ение лексического фонда иностранного языка в отрыве от культуры обычно не приводит к кар-динальной трансформации картины мира: ком-муникант лишь применяет иностранные слова к собственному, уже существующему видению дей-ствительности. Современная лексическая семан-тика в своем стремлении очертить сферу значе-ния отдельного слова приходит к необходимости описывать помимо собственно лексической семантики (в ее экстенциональном, денотативном и интенциональном – или сигнификативном – смысле), коннотативную семантику, т. е. «над-страивающиеся» над лексическим значением фо-ново-культурологические, символические, мифо-поэтические и прочие «созначения» – особые «непрямые» употребления слова. «Выучив инст-ранное слово, человек как бы извлекает кусочек мозаики из чужой, неизвестной еще ему до кон-ца картины и пытается совместить его с имею-щейся в его сознании картиной мира, заданной ему родным языком. Неспособность совместить «кусочки мозаики» при межъязыковых контак-тах, визуализировать объект, обозначаемый сло-вом, приводит к неверным референциям и оши-бочным переводам культурных реалий» [4].

На данном этапе развития общества проис-ходят обширные интеграционные процессы, ак-туальность которых неизбежно приводит к сбли-жению и диалогу культур. Люди воспринимают окружающий мир через призму своих языков по-разному, и поэтому способ концептуализа-ции действительности в каждом языке и уни-версален и специфичен. По мнению В. В. Ощеп-ковой, национальное своеобразие каждого на-рода осознается и исследуется при сравнении и контактах с другими народами. На пути реше-ния данной проблемы стоит идеоматичность языков, которая также выражается в паремио-

логии [5]. С этих позиций культурно значимыми оказываются не только языковые единицы, обозначающие культурно маркированные реалии, но и те, в которых культурная информация залегает на более глубинном уровне семантики. Таким образом, основным методом анализа языковых знаков в лингвокультурологии, ведущим способом экспликации заключенной в них культурной информации, другими словами – экспликации их культурной значимости, является процедура соотнесения групп или массивов языковых знаков со знаками (категориями, таксонами) культуры. Именно массивы номинативных единиц, лексических и фразеологических, функционирующих в том или ином языке, манифестируют значимость определенных установок культуры для той или иной лингвокультурной общности. Культурная информация рассеяна в языке, она осознанно или неосознанно воспроизводится носителями языка, употребляющими языковые выражения в определенных ситуациях, с определенными интенциями и с определенной эмотивной модальностью.

Задача исследователя состоит в интерпретации денотативного или образно-мотивированного аспектов значения языковых знаков в категориях культуры, т. е. в соотнесении единиц системы языка с единицами культуры [6]. Главным условием эффективности межкультурного общения является взаимопонимание, диалог культур, терпимость и уважение к культуре партнеров по коммуникации. Как утверждает С. Г. Тер-Минасова, «культура объединяет людей и одновременно отделяет их от других, чужих культур. Иначе говоря, родная культура – это и щит, охраняющий национальное своеобразие народа, и глухой забор, отгораживающий от других народов и культур» [7].

Р. Киплинг поэтично отмечает: «Запад есть Запад, Восток есть Восток. И им не сойтись никогда, пока будут Небо с Землей таковыми, какими их Бог создал». Весь мир делится таким образом на своих, объединенных языком и куль-

турой людей и на чужих, не знающих языка и культуры.

Каждая национальность отражает различные стороны мировой цивилизации, которые могут остаться не выявленными в других культурах. Нации равны в том смысле, что являются носителями общечеловеческих ценностей, которые создаются первоначально в национальных формах. При этом язык, будучи одним из основных признаков нации, выражает культуру народа, который на нём говорит, т. е. национальную культуру. Картина мира этноса кристаллизуется вокруг «центральной зоны», содержащей этнические константы, а также ценности и верования данного общества в свернутом виде, коими, безусловно, являются пословицы. Национально-культурная специфика пословиц функционирует в сознании носителей культуры, и культура заставляет язык акцентировать эту специфику, поэтому изучение пословиц может помочь понять национально значимые ценности, найти наиболее адекватные пути решения проблем, возникающих при контакте культур.

Примечания

1. Тер-Минасова, С. Г. Война и мир языков и культур: вопросы теории и практики [Текст] : учеб. пособие / С. Г. Тер-Минасова. М.: АСТ: Астрель: Хранитель, 2007. 286 с.
2. Леонтьев, А. А. Основы психолингвистики [Текст] / А. А. Леонтьев. М.: Смысл, 1997. 287 с.
3. Добровольский, Д. О. Образная составляющая в семантике идиом [Текст] / Д. О. Добровольский // Вопросы языкознания. 1996. № 1. С. 71–94.
4. Тер-Минасова, С. Г. Язык и межкультурная коммуникация [Текст] / С. Г. Тер-Минасова. М.: Слово/Slovo, 2000. С. 48.
5. Ощепкова, В. В. Культурологические, этнографические и типологические аспекты лингвострановедения [Текст] : автореф. дис. ... канд. филол. наук / В. В. Ощепкова. М., 1995. 32 с.
6. Картушина, Е. А. Гендерные аспекты фразеологии в массовой коммуникации [Текст] : дис. ... канд. филол. наук / Е. А. Картушина. Ижевск, 2003. 175 с.
7. Тер-Минасова, С. Г. Война и мир языков и культур...

Вестник
Вятского государственного гуманитарного университета
Информатика. Математика. Язык

Научный журнал № 5

Подписано в печать 03.12.2008 г.
Формат 60x84 $\frac{1}{8}$. Бумага офсетная. Гарнитура Mysl.
Печать офсетная. Усл. печ. л. 28,0. Тираж 300 экз. Заказ № 970.

Издательский центр ВятГУ
610002, г. Киров, ул. Ленина, 111
(8332) 673-674