

ИНФОРМАТИКА И ОБРАЗОВАНИЕ



www.infojournal.ru

3-2010

*ПОЗДРАВЛЯЕМ
С ПРАЗДНИКОМ
8 МАРТА!*

ISSN 0234-0453

Профильная школа

Информационный и научно-методический журнал

Журнал направлен на систематическое освещение проблем введения и развития профильного обучения в школах страны.

Нормативные документы, информация о ходе и результатах эксперимента по введению профильного обучения, вопросы предпрофильной подготовки и профориентации учащихся, проблемы содержания профильного образования, организации профильного обучения в городах и сельской местности в разных регионах страны, широкий спектр методических рекомендаций и разработок по отдельным профилям и учебным предметам, вопросы единого государственного экзамена, помощь абитуриентам, связь школы и вуза.

Журнал предназначен для руководителей и работников органов управления образованием, директоров образовательных учреждений методистов и учителей.

Подписные индексы в каталогах

Агентства «Роспечать»:
полугодовой – 82390,
годовой – 81239;

Агентства «Почта России»: 24557

Подписка на журнал –
во всех отделениях связи России и СНГ

Профильная школа

В НОМЕРЕ:

Управленческое образование
в профильных классах

Формирование
кадрового ресурса в системе
профильного обучения

Интеграция школы
и учреждения НПО

Телекоммуникационные
проекты в развитии научного
потенциала старшеклассников

Эколого-профильная модель
образования в сельской школе



Подписные индексы в каталогах
Агентства «Роспечать»:
полугодовой – 82390, годовой – 81239;
«Почта России» – 24557

1'2010 (40)

Основные рубрики журнала:

- Официальная информация
- Общественное мнение
- Вопросы теории
- Практика
- Эксперимент
- Элективные курсы
- Зарубежный опыт и другие темы

■ ■ ■

Издается с 2003 г.
Выходит 6 раз в год

■

Адрес редакции:
125212,
Москва, а/я 133

Тел./факс:
(495) 459-13-77

e-mail: info@russmag.ru
<http://russmag.ru>

СОДЕРЖАНИЕ

УЧРЕДИТЕЛИ

Российская Академия
образования

Издательство
«Образование
и Информатика»

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Кузнецов А. А.,
*председатель
редакционной коллегии*

Кравцова А. Ю.,
главный редактор

Бешенков С. А.

Болотов В. А.

Григорьев С. Г.

Жданов С. А.

Кинелев В. Г.

Лапчик М. П.

Роберт И. В.

Семенов А. Л.

Угринович Н. Д.

Христочевский С. А.

ГОТОВИМСЯ К ЕГЭ ПО ИНФОРМАТИКЕ

Малясова С. В. Алгоритмизация, программирование
и технология программирования 3

МЕТОДИКА

Жилин С. А. К вопросу о выборе языка
программирования для изучения в школьном курсе
информатики 13

Дергачева Л. М. Технология создания, хранения,
поиска и сортировки информации в базе данных 18

Моисеева Н. Н. Элективный курс
«Дополнительные возможности форматирования
в документах HTML» 29

ПЕДАГОГИЧЕСКИЕ ИЗМЕРЕНИЯ И ТЕСТЫ

Юнов С. В. Фасетные и псевдофасетные тестовые
задания в измерениях качества информационной
подготовки 35

Сердюков В. И. О количественном оценивании
достоверности результатов автоматизированного
контроля знаний 39

ПЕДАГОГИЧЕСКИЙ ОПЫТ

Заславская О. Ю., Галеева Н. Л. Подходы
к управлению учебной деятельностью учащихся
на уроках информатики 44

Газарян Р. М., Петросян В. Г. От абака через счеты
к компьютеру 50

Щербакова О. В. К вопросам пропедевтической
подготовки по логике школьников второй ступени 53

Кильдишов В. Д. Моделирование роботов
с использованием электронной таблицы MS Excel 56

Невзорова Е. В. Построение треугольника Паскаля
с применением средств языка программирования
QBasic 60

ЗАДАЧИ

Судейманов Р. Р. Занимательные задачи
с историческим материалом по информатике 64

Овчинникова И. Г., Сахнова Т. Н. Особенности изучения
подпрограмм в школьном курсе информатики 72

ИНФОРМАТИКА В НАЧАЛЬНОЙ ШКОЛЕ

Быкова Т. П. Формирование у младших школьников
навыка структурирования информации с помощью
таблиц 78

Баракина Т. В., Поморцева С. В. Изучение
элементов логики и теории множеств в начальном
курсе информатики 82

ТЕОРИЯ И ПРАКТИКА ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ

Мацуца К. И. Медиаобразовательный анализ целей обучения информатике в общеобразовательной школе	97
Васенина Е. А. Эксперимент и исследовательская деятельность на компьютерном уроке информатики	100
Тарнаева С. А. Комплекс программно-педагогических и телекоммуникационных средств по информатике как средство формирования информационной компетентности будущих инженеров в техническом вузе	104
Суханов М. Б., Суханова А. Г. Определение коэффициентов нелинейных зависимостей методом наименьших квадратов в Open Office.org Calc	105
Кудинов В. А. Методические основы использования корпоративных порталов управления знаниями для организации адаптивного обучения	108
Югрин Е. В. Когнитивно-прагматический подход в обучении информатиков-экономистов разработке прикладных решений в системе «1С:Предприятие»	111
Шутикова М. И., Чеснокова И. А. О моделях учебного процесса в системе зачетных единиц	114
Шевчук Е. П. Индивидуальный подход к формированию профессионального самосознания старшеклассников в современном информационном обществе	118
Мусаева Х. С. Некоторые особенности обучения русскому правописанию в национальной школе в условиях информационного общества	120
Гейн А. Г., Кокшарова Е. А. Использование педагогической экспертной системы для оценки качества обучающих тестов	121
Епанчинцева М. В. Формирование психических процессов через комплекс развивающих задач в пропедевтическом курсе информатики	123
Богомазова М. М. Концепции методических подходов к формированию информационной подготовки курсантов военного вуза	125

РЕДАКЦИЯ

Иванова Т. В.,
зам. главного редактора

Дергачева Л. М.

Кириченко И. Б.

Козырева Н. Ю.

Коптева С. А.

Реутова Е. А.

Тарасов Е. В.

Присланные рукописи не рецензируются и не возвращаются.

Редакция не вступает в переписку.

Точка зрения редакции может не совпадать с мнениями авторов публикуемых материалов.

Редакция оставляет за собой право менять заголовки, сокращать тексты статей и вносить в них необходимую стилистическую правку без согласования с авторами.

Ответственность за достоверность фактов несут авторы публикуемых материалов.

Редакция не несет ответственности за содержание рекламных материалов.

Адрес редакции: 125362, Москва, ул. Свободы, дом 35, стр. 39

Телефон: (495) 210-56-89 Факс: (495) 497-67-96 E-mail: readinfo@infojournal.ru

Отдел подписки и распространения: info@infojournal.ru Сайт в Интернете: www.infojournal.ru

Подписано в печать с оригинал-макета 01.03.2010. Формат 70×108^{1/16}. Бумага газетная. Печать офсетная. Усл. печ. л. 11,2. Уч.-изд. л. 13,52. Тираж 2870 экз. Заказ № 0526.

Все права защищены. Никакая часть журнала не может быть воспроизведена в любой форме или любыми средствами, электронными или механическими, включая фотографирование, сканирование, магнитную запись, размещение в Интернете или иные средства копирования или сохранения информации, без письменного разрешения издательства.

Свидетельство о регистрации средства массовой информации ПИ № 77-7065 от 10 января 2001 г.

Отпечатано в ОАО «Московская газетная типография», 123995, Москва, Улица 1905 года, д. 7, стр. 1.

© «Образование и Информатика», 2010



ГОТОВИМСЯ К ЕГЭ ПО ИНФОРМАТИКЕ

С. В. Малясова,

учитель информатики средней общеобразовательной школы № 22,
пос. Беркакит, Нерюнгринский район, Республика Саха (Якутия)

АЛГОРИТМИЗАЦИЯ, ПРОГРАММИРОВАНИЕ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ*

**Тема: «Исполнение алгоритма для конкретного исполнителя
с фиксированным набором команд. Исполнение алгоритма
в среде формального исполнителя»**

В данной теме представлены два варианта заданий контрольно-измерительных материалов ЕГЭ — задания типа А18 и В5.

Теоретический материал (некоторые определения, правила)

Алгоритм — это конечная последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью точных и понятных исполнителю команд.

Алгоритм — это понятное и точное указание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

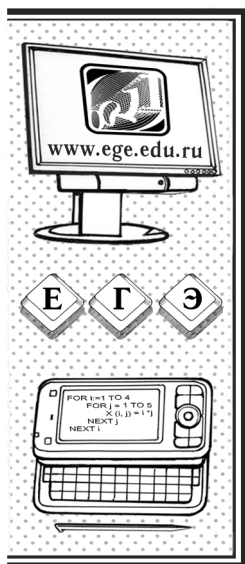
Исполнитель — объект, который выполняет алгоритм.

Идеальным исполнителем алгоритмов является компьютер.

Исполнитель-человек может выполнять алгоритмы формально, не вникая в содержание поставленной задачи, а только строго выполняя последовательность действий, содержащихся в алгоритме.

Одно из принципиальных обстоятельств состоит в том, что исполнитель не вникает в смысл того, что он делает, но получает необходимый результат. В таком случае говорят, что исполнитель действует *формально*, т. е. отвлекается от содержания поставленной задачи и только строго выполняет некоторые правила, инструкции. Это важная особенность алгоритмов. Наличие алгоритма формализует процесс решения задачи, исключает рассуждение исполнителя. Использование алгоритма дает возможность решать задачу формально, механически исполняя команды алгоритма в указанной последовательности. Целесообразность предусматриваемых алгоритмом действий обеспечивается точным анализом со стороны того, кто составляет этот алгоритм.

Достаточно распространенным способом представления алгоритма является его запись на *алгоритмическом языке*, представляющем в общем случае систему обозначений и правил для единообразной и точной записи алгоритмов и исполнения их. Отметим, что между понятиями «алгоритмический язык»



* Продолжение. Начало см.: Информатика и образование. 2009. № 12; 2010. № 1, 2.

и «язык программирования» есть различие: прежде всего, под исполнителем в алгоритмическом языке может подразумеваться не только компьютер, но и устройство для работы «в обстановке». Программа, записанная на алгоритмическом языке, не обязательно предназначена компьютеру. Практическая же реализация алгоритмического языка — отдельный вопрос в каждом конкретном случае.

Алгоритм, записанный на алгоритмическом языке, должен иметь название. Название желательно выбирать так, чтобы было ясно, решение какой задачи описывает данный алгоритм. Для выделения названия алгоритма перед ним записывают служебное слово алг (*алгоритм*). За названием алгоритма (обычно с новой строки) записывают его команды. Для указания начала и конца алгоритма его команды заключают в пару служебных слов нач (*начало*) и кон (*конец*). Команды записывают последовательно.

Алгоритм, в котором действия выполняются последовательно одно за другим, называется *линейным*.

Пример линейного алгоритма:

```
нач  
вперед  
вправо  
вперед  
вперед  
влево  
вперед  
кон
```

Алгоритмы, при исполнении которых порядок следования команд определяется в зависимости от результатов проверки некоторых условий, называют *разветвляющимися*. Для их описания в алгоритмическом языке используют специальную составную команду — *команду ветвления*.

Пример разветвляющегося алгоритма:

```
если слева свободно то  
влево  
иначе  
вниз  
конецесли
```

В указанном примере приведено *полное ветвление*, т. е. ветвление, имеющее две ветви — ветвь ДА (записывается после слова ТО и выполняется, если условие истинно) и ветвь НЕТ (записывается после слова ИНАЧЕ и выполняется, если условие ложно). Возможно в алгоритме использование только одной ветви ДА, тогда ветвление можно назвать *неполным*.

Алгоритмы, при исполнении которых отдельные команды или серии команд выполняются неоднократно, называют *циклическими*. Для организации циклических алгоритмов в алгоритмическом языке используют специальную составную команду *цикла*. В школьном алгоритмическом языке нц обозначает «начало цикла», а кц — «конец цикла»; все команды между нц и кц — это тело цикла, они выполняются несколько раз. Различают циклы с параметром, с постусловием и предусловием.

Цикл с параметром на алгоритмическом языке может выглядеть следующим образом:

```
нц для  $i$  от 1 до  $n$   
серия команд  
кц
```

Эта запись обозначает цикл, в котором переменная i (она называется *переменной цикла*) принимает последовательно все значения от 1 до n с шагом 1.

Пример цикла с параметром:

```

нц для  $i$  от 1 до 5
   $c :=$  Извлечь( $a, i$ )
   $b :=$  Склеить( $b, c$ )
   $i := i - k$ 
кц

```

Цикл с предусловием, или цикл ПОКА, — это цикл, в котором условие проверяется перед выполнением команд. Цикл можно записать в следующем виде:

```

пока условие
нц
  серия команд
кц

```

Пример цикла с предусловием:

```

пока  $i > 0$ 
нц
   $c :=$  Извлечь( $a, i$ )
   $b :=$  Склеить( $b, c$ )
   $i := i - k$ 
кц

```

Цикл с постусловием, или цикл ДО, — это цикл, в котором условие проверяется после выполнения команды. Цикл можно записать в следующем виде:

```

нц
  серия команд
до условие
кц

```

Пример цикла с постусловием:

```

нц
   $c :=$  Извлечь( $a, i$ )
   $b :=$  Склеить( $b, c$ )
   $i := i - k$ 
до  $i > 0$ 
кц

```

В алгоритмах могут быть использованы различные типы данных: целые числа, вещественные числа, символы, строки и др.

Основные операции с числами были рассмотрены в первой теме.

Основные операции с символьными строками: определение длины, выделение подстроки, удаление и вставка символов, «сцепка» двух строк в одну. Например:

Длина(a) — возвращает количество символов в строке a , тип «целое»;

Извлечь(a, i) — возвращает i -й (слева) символ в строке a , тип «строка»;

Склеить(a, b) — возвращает строку, в которой записаны сначала все символы строки a , а затем все символы строки b , тип «строка».

Примеры заданий

Рекомендуемое время на выполнение подобных заданий — 1 мин.

Задача 1 (задание типа A18).

Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известны положение исполнителя и направление его движения. У исполнителя существуют две команды:

Вперед n , вызывающая передвижение Черепашки на n шагов в направлении движения.

Направо m , вызывающая изменение движения на m градусов по часовой стрелке.
Вместо n и m должны стоять целые числа.

Запись:

Повтори 5 [Команда 1 Команда 2]

означает, что последовательность команд в квадратных скобках повторится 5 раз.

Какое число необходимо записать вместо n в следующем алгоритме:

Повтори 7 [Вперед 40 Направо n],

чтобы на экране появился правильный шестиугольник?

Решение.

Примечание. Задания данного типа в КИМах ЕГЭ идут с выбором ответа. Этот пример приведен без вариантов ответов, чтобы учащиеся могли полагаться на логику рассуждения.

Сумма углов правильного шестиугольника равна 360 градусов. Значит, на один угол приходится:

$$360 : 6 = 60 \text{ градусов.}$$

То есть вместо n должно быть записано число 60.

Получается, что задача слишком проста и сводится к тому, что необходимо знать, чему равна сумма углов многоугольника.

Задача 2 (задание типа А18).

Система команд Робота, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх ↑,
вниз ↓,
вправо →,
влево ←.

При выполнении этих команд Робот перемещается на одну клетку соответственно вверх, вниз, вправо, влево.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится Робот:

сверху свободно,
снизу свободно,
слева свободно,
справа свободно.

Цикл

ПОКА <условие> команда

выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, Робот остановится в той же клетке, с которой он начал движение?

- 1) 1
- 2) 2
- 3) 3
- 4) 4

НАЧАЛО

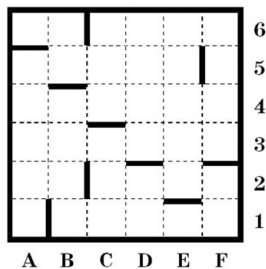
ПОКА <справа свободно> вправо

ПОКА <сверху свободно> вверх

ПОКА <слева свободно> влево

ПОКА <снизу свободно> вниз

КОНЕЦ



Решение.

Способ 1.

Воспользуемся методом «ручной прокрутки», т. е. рассмотрим каждую клетку как исходную и проделаем путь Робота.

Исходная клетка — А1:

НАЧАЛО	А1
ПОКА <справа свободно> вправо	Не выполняется, так как стена справа
ПОКА <сверху свободно> вверх	Движение до А5
ПОКА <слева свободно> влево	Не выполняется, так как стена слева
ПОКА <снизу свободно> вниз	Движение до А1
КОНЕЦ	Робот остался в той же клетке, с которой начал движение, значит, клетка подходит

Исходная клетка — А2:

НАЧАЛО	А2
ПОКА <справа свободно> вправо	Движение до В2, далее встречается стена
ПОКА <сверху свободно> вверх	Движение до В4
ПОКА <слева свободно> влево	Движение А4
ПОКА <снизу свободно> вниз	Движение до А1
КОНЕЦ	Робот остановился в клетке А1, значит, клетка не подходит

Далее рассматривается решение в краткой записи. Первая запись — это исходная клетка, далее после тире действия программы, затем вывод.

А3 — F3 — F6 — С6 — С4 — не подходит

А4 — F4 — F6 — С6 — С4 — не подходит

А5 — E5 — E6 — С6 — С4 — не подходит

А6 — В6 — А6 — подходит (выполняются всего два действия программы, т. е. Робот движется только вправо и обратно. Вверх он двигаться не может, так как находится в верхней крайней точке. Движению вниз ему препятствует стена).

В1 — F1 — F2 — С2 — С1 — не подходит

В2 — В4 — А4 — А1 — не подходит

В3 — F3 — F6 — С6 — С4 — не подходит

В4 — F4 — F6 — С6 — С4 — не подходит

В5 — E5 — E6 — С6 — С4 — не подходит

В6 — А6 — не подходит

С1 — F1 — F2 — С2 — С1 — подходит

С2 — F2 — С2 — С1 — не подходит

С3 — F3 — F6 — С6 — С4 — не подходит

С4 — F4 — F6 — С6 — С4 — подходит

Мы уже получили четыре правильных ответа — это максимальное число, которое предложено в ответе, поэтому можно остановиться. Но задание могло бы быть и без выбора ответа. Поэтому мы проверим все оставшиеся точки и убедимся в правильности нашего решения.

С5 — E5 — E6 — С6 — С4 — не подходит

С6 — F6 — С6 — С4 — не подходит

Д1 — F1 — F2 — С2 — С1 — не подходит

Д2 — F2 — С2 — С1 — не подходит

Д3 — F3 — F6 — С6 — С4 — не подходит

Д4 — F4 — F6 — С6 — С4 — не подходит

Д5 — E5 — E6 — С6 — С4 — не подходит

Д6 — F6 — С6 — С4 — не подходит

Е1 — F1 — F2 — С2 — С1 — не подходит

E2 — F2 — C2 — C1 — не подходит
 E3 — F3 — F6 — C6 — C4 — не подходит
 E4 — F4 — F6 — C6 — C4 — не подходит
 E5 — E6 — C6 — C4 — не подходит
 E6 — C6 — C4 — не подходит
 F1 — F2 — C2 — C1 — не подходит
 F2 — C2 — C1 — не подходит
 F3 — F6 — C6 — C4 — не подходит
 F4 — F6 — C6 — C4 — не подходит
 F5 — F6 — C6 — C4 — не подходит
 F6 — C6 — C4 — не подходит

Ответ. 4.

Способ 2.

Как мы видим, простое исполнение алгоритма с поочередным рассмотрением каждой из клеток в качестве начальной — процедура достаточно трудоемкая.

Попытаемся сузить круг поиска. Поскольку движение Робота должно закончиться в той же клетке, что и началось, а последняя команда алгоритма предполагает возможность движения вниз, то необходимым условием для искомой начальной клетки является наличие границы снизу — эта граница не позволит Роботу спуститься ниже начального положения, если он в него попадет. Таких клеток с границей снизу на рисунке 12:

A1, B1, C1, D1, E1, F1, A6, B5, C4, D3, E2, F3.

Рассмотрим только их:

A1—A5—A1 — подходит

B1 — F1 — F2 — C2 — C1 — не подходит

C1 — F1 — F2 — C2 — C1 — подходит

D1 — F1 — F2 — C2 — C1 — не подходит

E1 — F1 — F2 — C2 — C1 — не подходит

F1 — F2 — C2 — C1 — не подходит

A6 — B6 — A6 — подходит

B5 — E5 — E6 — C6 — C4 — не подходит

C4 — F4 — F6 — C6 — C4 — подходит

D3 — F3 — F6 — C6 — C4 — не подходит

E2 — F2 — C2 — C1 — не подходит

F3 — F6 — C6 — C4 — не подходит

Таким образом, искомых клеток 4, что соответствует варианту ответа номер 4.

Ответ. 4.

Задача 3 (задание типа A18).

В приведенном ниже фрагменте алгоритма, записанном на алгоритмическом языке, переменные a , b , c имеют тип «строка», а переменные i , k — тип «целое». Используются следующие функции:

Длина(a) — возвращает количество символов в строке a . Тип «целое».

Извлечь(a , i) — возвращает i -й (слева) символ в строке a . Тип «строка».

Склеить(a , b) — возвращает строку, в которой записаны сначала все символы строки a , а затем все символы строки b . Тип «строка».

Значения строк записываются в одинарных кавычках. Например, $a := \text{'дом'}$.

Фрагмент алгоритма:

$i := \text{Длина}(a)$

$k := 2$

$b := \text{'А'}$

пока $i > 0$

нц

$c := \text{Извлечь}(a, i)$
 $b := \text{Склеить}(b, c)$
 $i := i - k$

кц

$b := \text{Склеить}(b, 'Т')$

Какое значение будет у переменной b после выполнения вышеприведенного фрагмента алгоритма, если значение переменной a было 'ПОЕЗД'?

- 1) 'АДЕПТ'
- 2) 'АДЗЕОП'
- 3) 'АДТЕТПТ'
- 4) 'АДЗОТ'

Решение.

Способ 1.

В данном алгоритме используется цикл с предусловием. Чтобы определить, сколько раз будут выполняться действия внутри цикла, необходимо вычислить значение переменной i .

Выполним «прокрутку» алгоритма.

1. Значение переменной $a = \text{'ПОЕЗД'}$.
 2. Значение переменной i вычисляется по формуле $i := \text{Длина}(a)$. В слове ПОЕЗД 5 букв, значит, первоначальное значение переменной $i = 5$.

3. Значение переменной $k = 2$.

4. Первоначальное значение переменной b строкового типа равно 'А'.

5. Проверяется условие цикла с предусловием: значение переменной $i > 0$. Так как $5 > 0$, значит, выполняются действия внутри цикла:

- значение переменной c вычисляется по формуле $c := \text{Извлечь}(a, i)$, т. е. необходимо из слова ПОЕЗД взять 5-й символ — это символ Д, значит, $c = \text{'Д'}$;
- значение переменной b вычисляется по формуле $b := \text{Склеить}(b, c)$, т. е. необходимо в новую строку сначала записать первоначальное значение переменной b , а затем последнее значение переменной c , получаем новое значение переменной $b = \text{'АД'}$;
- новое значение переменной i вычисляется по формуле $i := i - k$, значит, $i = 5 - 2 = 3$.

6. Проверяется условие цикла: значение переменной $i > 0$. Так как $3 > 0$, значит, выполняются действия внутри цикла:

- значение переменной c вычисляется по формуле $c := \text{Извлечь}(a, i)$, т. е. необходимо из слова ПОЕЗД взять 3-й символ — это символ Е, значит, $c = \text{'Е'}$;
- значение переменной b вычисляется по формуле $b := \text{Склеить}(b, c)$, т. е. необходимо в новую строку сначала записать последнее значение переменной b — это 'АД', а затем последнее значение переменной c , получаем новое значение переменной $b = \text{'АДЕ'}$;
- новое значение переменной i вычисляется по формуле $i := i - k$, значит, $i = 3 - 2 = 1$.

7. Проверяется условие цикла: значение переменной $i > 0$. Так как $1 > 0$, значит, выполняются действия внутри цикла:

- значение переменной c вычисляется по формуле $c := \text{Извлечь}(a, i)$, т. е. необходимо из слова ПОЕЗД взять 1-й символ — это символ П, значит, $c = \text{'П'}$;
- значение переменной b вычисляется по формуле $b := \text{Склеить}(b, c)$, т. е. необходимо в новую строку сначала записать последнее значение переменной b — это 'АДЕ', а затем последнее значение переменной c , получаем новое значение переменной $b = \text{'АДЕП'}$;
- новое значение переменной i вычисляется по формуле $i := i - k$, значит, $i = 1 - 2 = -1$.

8. Проверяется условие цикла: значение переменной $i > 0$. Так как $-1 < 0$, условие не выполняется, значит, выполняются действия после цикла: $b := \text{Склеить}(b, 'Т')$.

Необходимо в новую строку сначала записать последнее значение переменной b — это 'АДЕП', а затем букву Т, получаем новое значение переменной $b = \text{'АДЕПТ'}$.

Ответ. 1.

Способ 2.

Представленную выше последовательность действий можно кратко записать в виде таблицы:

a	i	k	b	Цикл			
				Проверка условия	c	b	$i = i - k$
ПОЕЗД	5	2	А	$5 > 0$ (да)	Д	АД	3
	3	2	АД	$3 > 0$ (да)	Е	АДЕ	1
	1	2	АДЕ	$1 > 0$ (да)	П	АДЕП	-1
	-1	2	АДЕП	$-1 > 0$ (нет), цикл закончен			
			АДЕПТ				

Ответ. 1.

Способ 3.

Данный способ основывается на логических рассуждениях.

1. Последнее действие алгоритма показывает, что последняя буква в полученной строке будет 'Т', значит, вариант ответа 2 не подходит.

2. Цикл будет «прокручиваться» 3 раза, так как значение переменной i вычисляется по формуле $i := i - k$, а $k = 2$. То есть значений переменной i , удовлетворяющих условию, будет всего 3 — это 5, 3 и 1.

3. Из предыдущего пункта можно сделать вывод, что переменная c будет принимать значения Д, Е, П. Поэтому можно исключить варианты ответов 3 и 4, так как там значение переменной c во второй «прокрутке» цикла равно Т, З.

Ответ. 1.

Задача 4 (задание типа В5).

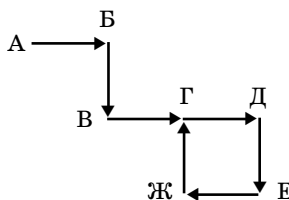
Исполнитель Робот действует на клетчатой доске, между соседними клетками которой могут стоять стены. Робот передвигается по клеткам доски и может выполнить команды 1 (вверх), 2 (вниз), 3 (вправо), 4 (влево), переходя на соседнюю клетку в направлении, указанном в скобках. Если в этом направлении между клетками стоит стена, то Робот разрушается. Робот успешно выполнил программу вида: 3233241.

Какую последовательность из трех команд должен выполнить Робот, чтобы вернуться в ту клетку, где он был перед началом выполнения программы, и не разрушиться вне зависимости от того, какие стены стоят на поле?

Решение.

Задания подобного типа представлены без вариантов ответов, т. е. это задания с кратким ответом, которые могут быть представлены в виде той или иной последовательности (цифр, символов).

Для решения данной задачи нарисуем путь, который проделал Робот:



Получили ломаную линию АБВГДЕЖГ. То есть Робот остановился в точке Г. Для того чтобы ему вернуться в точку А, при этом не столкнуться со стеной и не разрушиться, необходимо воспользоваться частью успешно проделанного пути. Из точки Г в точку А можно проделать следующий путь: влево, вверх, влево. Заменяем команды их числовым значением и получим последовательность 414.

Ответ. 414.

Задача 5 (задание типа В5).

У исполнителя Утроитель две команды, которым присвоены номера:

1. вычти 1
2. умножь на 3

Первая из них уменьшает число на экране на 1, вторая — увеличивает его в 3 раза. Запишите порядок команд в программе получения из числа 3 числа 16, содержащей не более 5 команд, указывая лишь номера команд.

Например, программа 21211 — это программа:

умножь на 3
вычти 1
умножь на 3
вычти 1
вычти 1

которая преобразует число 1 в 4.

Решение.

Способ 1.

Воспользуемся методом подбора в прямой последовательности, построим цепочки из математических действий:

$3 \cdot 3 = 12 \rightarrow 12 \cdot 3 = 36 \rightarrow 36 - 1 = 35 \rightarrow 35 - 1 = 34 \rightarrow 34 - 1 = 33$ — цепочка будет содержать более пяти команд;

$3 - 1 = 2 (1) \rightarrow 2 \cdot 3 = 6 (2) \rightarrow 6 \cdot 3 = 18 (2) \rightarrow 18 - 1 = 17 (1) \rightarrow 17 - 1 = 16 (1)$ — цепочка содержит пять команд.

Примечание. В скобках указаны номера команд.

Ответ. 12211.

Способ 2.

Подберем решение, составляя обратную последовательность, т. е. из 16 получим 3. Понятно, что к умножению должно быть применено противоположное действие — деление, а к вычитанию — сложение, и ответ должен быть записан в обратной последовательности:

- 16 не делится на 3, значит, первое действие — это действие, обратное вычитанию, — сложение: $16 + 1 = 17 (1)$;
- 17 не делится на 3, значит, здесь тоже будет сложение: $17 + 1 = 18 (1)$;
- 18 делится на 3: $18 : 3 = 6 (2)$;
- 6 делится на 3: $6 : 3 = 2 (2)$;
- увеличиваем последний результат на 1: $2 + 1 = 3 (1)$.

Записываем полученную цепочку действий в обратной последовательности:

12211.

Примечание. В скобках указаны номера команд.

Ответ. 12211.

Примечание. Какой из этих способов лучше? Наверно, выбрать нужно тот, в котором вы больше уверены. Но решение в обратной последовательности поможет однозначно определить первое действие, следовательно, и цепочка будет сразу подобрана.

Литературные и интернет-источники

1. *Босова Л. Л.* Линия алгоритмизации: от пропедевтики к ЕГЭ // Информатика и образование. 2009. № 3.
2. *Гусева И. Ю.* ЕГЭ. Информатика: раздаточный материал тренировочных тестов. СПб.: Тригон, 2009.
3. *Зорин М. В., Зорина Е. М.* Информатика. Тестирование в формате ЕГЭ. Волгоград: Учитель, 2009.
4. *Крылов С. С., Лецинер В. Р., Якушкин П. А.* Единый государственный экзамен 2007. Информатика: Учебно-тренировочные материалы для подготовки учащихся. ФИПИ. М.: Интеллект-Центр, 2007.
5. *Молодцов В. А., Рыжкова Н. Б.* Информатика: тесты, задания, лучшие методики. Ростов н/Д: Феникс, 2009.
6. *Сафронов И. К.* Готовимся к ЕГЭ по информатике. СПб.: БХВ-Петербург, 2009.
7. *Ярцева О. В., Цикина Е. Н.* ЕГЭ-2009: Самые новые задания. М.: АСТ; Астрель, 2009.
8. <http://www1.ege.edu.ru/content/view/21/43/> — Демонстрационные варианты ЕГЭ 2004—2010 гг.

Тренировочные упражнения и задания для самостоятельного решения по данной теме будут представлены в следующем выпуске журнала «Информатика и образование».

Продолжение следует

НОВОСТИ ЦИФРОВОГО МИРА

«И так близка — дорога в облака...»

Вот уже несколько лет пользователи Интернета привыкают, и довольно успешно, к общедоступным вычислениям в сетевых облаках, подключаясь к централизованно управляемым приложениям и сервисам Amazon, Facebook, Google и других провайдеров. Теперь начинается корпоративная фаза облачной обработки данных — этап строительства так называемых Private Clouds, т. е. защищенных информационных пространств для сетевых сервисов. Облачная обработка данных, ставшая возможной благодаря виртуализации программного обеспечения, вызвала самый значительный прорыв в области информационной инфраструктуры и архитектуры систем и приложений со времен коммерциализации Интернета. Этот подход может кардинально изменить всю область информационных технологий.

Что же такое облачные вычисления? Попросту говоря, это виртуальный доступ к программам через сеть. Как и общедоступные вычисления в сетевых облаках, Private Clouds обеспечивают повсеместный доступ к таким программам, но при этом используют лишь проверенные сервисы, которые находятся под надежным контролем предприятия. Эффективность этих сервисов достигается за счет использования виртуализации, с помощью которой информационные ресурсы и услуги отделяются от физической компьютерной инфраструктуры. Это позволяет предприятиям использовать сеть как платформу для безопасного доступа ко всему, что им требуется для ведения бизнеса (т. е. к инфраструктуре, различным платформам и программам) по первому требованию и в нужном объеме. Результат — огромная экономия и более простое управление центрами обработки данных (ЦОД).

Подобно тому как виртуализация серверов позволила рассматривать приложения как работающие на собственном сервере, сеть позволил предприятиям иметь дело как бы с одним ЦОД, хотя в действительности задания по обработке данных могут выполняться инфраструктурой, принадлежащей разным провайдерам либо самим предприятиям.

Преимущества обработки данных в сетевых облаках громадны. В случае необходимости корпоративные пользователи получают возможность мгновенно привлечь либо «высвободить» ресурсы, оплачивая при этом только те из них, которые использовались фактически. А в самом сетевом облаке многопользовательская среда позволяет провайдерам эффективно применять находящуюся в общем пользовании физическую инфраструктуру для предоставления услуг.

(Обзор подготовлен по материалам журнала «Мир ПК»)



МЕТОДИКА

С. А. Жилин,

учитель информатики и ИКТ средней общеобразовательной школы № 887, Москва

К ВОПРОСУ О ВЫБОРЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ ИЗУЧЕНИЯ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Языки программирования бывают разные.

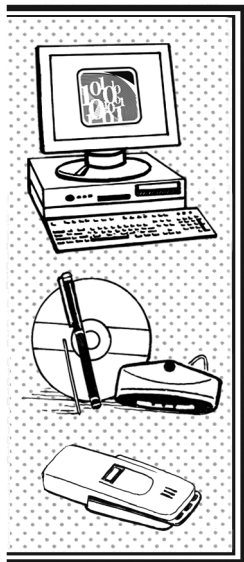
Бывают очень «строгие» языки, языки-тираны — такие, как Pascal, Ada, Eiffel, Oberon, Java. Эти языки не пропустят ни одной ошибки программиста, которую можно обнаружить на этапе компиляции. Попробуйте забыть определить тип переменной или передать процедуре параметр, тип которого не соответствует сигнатуре данной процедуры, т. е. ее формальному определению, — дальше компиляции дело не пойдет.

Но не все языки являются «строгими». Бывают языки «нестрогие», иногда настолько, что хочется называть их «добрыми». Эти языки дают больше свободы, разрешают программисту экономить время. Одни языки разрешают вообще не объявлять типы данных, другие не обращают внимания на преобразования этих типов: во время работы программы строка может превратиться в число, действительное число — в целое и т. д. К этой группе языков относятся многие языки сценарного программирования, например JavaScript (ECMAScript), VBScript, PHP. Из языков общего назначения самым «добрым» является язык Visual Basic.

Вот тут мы и подошли к главному вопросу. Понятно, что для работы нужны разные языки, выбор в каждом случае конкретного языка зависит от типа решаемой задачи, квалификации программиста и т. д. Но *какие языки лучше подходят для обучения, особенно начального? «Добрые» или «строгие»?*

Вопрос не простой. Существует мнение, что «добрые» языки проще, а значит, и учиться на них легче. Возможно, это одна из причин, по которой во многих учеб-

никах информатики язык Visual Basic рассматривается как основное и единственное средство для обучения основам современного программирования. Но уже давно специалисты по программированию заметили: в этой области тоже действует принцип «за все надо платить». Хочешь больше свободы — будь готов к тому, что ошибок в программе будет много и исправлять их придется дольше, чем при работе со «строгими» языками. В начале своего развития язык Basic давал очень много свободы, программы получались запутанные, с трудно читаемой структурой, ошибки можно было допустить очень легко. И мы прекрасно понимаем чувства известного голландского математика и программиста Э. В. Дейкстры, который прокомментировал ситуацию такими словами: «Студентов, ранее изучавших Basic, практически невозможно обучить хорошему стилю программирования. Как потенциальные программисты они умственно искалечены без надежды на исцеление». Сейчас можно услышать, что всё это в прошлом, новые версии языка избавлены от всех недостатков и могут учить хорошему стилю написания программ не хуже других языков программирования. Однако реальная учебная практика говорит о другом.



Автор статьи был свидетелем одной интересной ситуации, которая произошла на уроке информатики. Ученики решали простую учебную задачу с помощью табличного процессора Microsoft Excel. Надо было написать процедуру на языке VBA. Эта процедура должна была вводить два числа с помощью объектов типа *TextBox* и вычислять квадратный корень из суммы этих чисел.

С созданием визуальных элементов управления ученики справились легко. А вот результат работы процедуры у нескольких человек получился, мягко говоря, неожиданным. Представьте себе, что вы ввели в поля редактирования две двойки и получили вместо корня из четырех число 4,69041575982343. Откуда же появился такой результат? Ученики были растеряны и никак не могли понять, какую ошибку они допустили в такой простой программе. Особенно сбивало с толку, что ошибок трансляции не было ни одной, следовательно, искать нужно было ошибку «стадии выполнения». И они ее искали. И естественно, не могли найти, потому что ошибка была именно формальная. И была эта ошибка из той области, в которой язык VBA, как и любой другой язык из этого семейства, ничего не может сделать и ничему не может научить. В науке программирования это называется «несогласованным и неявным преобразованием типа данных».

Для полной ясности приведем текст процедуры и выполним краткий анализ ситуации, в которую попали школьники.

```
Private Sub CommandButton1_Click()  
    Dim x As Double  
    x = Math.Sqrt(TextBox1.Text + TextBox2.Text)  
    TextBox3.Text = x  
End Sub
```

Теперь любому профессионалу понятен смысл ошибки. Ученики передали функции *Sqr* из объекта *Math* сумму значений, которые введены в поля *Text* объектов *TextBox1* и *TextBox2*. Но при этом они не подумали, что эти значения являются числами только по внешнему виду. Тип свойства *Text* — это строка символов. Таким образом, параметром функции *Sqr* оказалась сумма двух строк, т. е. результат их конкатенации. При вводе числа 2 в поле редактирования значением свойства *Text* будет строка '2', а результат соединения двух строк с таким значением — это строка '22'. Именно из числа 22 функция *Sqr* извлекла квадратный корень.

Человеку свойственно ошибаться, и в этой ошибке нет ничего необычного. Но в данной ситуации удивляет другое. Как могла стандартная математическая функция языка так легкомысленно отнестись к собственному формальному определению?

Давайте посмотрим на сигнатуру этой функции, тем более язык VBA ее и не скрывает:

```
Sqr(Number As Double) As Double
```

Данная сигнатура означает, что тип параметра у этой функции — действительное число двойной точности. И такой же тип у возвращаемого значения. Таким образом, получив параметр совсем другого типа, функция и не думает предупредить программиста о явном несоответствии. Вместо этого начинается тайная работа по преобразованию типа. Строка символов преобразуется в число, из числа извлекается квадратный корень, пользователь начинает искать причину ошибки.

Может ли что-нибудь подобное случиться при использовании языка со строгим контролем типов? Разумеется, нет. Ошибка будет немедленно обнаружена компилятором. И никакого арифметического «чуда» не будет. Ошибки в программе, возможно, останутся, но это будут только те ошибки, которые нельзя обнаружить до запуска программы.

После этого случая было решено продолжить исследование «особенностей» языков этого семейства. Для продолжения эксперимента мы использовали язык Visual Basic 2008 Express Edition. Для проверки были использованы те же вычисления. На этот раз «ошибки» были сделаны сознательно. Кроме того, результат работы функции был присвоен переменной целого типа. Текст процедуры получился следующим:


```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object,
                          ByVal e As System.EventArgs)

Handles Button1.Click
    Dim x As Integer
    x = Math.Sqrt(TextBox1.Text + TextBox2.Text)
    TextBox3.Text = x
End Sub
End Class
```

Компилируем, запускаем. Снова ни одной ошибки! По мнению компилятора просто отличный код. Для проверки вводим два числа в поля формы, например 2 и 2. Нажимаем кнопку и получаем результат. Число 5 (!?). Здесь уже язык допустил две погрешности. Прежде всего, не проверил соответствие типа параметра. Хотя сигнатура функции утверждает, что параметр должен иметь действительный тип, т. е. Double:

```
Public Shared Function Sqrt(d As Double) As Double
```

Но дальше еще интереснее. Мы «по ошибке» присвоили действительное значение корня переменной целого типа (x). Но дробные разряды в целое число «не влезают»! И Basic их просто отбросил! Да еще и с округлением. И при этом нас опять ни о чем не предупредили. Сами виноваты, следите за типами данных. Не понятно только одно: почему мы должны выполнять работу компилятора?

Проблема возникла из-за отсутствия явного преобразования типов. Попробуем использовать это преобразование. С помощью функции *Val* преобразуем строки символов в числовые значения и передадим их сумму функции вычисления корня. Теперь наша программа будет иметь следующий вид:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object,
                          ByVal e As System.EventArgs)

Handles Button1.Click
    Dim x As Integer
    x = Math.Sqrt(Val(TextBox1.Text)+Val(TextBox2.Text))
    TextBox3.Text = x
End Sub
End Class
```

Теперь вроде всё работает правильно. С учетом преобразования функция *Sqrt* получает число 4 в качестве параметра и корень из этого числа равен 2. Но не всё так просто. Оказывается, что функция *Val* — тоже очень «добрая» функция. Проверить это очень легко. Введем в поля редактирования что-нибудь, что числом не является. Например, букву «А» или «Я». Нажимаем кнопку и получаем результат 0 (нуль!). Не сообщение об ошибке преобразования типа, а просто нуль. Это значит, что функция *Val* получила в качестве параметра символ буквы и спокойно преобразовала его в число. Получается, что для этой функции преобразование любой буквы алфавита в значение целого числа — обычная математическая операция с точно определенным нулевым результатом. И никакой причины для создания исключительной ситуации.

Кто-то может сказать, что всё не так страшно. Такие глупые ошибки в своих программах могут допускать только новички в программировании. А потом они начинают четко следить за преобразованием типов и контролировать его. И никогда в жизни не смогут передать функции значение параметра, которое приведет к некорректному преобразованию.

Вряд ли. Чтобы это контролировать и не допускать, надо этому учиться! А язык этому как раз учить и не собирается!

Всё познается в сравнении. Сделаем попытку так же неаккуратно поработать с другим языком программирования. Например, с языком Delphi (Object Pascal).

Напишем процедуру, которая будет делать то же самое, что и в предыдущем эксперименте, — вычислять корень квадратный из суммы двух действительных чисел.

Сначала попробуем «забыть» о контроле типов — напишем процедуру примерно в том же стиле, что и раньше:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Edit3.Text:=sqrt(Edit1.Text+Edit2.Text)
end;
```

Запускаем компиляцию и сразу получаем сообщение об ошибке. В нашей процедуре не согласованы два типа: строковый (string) и тип с плавающей запятой (Extended):

```
E2010 Incompatible types: 'string' and 'Extended'
```

Теперь запишем программу правильно. Процессы преобразования типов берем под контроль и выполняем их с помощью функций *StrToFloat* и *FloatToStr*:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  a, b: real;
  x: string;
begin
  a:=StrToFloat(Edit1.Text);
  b:=StrToFloat(Edit2.Text);
  x:=FloatToStr(sqrt(a+b));
  Edit3.Text:=x
end;
```

Теперь всё работает и корень из 4 равен 2. Как положено. Но самое интересное в этой программе другое! Мы просто не можем ничего напутать с типами. Все наши попытки обмануть язык обречены. Процедура *sqrt* не может принять в качестве параметра что-либо, кроме числа. Соответственно, и результат процедуры является числом с плавающей запятой и не может быть присвоен переменной целого или другого несовместимого типа. Язык со строгим контролем типов защищает нас и от потери дробной части числа, и от явной глупости — нельзя вычислять квадратные корни из того, что не является числом.

Использование нестрогих языков в промышленном программировании нельзя механически переносить в сферу образования. Программисты-профессионалы знают все слабые места этих языков, знают, как избежать ловушек, аналогичных той, с которой мы разбирались. Но школьник, который только начинает осваивать эту трудную науку, не обладает опытом и знаниями квалифицированного программиста. Ему надо последовательно и грамотно прививать основы того искусства, которое Дейкстра называл «дисциплиной программирования». Основой этого искусства является аккуратная работа с типами данных.

Ошибки, связанные с некорректным и неявным преобразованием типа, составляют наиболее распространенную группу риска в современном программном обеспечении. Французская ракета «Ариан» потерпела аварию из-за подобной ошибки в программном обеспечении бортового компьютера.

Что же мы видим в современном школьном образовании? Некоторые учебные пособия позиционируют Visual Basic почти как основной язык, который призван сформировать у школьников базовые знания и умения в области современного программирования. Но все рассмотренные примеры очень убедительно показывают: языки этого семейства в принципе не могут научить никакой «дисциплине программирования». Очевидные погрешности программиста при обращении к стандартной функции могут привести к неправильной работе программы. При этом средства языка не позволяют обнаружить ошибку ни на стадии компиляции, ни на стадии выполнения.

Можно ли сформировать навыки внимательного отношения к согласованию и преобразованию типов с помощью языков, которые сами не относятся к этому с должным вниманием? Можно ли при таком невнимательном отношении к типизации данных научить школьника соблюдать и уважать правила надежного програм-

мирования? Ответ очевиден: разумеется, нет. Искусству составления правильных и надежных программ необходимо учить только с помощью таких же надежных языков и технологий программирования. К сожалению, за 45 лет эволюции надежность языка Basic существенно не изменилась. И при создании школьных курсов информатики не следует об этом забывать.

Литература

1. *Вирт Н.* Преподавание информатики: потерянная дорога: Приветствие на открытии Международной конференции по преподаванию информатики ITiCSE, г. Аархус (Дания), 24 июня 2002 г.

2. *Пешио К.* Никлаус Вирт о культуре разработки ПО // Открытые системы. 1998. № 1.

3. *Edsger W. Dijkstra.* Selected Writings on Computing: A Personal Perspective, N. Y.: Springer-Verlag, 1982.



ФАКУЛЬТЕТ ИНФОРМАТИКИ Уральского государственного педагогического университета

www.uspu.ru

реализует специальности и направления:

- ◆ **«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ»:**
квалификация «Инженер»,
специализация
 - «Телекоммуникационные технологии в образовании»;
- ◆ **«ИНФОРМАТИКА»:**
квалификация «Учитель информатики»,
специализации:
 - «Компьютерные игровые технологии»,
 - «Web-технологии в образовании»,
 - «Защита информации и информационная безопасность»,
 - «Компьютерная графика и дизайн»;
- ◆ профиль «ИНФОРМАТИКА»,
направление «Физико-математическое образование»,
степени:
 - бакалавр физико-математического образования,
 - магистр физико-математического образования
(магистерская программа «Информатика в образовании»).

Формы обучения: очная и заочная; бюджетная и договорная.

Факультет информатики:

Адрес: 620015, г. Екатеринбург, ул. К. Либкнехта, 9

Телефоны: (343) 371-35-27, 371-16-95

Интернет-адрес: www.inform.uspu.ru

Приемная комиссия:

Адрес: 620017, г. Екатеринбург, пр. Космонавтов, 26, каб. 152

Телефон: (343) 336-12-43

Л. М. Дергачева,

*канд. пед. наук, доцент кафедры информатики и прикладной математики
Московского городского педагогического университета*

ТЕХНОЛОГИЯ СОЗДАНИЯ, ХРАНЕНИЯ, ПОИСКА И СОРТИРОВКИ ИНФОРМАЦИИ В БАЗЕ ДАННЫХ*

После выполнения предложенных работ необходимо отметить, что очень часто при формировании БД сложно ограничиться одной таблицей. Данное положение можно проиллюстрировать примерами из жизни. В частности, БД «Классный журнал» — это несколько взаимосвязанных таблиц по различным предметам (рис. 1—3).

Фамилия	Самостоятельная работа	Домашнее задание	Диктант	Итоговая работа
Добряков	5	5	5	5
Мартышкин	3	2	3	2
Тимощук	5	5	5	5
...				

Рис. 1. Таблица «Русский язык»

Фамилия	Самостоятельная работа	Домашнее задание	Диктант	Итоговая работа
Добряков	5	5	5	5
Мартышкин	3	2	3	2
Тимощук	5	5	5	5
...				

Рис. 2. Таблица «Информатика»

Фамилия	Самостоятельная работа	Домашнее задание	Диктант	Итоговая работа
Добряков	5	5	5	5
Мартышкин	3	2	3	2
Тимощук	5	5	5	5
...				

Рис. 3. Таблица «Математика»

Можно ли из них сформировать одну таблицу? Да, но она получится достаточно громоздкой, поскольку содержит дополнительное поле «Предмет» (рис. 4).

Фамилия	Самостоятельная работа	Домашнее задание	Диктант	Итоговая работа	Предмет
Добряков	5	5	5	5	Русский язык
Мартышкин	3	2	3	2	Русский язык
Тимощук	5	5	5	5	Русский язык
Добряков	5	5	5	5	Информатика
Мартышкин	3	2	3	2	Информатика
Тимощук	5	5	5	5	Информатика
Добряков	5	5	5	5	Математика
Мартышкин	3	2	3	2	Математика
Тимощук	5	5	5	5	Математика
...					

Рис. 4. Таблица «Предметы»

* Продолжение. Начало см.: Информатика и образование. 2010. № 1, 2.

Далее нужно перейти к рассмотрению понятия «*нормализация БД*» — другими словами, оптимизация, избавление от недостатков.

Следует отметить, что в результате нормализации обычно получается *многотабличная БД*. Учащимся можно предложить для самостоятельного выполнения задания по нормализации БД, после чего обсудить с ними, насколько эффективно она произведена.

Затем необходимо рассмотреть вопросы установления *связей между таблицами, обеспечения целостности данных*, а также понятия *базовой и подчиненной таблиц* на конкретных примерах. При этом сделать акцент на том, что:

- записи называются *связанными*, если значение в связанном поле записи базовой таблицы совпадает со значением в связанном поле подчиненной таблицы;
- если связанное поле подчиненной таблицы содержит только уникальные значения, будет создана связь «один к одному»;
- если связанное поле подчиненной таблицы содержит или может содержать повторяющиеся значения, то создается связь «один ко многим».

Закрепление данных вопросов возможно в ходе выполнения практического задания.

6. Практическое задание: «Многотабличная БД. Установление связей между таблицами»

1. Создать таблицу *Учащиеся*, воспользовавшись **Конструктором**.

Поля базы данных должны соответствовать следующей таблице (рис. 5):

Имя поля	Тип данных	Размер поля
Номер	Текстовый	5
Фамилия	Текстовый	15
Имя	Текстовый	10
Отчество	Текстовый	15
Дата рождения	Дата/время	Краткий формат даты
Класс	Текстовый	3
Дом. адрес	Текстовый	20

Рис. 5

2. Определить первичный ключ для таблицы. В данной таблице ключевым является поле «Номер». Чтобы сделать поле ключевым, необходимо выделить его и выбрать команду меню **Правка, Ключевое поле** или щелкнуть на кнопке **Ключевое поле** на панели инструментов. При этом слева от имени ключевого поля появится изображение ключа.

3. Занести в таблицу не менее 6 записей.

4. Создать таблицы *Экзамены* (рис. 6) и *Результаты* (рис. 7).

Состав полей и их свойства представлены на рис. 6, 7.

Таблица *Экзамены*:

	Имя поля	Тип данных	Размер поля
Ключевое поле	Номер	Текстовый	5
	Оценка1	Числовой	Длинное целое
	Оценка2	Числовой	Длинное целое
	Оценка3	Числовой	Длинное целое
	Оценка4	Числовой	Длинное целое
	Результат	Текстовый	5

Рис. 6

Таблица *Результаты*:

	Имя поля	Тип данных	Размер поля
Ключевое поле	Результат	Текстовый	5
	Процент	Числовой	Действительное

Рис. 7

5. Заполнить таблицы данными, записи ввести на свое усмотрение так (количество записей сделать одинаковым в первых двух таблицах), чтобы в них присутствовали разные комбинации оценок из четырех групп (рис. 8):

Неуд	Удовл	Хор	Отл
За 3 и 2	За все 3	За одну 4 и более	5555

Рис. 8

6. В поле «Результат» данные заносить в соответствии с представленной таблицей. Например, если в записи три оценки «5» и одна оценка «4», то в результат занести «Хор».

7. Поле «Процент» заполнить в соответствии со следующей таблицей (рис. 9):

Результат	Процент
Неуд	0,00
Удовл	50,00
Хор	75,00
Отл	100,00

Рис. 9

8. Сохранить таблицы и закрыть их.

9. В окне **Школа: база данных** должны быть имена трех таблиц: *Учащиеся*, *Экзамены*, *Результаты* (рис. 10).

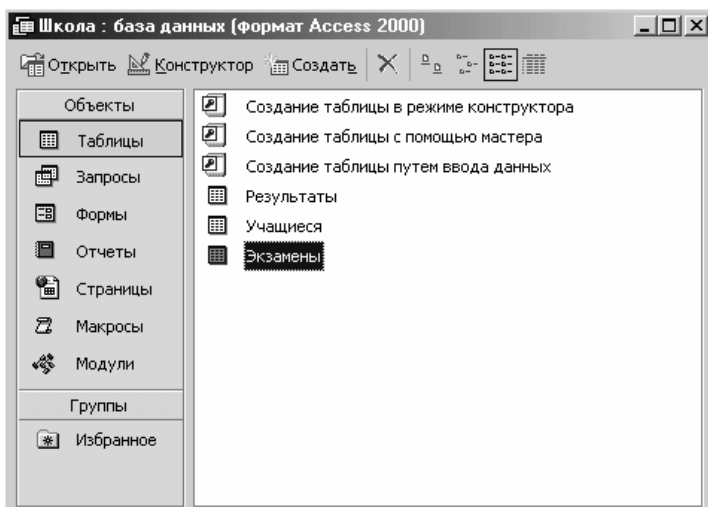


Рис. 10

10. В появившемся окне **Схема данных** выполнить добавление всех трех таблиц в схему (рис. 11, 12).

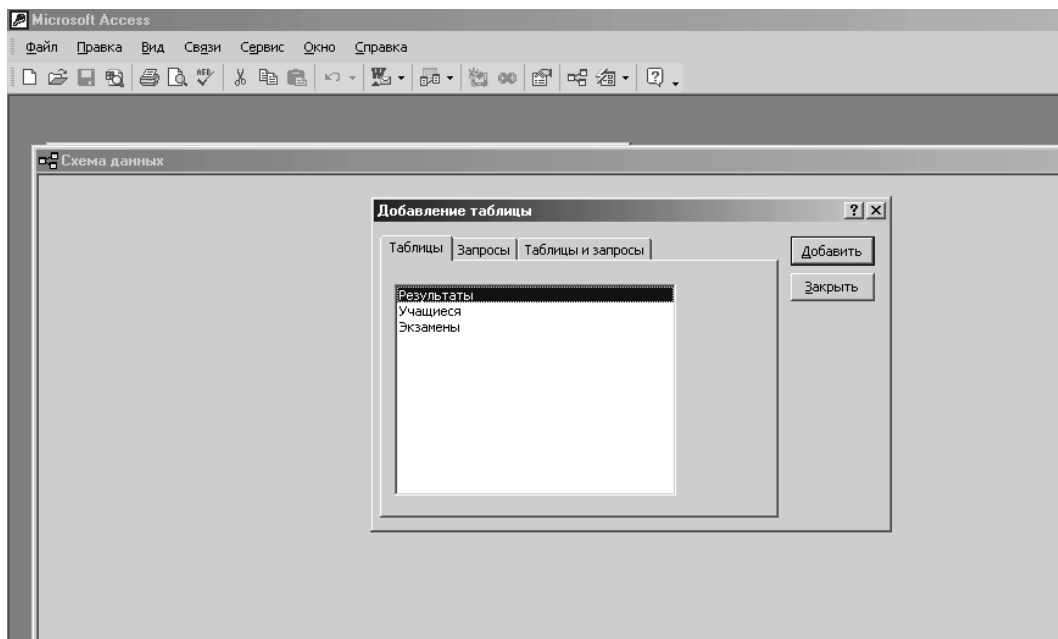


Рис. 11

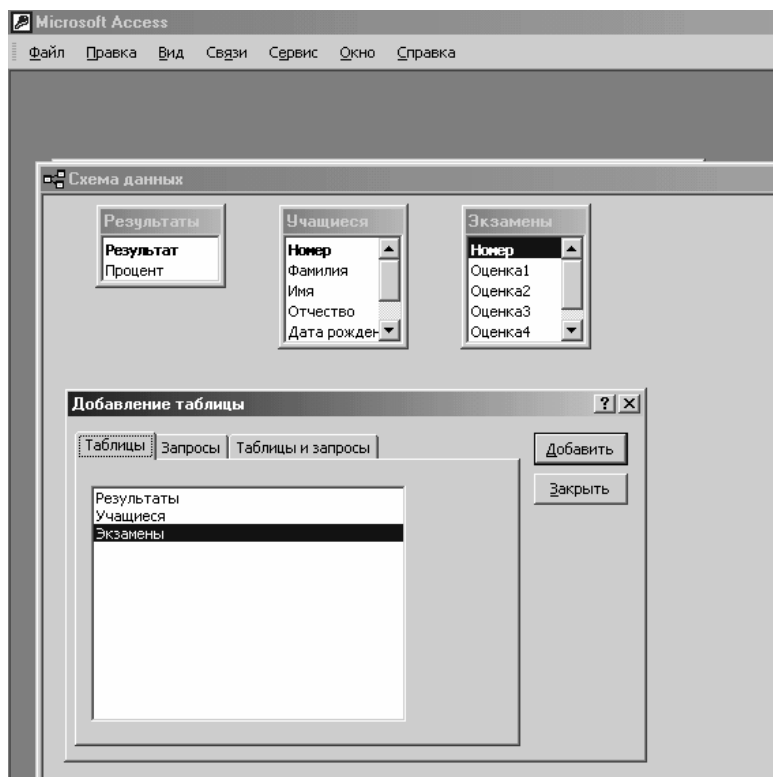


Рис. 12

11. Для установления связей выполнить команду **Сервис, Схема данных**.

12. Установить связи между таблицами *Учащиеся* и *Экзамены*. Для этого протаскать указатель мыши от поля **Номер** таблицы *Учащиеся* к полю **Номер** таблицы *Экзамены* при нажатой кнопке мыши.

13. В появившемся диалоговом окне **Изменение связей** активизировать значки **обеспечение целостности данных**, **каскадное обновление связанных полей** и **каскадное удаление связанных записей**. (Как вы считаете, почему между таблицами данное отношение? Рис. 13). Щелкнуть на кнопке **Создать**.

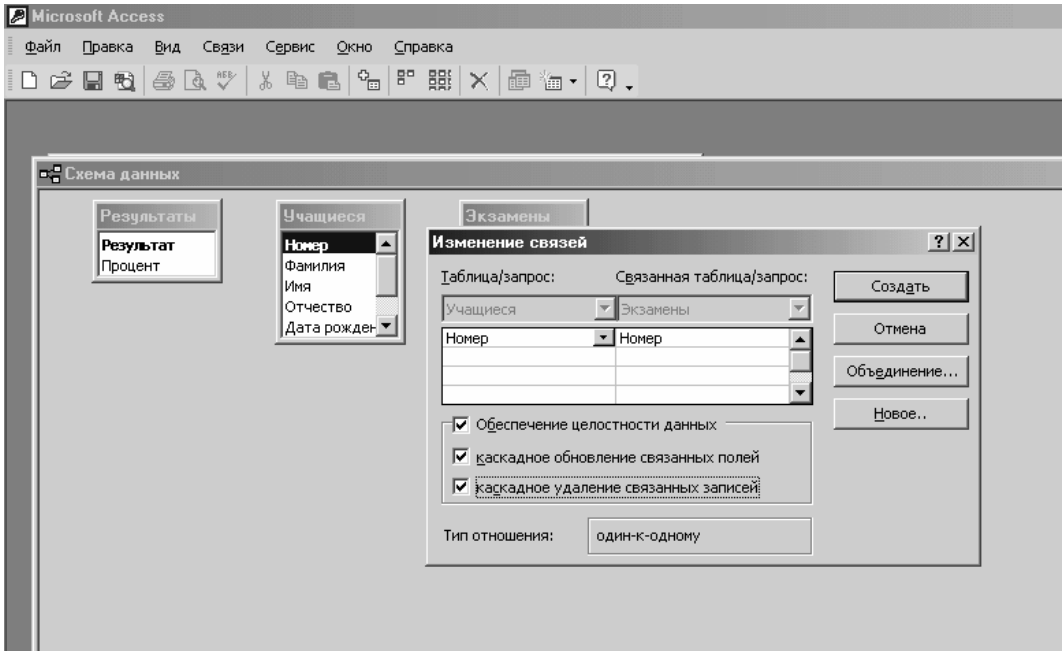


Рис. 13

14. Установить связи между таблицами *Результаты* и *Экзамены*. Для этого протаскать указатель мыши от поля **Результат** таблицы *Результаты* к полю **Результат** таблицы *Экзамены* (рис. 14).

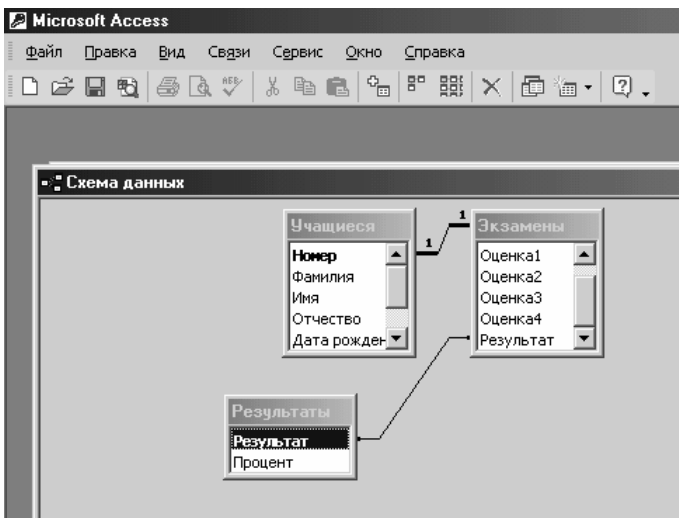


Рис. 14

15. Закрывать окно **Схема данных**, при выходе сохранить связи.

16. Если связи установлены верно, то, открыв таблицу *Учащиеся*, можно посмотреть оценки каждого (рис. 15).

Номер	Фамилия	Имя	Отчество	Дата рождения	Класс	Домашний адрес
1	Загородников	Павел	Николаевич	11.01.1999	7	г. Москва
	Оценка1	Оценка2	Оценка3	Оценка4	Результат	
	5	5	5	5	5	Отл
*	0	0	0	0	0	
2	Иванов	Сергей	Викторович	29.05.1999	8	г. Москва
	Оценка1	Оценка2	Оценка3	Оценка4	Результат	
	4	4	4	4	4	Хор
*	0	0	0	0	0	
3	Ланской	Николай	Георгиевич	07.12.1997	9	г. Москва
4	Мартышкин	Михаил	Викторович	19.06.1999	7	г. Москва
5	Овчиникова	Светлана	Васильевна	17.05.1999	7	г. Москва
6	Петрова	Екатерина	Павловна	14.02.1998	8	г. Москва
7	Рыжаков	Семен	Петрович	28.10.1999	7	г. Москва
8	Ужиков	Артем	Николаевич	14.12.1999	7	г. Москва
*						

Рис. 15

А при открытии, например, таблицы *Результаты* станет доступна информация о том, сколько учащихся получили тот или иной набор оценок, и фамилии этих учеников (рис. 16).

Результат	Процент				
Неуд	0				
Оценка1	Оценка2	Оценка3	Оценка4		
2	2	2	3		
Фамилия	Имя	Отчество	Дата рождения	Класс	Домашний адрес
Овчиникова	Светлана	Васильевна	17.05.1999	7	г. Москва
*	0	0	0	0	
Отл	100				
Удвл	50				
Хор	75				
Номер	Оценка1	Оценка2	Оценка3	Оценка4	
2	4	4	4	4	
Фамилия	Имя	Отчество	Дата рождения	Класс	Домашний адрес
Иванов	Сергей	Викторович	29.05.1999	8	г. Москва
*					
3	4	5	5	4	
Фамилия	Имя	Отчество	Дата рождения	Класс	Домашний адрес
Ланской	Николай	Георгиевич	07.12.1997	9	г. Москва
*					
6	5	4	4	4	
7	4	4	4	4	
8	4	5	5	5	
*	0	0	0	0	

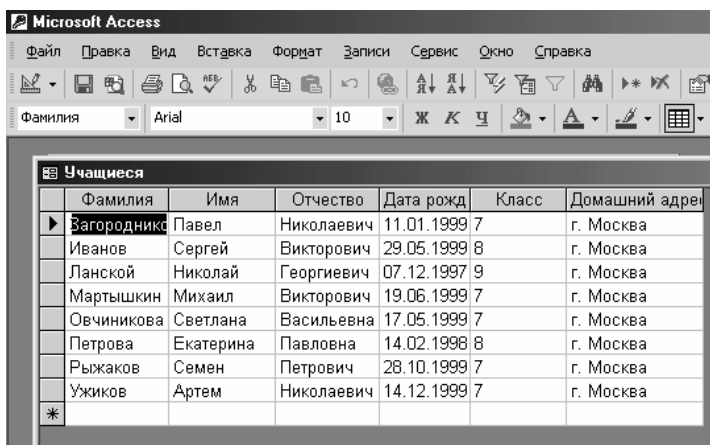
Рис. 16

После выполнения задания 6 можно выполнить задание по подготовке формы для многотабличной БД, сакцентировав внимание учащихся на том, что при создании запросов для БД, содержащей несколько таблиц, следует указывать, к какой таблице относится то или иное поле.

7. Практическое задание: «Разработка многотабличной пользовательской формы ввода данных»

1. Создать форму на основе таблицы *Экзамены* с использованием **Мастера**, включив в форму все поля таблицы. При выборе внешнего вида формы использовать расположение **В один столбец**. Дать форме имя *Экзамены*.

2. Создать форму на основе таблицы *Учащиеся* с использованием **Мастера**, включив в нее все поля, кроме поля «Номер». При выборе внешнего вида формы использовать расположение **Табличный вид** (рис. 17).

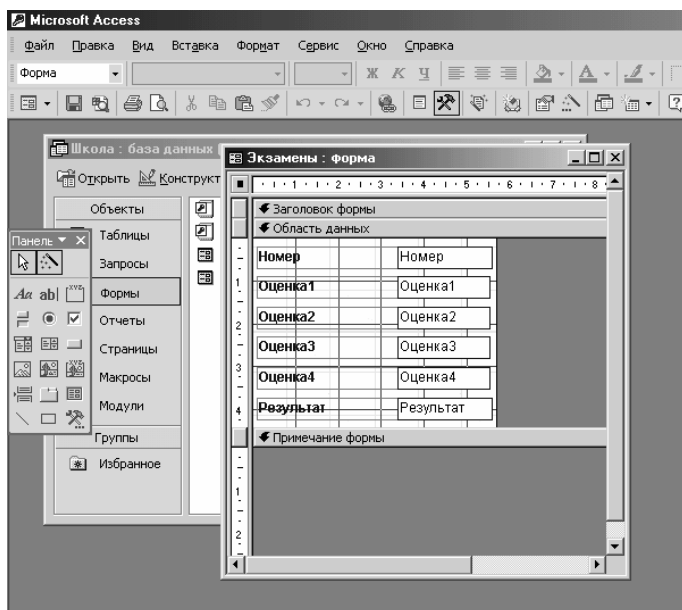


Фамилия	Имя	Отчество	Дата рожд	Класс	Домашний адрес
Загородник	Павел	Николаевич	11.01.1999	7	г. Москва
Иванов	Сергей	Викторович	29.05.1999	8	г. Москва
Ланской	Николай	Георгиевич	07.12.1997	9	г. Москва
Мартышкин	Михаил	Викторович	19.06.1999	7	г. Москва
Овчиникова	Светлана	Васильевна	17.05.1999	7	г. Москва
Петрова	Екатерина	Павловна	14.02.1998	8	г. Москва
Рыжаков	Семен	Петрович	28.10.1999	7	г. Москва
Ужиков	Артем	Николаевич	14.12.1999	7	г. Москва

Рис. 17

Дать форме имя *Школьник*.

3. Закрыть форму *Школьник*. Форму *Экзамены* открыть в режиме конструктора (рис. 18).



Школа : база данных

Экзамены : Форма

Объекты

- Таблицы
- Запросы
- Формы
- Отчеты
- Страницы
- Макросы
- Модули
- Группы
- Избранное

Заголовок формы

Область данных

Номер	Номер
Оценка1	Оценка1
Оценка2	Оценка2
Оценка3	Оценка3
Оценка4	Оценка4
Результат	Результат

Примечание формы

Рис. 18

Для переключения между режимом просмотра формы и режимом конструктора формы можно использовать меню Вид или пиктограмму Вид на панели инструментов.

4. Для оформления атрибутов текста подчиненной таблицы или любого другого объекта формы (например, для изменения цвета и начертания шрифта и т. д.) необходимо в режиме конструктора выделить этот объект и воспользоваться одним из вариантов: либо выбрать команду меню Вид, Свойства и задать параметры атрибутов текста в окне свойств (рис. 19), либо выполнить двойной щелчок на этом объекте для открытия окна свойств.

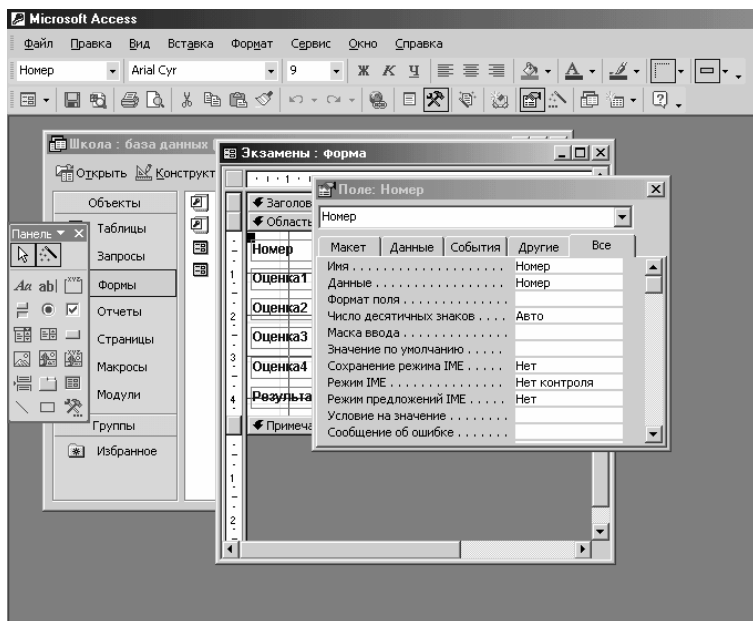


Рис. 19

После этого изменить форму по своему усмотрению.

5. Перенести из окна База данных пиктограмму формы Школьник в нижнюю часть поля формы Экзамены (рис. 20) и перейти в режим формы.

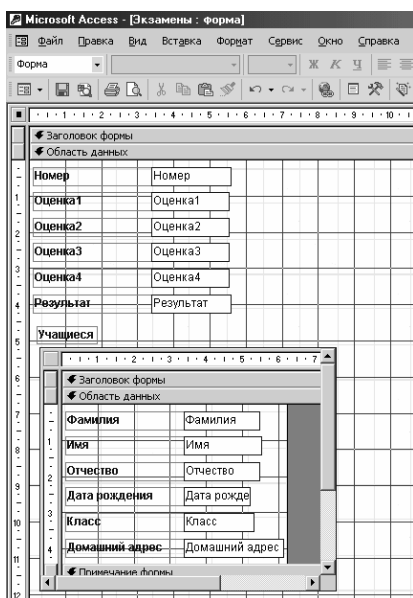


Рис. 20

6. Просмотреть полученную составную форму (рис. 21). Пролистать записи до конца.

Рис. 21

7. Добавить 2—3 записи, используя полученную составную форму: сначала заполнять поля из таблицы *Учащиеся*, затем поля из таблицы *Экзамены*.

8. Закрывать форму.

После выполнения задания 7 необходимо перейти к созданию запросов.

8. Практическое задание: «Формирование запросов для многотабличной базы данных»

Построить запрос, позволяющий выводить фамилию, имя, отчество и класс для всех хорошистов и отличников. Информация для получения таких данных находится в трех таблицах: *Учащиеся*, *Экзамены*, *Результаты*.

В данном случае создается новая таблица, содержащая сведения из разных взаимосвязанных таблиц.

1. В окне **База данных** создать новый запрос на основе связанных таблиц. Для этого активизировать вкладку **Запрос** и щелкнуть на кнопке **Создать**.

2. В появившемся окне **Новый запрос** выбрать **Простой запрос** (с использованием **Мастера запросов**) и подтвердить выбор.

3. В окне **Создание простых запросов** выбрать из таблицы *Учащиеся* поля «Фамилия», «Имя», «Отчество», «Класс»; из таблицы *Результаты* — поле «Процент» (рис. 22).

Рис. 22

4. Закончить работу с **Мастером запросов** самостоятельно.

5. В полученной таблице в строке **Условие отбора** установить по полю «Процент» выражение >50 , т. е. вывод тех школьников, у которых экзамены сданы на «4» и «5» (рис. 23).

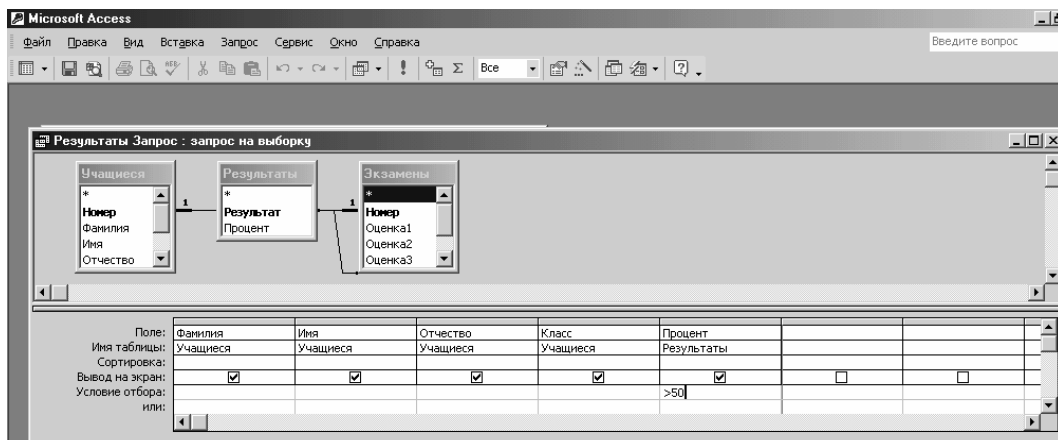


Рис. 23

6. Результат запроса должен быть таков (рис. 24):

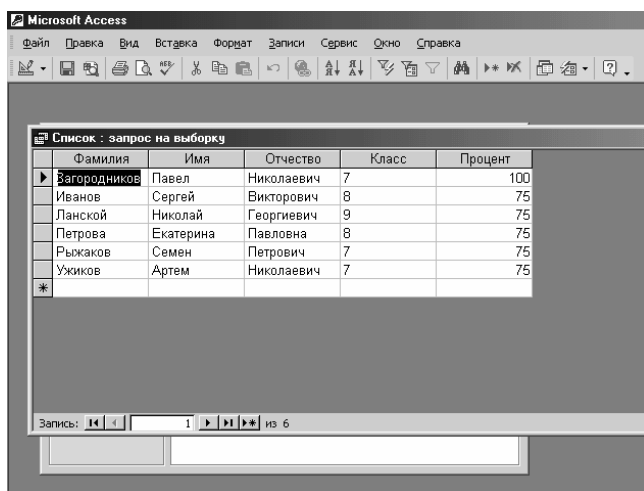


Рис. 24

7. Дать имя запросу *Список* и закрыть его.

Далее учащимся можно предложить следующее задание:

1. Подготовить список школьников, сдавших экзамены на «5».
2. Создать запрос, выводящий список школьников, имеющих хотя бы одну «3».
3. Создать запрос, выводящий список школьников, фамилия которых начинается на букву А.

Практические задания 9 и 10 нацелены не только на развитие практических навыков и умений, но и на воспитание эстетической культуры, а также понимания необходимости создания и улучшения интерфейса любого программного продукта как одного из средств повышения его конкурентоспособности.

9. Практическое задание: «Создание элемента управления»

Элемент управления — это графический объект в форме или отчете для представления данных или для выполнения определенных действий.

Создать элемент управления для ввода пола школьников в форме *Учащиеся*.

1. Создать форму *Учащиеся* (поля выбрать по усмотрению).
2. В таблице *Учащиеся* создать новое поле под именем «Пол» в режиме конструктора. В **Описании** данного поля ввести следующую запись: 1 — мужской, 2 — женский.
3. В окне **База данных** перейти в режим формы, открыть в режиме конструктора форму *Учащиеся*.
4. Увеличить высоту рабочего поля формы.
5. В режиме конструктора формы в дополнительной панели инструментов **Панель элементов** отключить кнопку **Мастер элементов** (должна быть не выделена). Это приведет к отключению мастеров.
6. Щелкнуть на кнопке **Класс** на панели элементов.
7. Открыть **Список полей**, используя соответствующую кнопку на верхней панели инструментов, а затем перетащить поле «Пол» из списка полей в рабочее поле формы при нажатой кнопке мыши. В рабочем поле формы должна появиться рамка с заголовком «Пол» в верхней ее части.
8. Щелкнуть на панели элементов кнопку **Переключатель**, а затем выбрать место, в которое помещается левый верхний угол флажка или выключателя. Подтвердить установку элемента щелчком мыши.
9. Установить курсор на появившийся текст (**Переключатель**) и ввести подпись «Мужской».
10. Аналогичным образом добавить элемент для женского пола.
11. Выровнять значки и отредактировать внешний вид созданной рамки.
12. Перейти в режим просмотра формы и, используя элемент управления, установить пол в записях.
13. Перейти в режим таблицы и просмотреть поле «Пол». В данном поле должны появиться цифры 1 или 2. При установлении курсора в данное поле в нижней области экрана в строке подсказки появится текст описания: 1 — мужской, 2 — женский.
14. Самостоятельно создать элемент управления — поле со списком для ввода оценок, используя **Мастер элементов**.

10. Практическое задание: «Вставка графических объектов в базу данных»

Графический объект можно вставить (или осуществить с ним связь) в форму или отчет как свободный рисунок. Также графический объект может быть вставлен в форму как объект OLE.

Вставить картинки людей из коллекции Clipart в таблицу *Школьник*. Если есть какие-либо фотоизображения людей, можно вставлять их, открыв в соответствующем приложении.

1. Добавить в таблицу *Учащиеся* поле «Фото», тип — **поле объекта OLE**.
2. Запустить графический редактор, например Paint.
3. Открыть любую картинку, выделить весь рисунок и занести его в буфер обмена.
4. Перейти в Access и установить курсор на первую запись поля «Фото».
5. Выполнить команду **Правка, Специальная вставка**.
6. Выбрать тип **Picture**. В поле «Фотография» появится запись (**Рисунок или Картинка**).
7. Аналогичным образом поместить еще 2—3 рисунка.
8. Закрыть таблицу.
9. Открыть форму *Учащиеся* **Конструктора**.
10. Из пиктограммы **Список полей** добавить поле «Фото».
11. Отредактировать его местоположение.
12. Перейти в режим формы и просмотреть записи. В них должны присутствовать изображения.
13. В окне свойств поля «Фото» в строке **Установка размеров** выбрать **Вписать в рамку** или **По размеру рамки**.

Окончание следует

Н. Н. Моисеева,

*почетный работник общего образования РФ, учитель информатики
Центра образования № 1432, Москва*

ЭЛЕКТИВНЫЙ КУРС «ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ФОРМАТИРОВАНИЯ В ДОКУМЕНТАХ HTML»*

Размещение свободно распространяемых программ на Java на странице HTML

Урок 18. Размещение программы на Java на странице. Вводные понятия

В Интернете выложено большое количество свободно распространяемых программ на Java, которые можно бесплатно включить в html-страницу. Примеры таких программ — часы, календари и т. д. Их включение в страницу существенно улучшает ее дизайн.

Условием включения программы на Java является загрузка на компьютер Java-машины. Эту загрузку можно выполнить бесплатно (например, со страницы <http://www.java.com>).

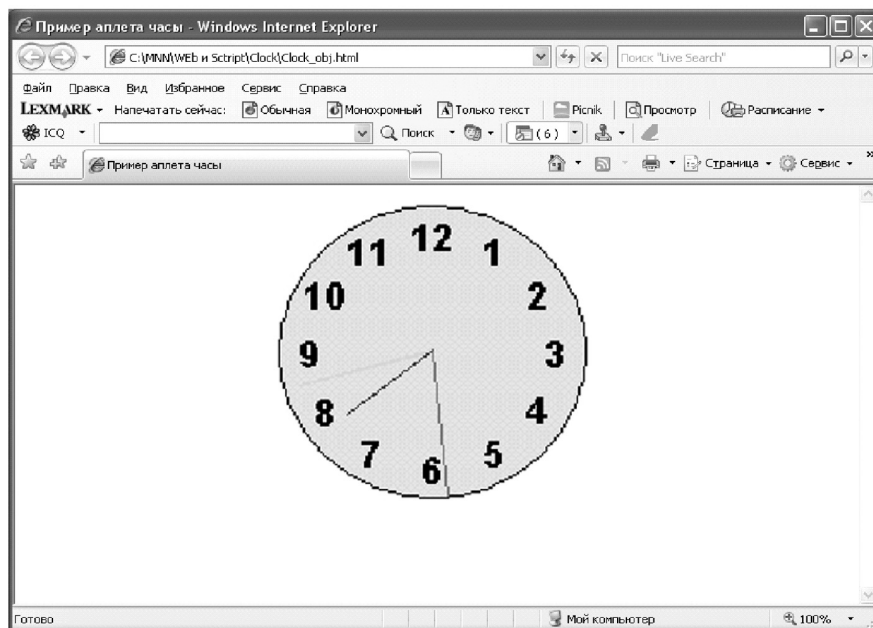
Для включения программы на Java используется тег `object`.

Следует установить следующие атрибуты тега:

Атрибут	Функция
<code>code</code>	Имя программы
<code>codebase</code>	Расположение программы (если атрибут пропущен, то программа располагается в текущем каталоге)
<code>codetype</code>	Тип объекта (для программы на Java application/java-vm)

Значения для подтега `param` зависят от конкретной программы.

Пример использования свободно распространяемой программы вывода времени с помощью программы `clock` (часы) на Java:



* Продолжение. Начало см.: Информатика и образование. 2010. № 1, 2.

Урок 19. Практическое занятие по использованию свободно распространяемых программ на Java на странице HTML

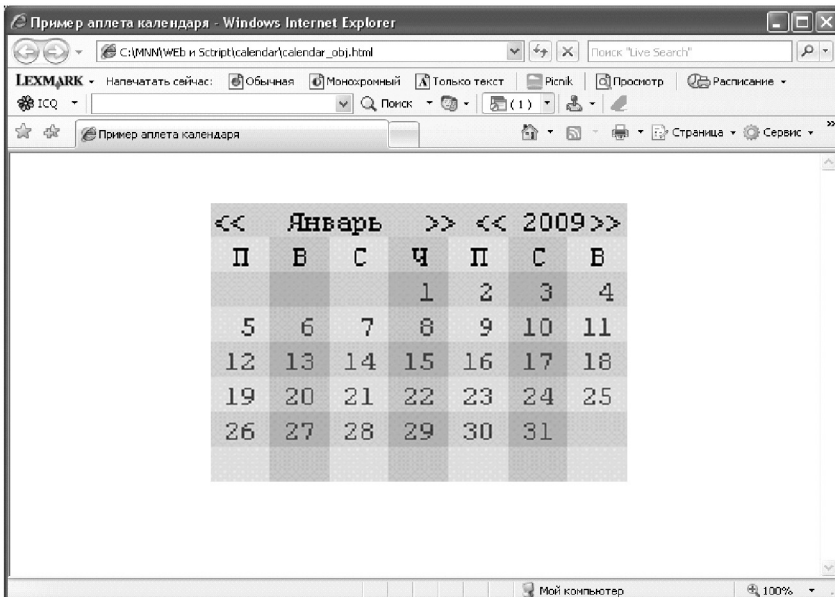
Часто полезно воспользоваться уже готовыми программами для размещения их на своей странице.

Например, свободно распространяемая программа календаря, скопированная с <http://javaboutique.internet.com>, дополнит наш сайт полезной информацией.

Установим размер календаря 182×128. С помощью параметров `monthnames`, `daysofweek` установим соответственно название месяцев и дней недели. Подберем подходящие цвета атрибутом `color`.

```
<html>
<head>
<title> Пример апплета календаря </title>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
</head>
<body bgcolor="#FFFFFF">
<center>
<object codetype="application/java-vm" code="calendar.class"
width=182 height=128>
<param name="datelist"
value="01.01.2009-05.01.2009&07.01.2009">
<param name="monthnames" value="Январь&Февраль&Март&Апрель&Май&
Июнь&Июль&Август&Сентябрь&Октябрь&Ноябрь&Декабрь">
<param name="daysofweek" value="ПВСЧПСВ">
<param name="color0" value="000000">
<param name="color1" value="DAD9AF">
<param name="color2" value="C6C5A2">
<param name="color3" value="CECDA6">
<param name="color4" value="B8B795">
<param name="color5" value="5F5F5F">
<param name="color6" value="3F3FCF">
</object>
</center>
</body>
</html>
```

Осталось только определиться с местом размещения календаря на странице, и нужная информация будет всегда доступна для просмотра.



Урок 20. Самостоятельная работа по использованию свободно распространяемых программ на Java на странице HTML

На сайте свободного программного обеспечения <http://javaboutique.internet.com> выберите программу на Java для календаря, часов или калькулятора. Скопируйте архив программы в формате zip на свой компьютер и разархивируйте его. Файл с расширением class и является программой на Java. Также в архиве есть образец файла HTML, который содержит пример задания параметров.

Вставьте программу на Java как объект на свою страницу HTML, как это описано выше. В качестве параметров программы скопируйте параметры из файла с расширением HTML.

Размеченные изображения

Урок 21. Размеченные изображения. Вводные понятия

Размеченные изображения — это одно из самых удобных и часто используемых при создании сайтов средств разбиения графического изображения для более детального изучения его фрагментов.

Любое изображение на странице может быть разбито на области. К каждой области можно привязать переход на собственную гиперссылку. Таким образом, создается очень удобный графический интерфейс для карт, путеводителей, справочников и т. д.

Для включения разметки следует добавить в тег `img` атрибут `usemap`, который указывает ссылку на описание областей.

Пример:

```

```

Описание областей задается тегом `map` и может размещаться в любой части документа. Для связи описания и изображения служит атрибут тега `name`, который должен содержать имя ссылки.

Между этими командами для каждой области изображения вставляется подтег `area` с необходимыми атрибутами.

Атрибут	Функция
<code>coords</code>	Координаты области
<code>href</code>	Ссылка для данной области
<code>nohref</code>	Отсутствие ссылки
<code>shape</code>	Форма области

Области могут перекрываться. В перекрытии будет действовать первая ссылка.

Атрибут `shape` принимает четыре значения: `rect` (прямоугольник), `circ` (круг), `poly` (многоугольник), `default` (прямоугольник). Если атрибут не задан, то будет задана форма прямоугольника.

Атрибут `href` задает гиперссылку, куда будет осуществлен переход при щелчке мышью на области.

Атрибут `nohref` указывает, что переходы делать не надо. Он может быть полезен, когда в области надо вырезать «нечувствительную» дырку.

Атрибут `coords` задает координаты фигур в пикселях. Все координаты записываются через запятую:

- прямоугольник — задаются координаты левого верхнего и правого нижнего углов. Например: `coords="0,0,49,49"`;
- круг — задаются координаты центра и радиус. Например: `coords="50,50,25"`;
- многоугольник — задается последовательностью координат от первой до предпоследней вершины. Считается, что последняя вершина совпадает с первой. Например: `coords="0,0,99,49,49,99"`.

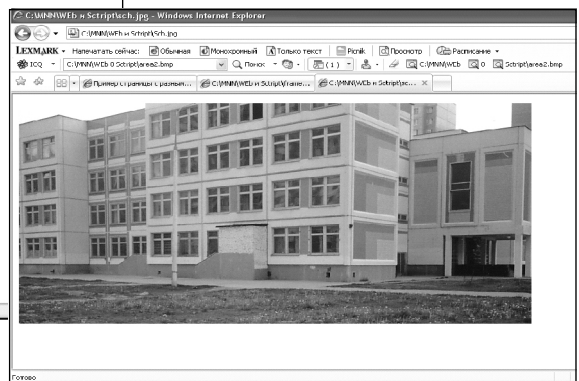
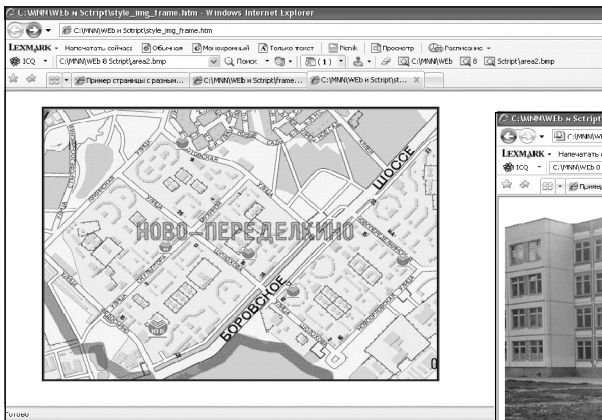
Урок 22. Практическое занятие по созданию размеченной карты

Создадим размеченную карту района. Карта разбита на четыре треугольника, как показано на рисунке:



При щелчке мышью на левом треугольнике должна появиться фотография школы. Остальные области не чувствительны к щелчку мышью.

```
<html>
<head>
  <map name="coord">
    <area coords="0,0,300,200,0,400" shape="poly" href="sch.jpg">
    <area coords="0,0,600,0,300,200" shape="poly" nohref>
    <area coords="600,0,600,400,300,200" shape="poly" nohref>
    <area coords="0,400,300,200,600,400" shape="poly" nohref>
  </map>
</head>
<body>
  <center>
    
  </center>
</body>
</html>
```



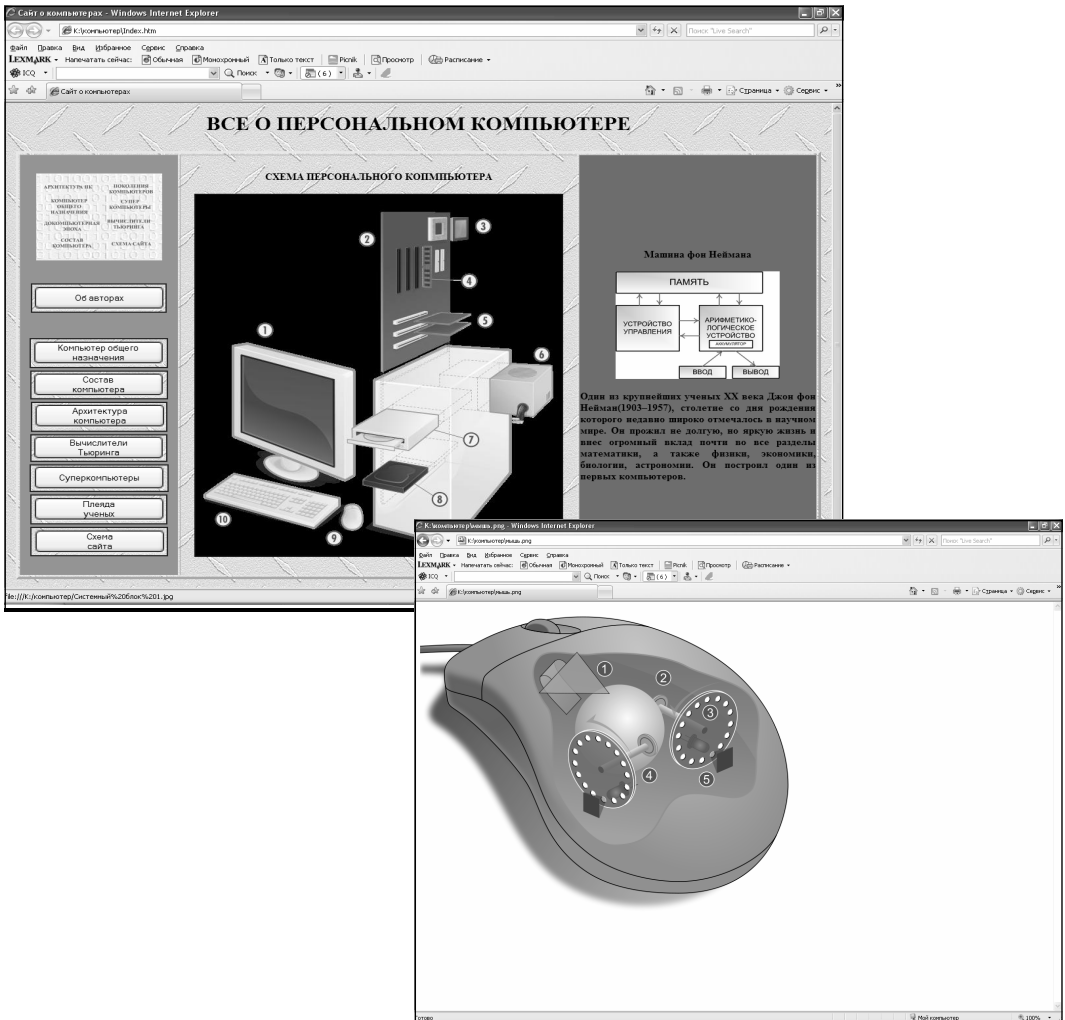
Урок 23. Практическое занятие по созданию размеченной схемы компьютера

Рассмотрим пример использования включения файлов анимации в формате Macromedia Flash и разбиения схемы компьютера для изучения его отдельных частей.

Для этого создадим новую страницу, выберем нужный файл анимации и подберем подходящий рисунок компьютера со схемой. Затем разметим изображение на нужные зоны и установим ссылки на изображения частей компьютера. При наведении указателя мыши на нужную зону открывается изображение выбранной части компьютера. Причем для определения довольно точных координат фрагментов изображения схемы компьютера можно использовать графический редактор Paint.

Определим координаты областей на схеме компьютера для мыши, клавиатуры, жесткого диска, привода CD-ROM, блока питания, системного блока, монитора, материнской платы. Затем запишем эти координаты с помощью параметра `coords` и обязательно подберем крупные изображения соответствующих устройств для установки гиперссылок.

```
<html>
  <head>
    <map name="coord">
      <area coords="223,553,225,531,231,517,251,521,254,545,238,558"
        shape="poly" href="мышь.png">
      <area coords="10,504,56,460,220,501,178,556"
        shape="poly" href="клавиатура.jpg">
      <area coords="253,481,295,442,352,455,317,492"
        shape="poly" href="жесткий диск.gif">
      <area coords="238,389,291,355,368,367,332,426"
        shape="poly" href="cd-rom.jpg">
      <area coords="446,308,481,278,543,284,542,336,512,372,448,361"
        shape="poly" href="блок питания.jpg">
      <area coords="270,306,384,246,480,225,482,460,386,584,270,550"
        shape="poly" href="системный блок.jpg">
      <area coords="53,404,44,246,242,287,228,477"
        shape="poly" href="монитор.jpg">
      <area coords="248,28,387,27,387,241,283,293"
        shape="poly" href="материнская плата.jpg">
    </map>
  </head>
  ...
  <th bgcolor="879898" width="20%" height="700">
    <object height="145" codetype="application/x-shockwave-flash"
      classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000">
      <param name="movie" value="Меню.swf">
      <param name="play" value="true">
      <param name="loop" value="true">
    </object>
    <p><a href="Авторы.htm"><br></a></p>
    <a href="Компьютер общего.htm"><br>
      </a>
    <a href="Состав ПК.htm"></a>
    <a href="Архитектура.htm"></a>
    <a href="Вычислители.htm"></a>
    <a href="Суперкомпьютеры.htm"></a>
    <a href="Плеяда ученых.htm"></a>
    <a href="Схема сайта.htm"><br></a>
  </th>
  <th width="50%" height="700">
    <p>СХЕМА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА</p>
    
  ...
</html>
```



Урок 24. Самостоятельная работа по созданию размеченного изображения

Подберите набор изображений по какой-либо теме (например, «Модели автомобилей») и сохраните этот набор на компьютере в виде файлов формата jpg. Создайте единое изображение из этого набора путем склейки уменьшенных изображений и сохраните его в виде файла формата jpg на компьютере. Сформируйте размеченное изображение на странице на базе единого файла, где отдельными областями будут уменьшенные изображения из набора. При щелчке мышью на отдельной области должно вызываться полное изображение из набора (эффект линзы).

Литература

1. *Ахметов К. С., Федоров А. Г.* Microsoft Internet Explorer 4 для всех. М.: КомпьютерПресс, 1997.
2. *Моисеева Н. Н.* Начала веб-дизайна // Информатика и образование. 2007. № 10—12.
3. *Штайнер Г.* HTML/XML/CSS. М.: Лаборатория Базовых Знаний, 2001.

Окончание следует



ПЕДАГОГИЧЕСКИЕ ИЗМЕРЕНИЯ И ТЕСТЫ

С. В. Юнов,

канд. физ.-мат. наук, доцент кафедры прикладной математики
Кубанского государственного университета, г. Краснодар

ФАСЕТНЫЕ И ПСЕВДОФАСЕТНЫЕ ТЕСТОВЫЕ ЗАДАНИЯ В ИЗМЕРЕНИЯХ КАЧЕСТВА ИНФОРМАЦИОННОЙ ПОДГОТОВКИ

Тестирование у нас в стране приобретает массовый характер. При этом автору еще не доводилось встречать специалистов в области информатики и ИКТ, которые положительно отзывались бы о тестах, составленных не ими. Ситуация чем-то напоминает фрагмент из книги Н. Н. Носова «Приключения Незнайки и его друзей», только наоборот: там всем нравились все портреты, нарисованные Незнайкой, кроме того, где был изображен сам опрашиваемый коротышка.

Конечно, составление тестовых заданий — задача далеко не простая. Ведь, как справедливо утверждает известный специалист в области педагогических измерений В. С. Аванесов, «написание хороших учебных текстов является не только научной деятельностью, но и искусством... Например, можно научиться складывать стихи, но возникнет ли из этого поэзия?» [2].

При тестировании удобно применять так называемые фасеты. «Фасет — это форма записи возможных параллельных вариантов задания, что является обязательным требованием при разработке теста, имеющего на каждое задание набор параллельных заданий. Все элементы одного фасета априорно считаются одинаково трудными по содержанию, что требует последующего эмпирического подтверждения» [2].

Принцип фасетности широко используется при тестировании именно благодаря возможности создавать в одном задании сразу несколько вариантов [1, 2].

Интересно, что сам термин «фасет» применяется в нескольких значениях, из которых, как будет видно ниже, нам ближе всего подойдет «Роговица каждого из отдельных глазков, из которых состоит сложный глаз членистоногих (зоол.)» [3].

Рассмотрим пример задания с одним фасетом (здесь он представлен в фигурных скобках) и двумя вариантами, приведенный в работе В. С. Аванесова [2]:

{Зимний, летний} муссон дует

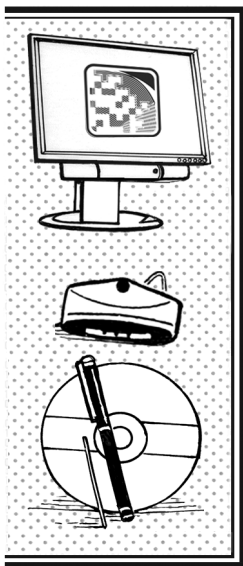
- 1) с суши на море
- 2) с моря на сушу

В этих заданиях один учащийся получает один из двух вариантов. В зависимости от полученного варианта правильный ответ испытуемого может меняться. Таким образом, фасетная запись задания предназначена не для тестируемого школьника (студента), а для тестирующего учителя (программиста).

Еще один важный для дальнейших рассуждений пример из работы В. С. Аванесова:

Глагол {бежать, прыгать, делать, ...}

- 1) переходный
- 2) непереходный



Автор анализирует свойства таких заданий, из которых выделим следующие:

- слова в фигурных скобках можно менять, поэтому такие задания не боятся рассекречивания после 1—2 тестирований, ибо для правильного ответа надо знать общие правила, а следовательно, быть готовым ответить на любой вариант;
- все варианты замены предполагаются равно трудными. Это позволяет проверить знание темы «Переходные и непереходные глаголы» у всех учащихся посредством одного задания с заменяющимися элементами. Точки в фасете означают, что число вариантов такого задания практически не ограничено.

Принципиально важным для нас является вывод исследователя о том, что «если подтверждается факт различающейся трудности отнесения отдельных глаголов в той или иной группе испытуемых, то такие элементы из фасета исключаются» [2].

Приведем пример задания на проверку качества информационной подготовки учащихся, которое обладает указанными выше свойствами [1, с. 40].

Пример 1.

Автомат производит измерения 20 раз в секунду. Результаты каждого измерения занимают 1 байт. Объем памяти, который потребует компьютеру для записи всех измерений за 1 минуту, составляет ...

- 1) 160 битов
- 2) 20 байтов
- 3) 1200 битов
- 4) 1200 байтов

Понятно, что, изменяя в условии числовые значения (конечно, в разумных пределах!), можно составить достаточно много вариантов равноценных заданий.

Однако такие «хорошие» для разработчиков тестов задания являются исключением. В практической педагогической деятельности при измерениях качества информационной подготовки достаточно сложно разработать тестовые задания, которые обладали бы указанными выше свойствами. *Задания, которые, по задумке разработчиков, должны быть фасетными, таковыми часто не являются.*

Приведем несколько примеров, которые это наглядно демонстрируют.

Пример 2.

Как изменить с помощью клавиатуры размер окна стандартной программы Windows {Paint, WordPad, Калькулятор, ...}

- 1) Вызвать системное меню с помощью комбинации клавиш Alt+пробел и выбрать пункт **Размер**
- 2) Вызвать верхнее меню программы с помощью клавиши Alt и выбрать пункт **Размер**

Правильным является первый вариант ответа. Но не для всех элементов, входящих в фасет! Учащиеся, которым «достанется» вариант со словом «Калькулятор», будут находиться в затруднении: эта программа не может произвольно менять размеры своего окна.

Пример 3.

В текстовом редакторе MS Word создана таблица {4×4, 5×5, ...}. Как добавить к ней {1, 2, ...} строчку?

В результате автоматической генерации заданий кто-то из учащихся получит вариант, в котором от него потребуются добавить к таблице последнюю строчку. Но в ряде ранних версий этого текстового редактора добавление последней строки (как и последнего столбца) производится нестандартным способом. Таким образом, здесь нарушается предъявляемое к фасетным заданиям требование, заключающееся в равной трудности вариантов.

Пример 4.

В табличном процессоре MS Excel создана база данных товаров. Отобрать товары, поставленные {весной, летом, ...} 2008 г.

Нарушение требования фасетности задания здесь заключается в том, что отобрать из базы данных товары, поставленные зимой, гораздо сложнее, чем товары, поставленные в другие времена года. Продемонстрируем этот факт.

Приведем фрагмент базы данных:

Наименование	Дата поставки
Трубы	11.07.2008
Елки	25.02.2008
Окна	12.12.2008
Двери	15.04.2008

Для отбора товаров, поставленных *весной*, с помощью автофильтра получим:

При решении задачи с помощью расширенного фильтра диапазон условий будет таким:

Дата поставки	Дата поставки
>=01.03.08	<01.06.08

Понятно, что *для лета и осени* достаточно просто поменять даты в приведенных окнах автофильтра и диапазона условий расширенного фильтра. Для *зимы* же ситуация усложняется.

С помощью автофильтра эту задачу решить нельзя (рассматривается версия MS Excel 2003).

Диапазон условий при использовании расширенного фильтра выглядит так:

Дата поставки	Дата поставки
>=01.01.08	<01.03.08
>=01.12.08	<01.01.09

Понятно, что задания получаются явно неравноценными.

Конечно, многие разработчики тестов осведомлены о том, что «задания имеют шанс стать тестовыми только после их эмпирической проверки» [2], однако именно в процессе измерения качества информационной подготовки делать это особенно трудно из-за постоянных изменений (кстати, не всегда оправданных) как программного, так и аппаратного обеспечения компьютеров. Меняются не только сами возможности средств информатизации, но и технологии их использования. Поэтому к составлению тестовых заданий нужно подходить предельно осторожно, медицинский принцип «не навредить» должен быть здесь главным.

Назовем **псевдофасетными** такие тестовые задания, которые внешне имеют форму фасетных, однако при анализе элементов, составляющих фасет (любой из фасетов задания), выявляется, что учащиеся могут получить задания, существенно отличающиеся по сложности. Такие задания, к сожалению, нередко встречаются на практике и, по аналогии со лжесъедобными грибами, представляют собой особую опасность, в данном случае опасность искажения результатов тестирования.

Однако если разработчики тестов должны избегать псевдофасетных тестов, то преподаватели информатики и ИКТ вполне могут их использовать в своей педагогической деятельности. Так, анализ псевдофасетных тестов, на наш взгляд, можно использовать при проблемном обучении возможностям современного программного обеспечения. Интересными являются также вопросы к учащимся с просьбой самостоятельно подготовить псевдофасетные тесты и проблемы выявления «критических элементов» фасетов.

Литературные и интернет-источники

1. Самылкина Н. Н. Современные средства оценивания результатов обучения. М.: БИНОМ. Лаборатория знаний, 2007.

2. <http://testolog.narod.ru/index.html> — Сайт научно-методической поддержки слушателей курсов В. С. Аванесова по проблеме «Педагогические измерения».

3. <http://www.vseslova.ru/index.php?dictionary=ushakov&word=faset> — Коллекция словарей на ВсеСлова.ру.

НОВОСТИ ЦИФРОВОГО МИРА

Юбилей: Adobe Photoshop исполнилось 20 лет

10 февраля 2010 г. всемирно известной программе Adobe Photoshop исполнилось 20 лет. Она прошла путь от мелкой утилиты, поставляемой вместе со сканерами, до мощнейшего графического пакета.

Все началось в 1987 г. с создания Томом Ноллом подпрограммы для конвертации монохромных изображений на компьютере в полутоновые. Успех утилиты вдохновил автора, и он принялся за разработку других средств, для применения различных эффектов к цифровым изображениям. В 1988 г. была создана программа Image Pro, которая через полгода стала поставляться вместе со сканерами фирмы Barneyscan.

Год спустя программа, обросшая функциями редактирования изображений, была лицензирована Adobe, и в 1990 г. поступила в продажу уже под своим современным названием. Это была Adobe Photoshop 1.0. При этом, по словам Томаса Нолла, предполагалось, что ежемесячно будет продаваться 500 копий программы. Сейчас, по данным Adobe, число пользователей Photoshop во всем мире превышает 10 миллионов. Надо полагать, речь идет только о легальных пользователях, поскольку в противном случае пришлось бы говорить о цифре другого порядка.

Интересно, что компания Adobe особенно выделяет в истории разработки знаменитого продукта две уже весьма старые версии. Так, по мнению специалистов компании, революцией оказалась реализация механизма слоев в Photoshop 3.0. Кроме того, революционную роль сыграла и седьмая версия программы с массой новых функций обработки изображений, в числе которых выделяется в первую очередь Healing Brush («Восстанавливающая кисть»). В 2003 г. была запущена линейка CS. Это уже всем известный и знакомый продукт. Сейчас уже появилась линия CS4, где реализована возможность работы с трехмерными объектами.

Специальное мероприятие, организованное Национальной ассоциацией профессиональных пользователей Photoshop, прошло в Сан-Франциско. Оно собрало более тысячи гостей, в том числе топ-менеджмент Adobe. Была организована трансляция вечера в Сети. Празднование юбилея Photoshop прошло в Германии, Франции, Индии и в других странах мира.

(По материалам международного компьютерного еженедельника Computerworld Россия)

В. И. Сердюков,

*доктор техн. наук, профессор, гл. науч. сотрудник
Института информатизации образования РАО, Москва*

О КОЛИЧЕСТВЕННОМ ОЦЕНИВАНИИ ДОСТОВЕРНОСТИ РЕЗУЛЬТАТОВ АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ ЗНАНИЙ

В практике образовательных учреждений всё чаще используются автоматизированные системы контроля знаний (АСКЗ). Возможности таких систем действительно высоки, а в перспективе будут только возрастать. Однако АСКЗ имеют свои особенности, которые нужно правильно учитывать при их разработке и эксплуатации.

Одна из особенностей заключается в том, что точечные оценки уровня знаний обучающихся, получаемые с использованием АСКЗ, могут оказаться смещенными в сторону их завышения относительно истинных значений.

Для оценки смещения была разработана вероятностная модель АСКЗ, входными данными которой являются количество тестовых заданий n ; количество предлагаемых и правильных вариантов ответов по каждому тестовому заданию, обозначаемых соответственно g_i и v_i , где $i = \overline{1, n}$.

Каждое тестовое задание будет представлять собой испытание Бернулли, т. е. такое испытание, в результате которого возможны только два исхода:

- успех (тестовое задание решено правильно) с вероятностью

$$p_i = \frac{v_i}{g_i};$$

- неуспех (тестовое задание решено неправильно или нерешено) с вероятностью

$$q_i = 1 - p_i, \quad i = \overline{1, n}.$$

Тогда вероятность случайного события, заключающегося в том, что обучаемый правильно решит m тестовых заданий из n ($m = \overline{0, n}$), будет равна:

$$P_m = \sum_{\{S_i = \{i_1, i_2, \dots, i_m\} \subset M; G_j = \{j_1, j_2, \dots, j_{n-m}\} \subset M \setminus S_i\}} p_{i_1} \cdot p_{i_2} \cdot \dots \cdot p_{i_m} \cdot q_{j_1} \cdot q_{j_2} \cdot \dots \cdot q_{j_{n-m}}.$$

В этой формуле:

S_i — множество, составленное из m порядковых номеров тестовых заданий, решенных обучаемым правильно;

G_j — множество, составленное из $(n - m)$ порядковых номеров тестовых заданий, решенных обучаемым неправильно;

$$M = S_i \cup G_j.$$

Данная формула задает распределение вероятностей случайной величины m — количества тестовых заданий, правильно решенных обучаемым в ходе автоматизированного контроля тестовых заданий.

Максимальный уровень смещения возможен при $g_i = 2v_i$ и $v_i = 1$. Ниже представлены результаты оценки, полученные при $n = 10$.

Задача в прямой постановке.

Допустим, что известен истинный уровень знаний обучаемого, т. е. известно, сколько тестовых заданий он изначально знает, как выполнить правильно, и обязательно сделает это в ходе тестирования. Обозначим количество таких тестовых заданий через k . Требуется определить, сколько тестовых заданий будет выполнено им правильно (m).

Дело в том, что обучаемый, не зная правильного ответа на то или иное тестовое задание, имеет возможность выбора ответа на него. При этом он, безусловно, знает, что один из вариантов ответа, предусмотренных тестовым заданием, является правильным. Этот правильный ответ он может угадать случайным образом. И если это

произойдет, то окажется, что данное тестовое задание выполнено обучаемым правильно. Таким образом, $m \geq k$. Вероятности возможных исходов тестирования сведены в табл. 1.

Таблица 1

k	m										
	0	1	2	3	4	5	6	7	8	9	10
0	0,001	0,010	0,044	0,117	0,205	0,246	0,205	0,117	0,044	0,010	0,001
1	—	0,002	0,018	0,070	0,164	0,246	0,246	0,164	0,070	0,018	0,002
2	—	—	0,004	0,031	0,109	0,219	0,274	0,219	0,109	0,031	0,004
3	—	—	—	0,008	0,055	0,164	0,273	0,273	0,164	0,055	0,008
4	—	—	—	—	0,016	0,094	0,234	0,312	0,234	0,094	0,016
5	—	—	—	—	—	0,031	0,156	0,313	0,313	0,136	0,031
6	—	—	—	—	—	—	0,062	0,250	0,376	0,250	0,062
7	—	—	—	—	—	—	—	0,125	0,375	0,375	0,125
8	—	—	—	—	—	—	—	—	0,250	0,500	0,250
9	—	—	—	—	—	—	—	—	—	0,500	0,500
10	—	—	—	—	—	—	—	—	—	—	1,000

Таким образом, при любом значении $k < 10$, благодаря «его величеству случаю», тестируемый может правильно выполнить все 10 тестовых заданий, причем вероятность такого исхода достаточно высока.

Элемент «игры «Угадай-ка»» приводит к тому, что математическое ожидание суммарного количества правильно выполненных тестовых заданий смещено относительно k в положительную сторону на величину $Z = Z(k)$, значения которой сведены в табл. 2.

Таблица 2

k	0	1	2	3	4	5	6	7	8	9
$Z(k)$	+5,0	+4,5	+4,0	+3,5	+3,0	+2,5	+2,0	+1,5	+1,0	+0,5

Таким образом, при значениях k , меньших 10, будет иметь место положительное смещение значения математического ожидания количества правильно выполненных обучаемым тестовых заданий относительно k . Это смещение отсутствует только при значении k , равном 10, да и то потому, что смещать математическое ожидание в положительную сторону некуда: все тестовые задания выполнены. Средняя величина смещения по множеству k будет равна +2,5. Это среднее смещение результатов тестирования за счет «игры «Угадай-ка»», оцененное с позиции обучаемого. Максимум смещения достигается при $k = 0$, т. е. на наибольшую выгоду от «игры «Угадай-ка»» может рассчитывать в среднем тот обучаемый, который обладает наименьшим уровнем знаний.

Задача в обратной постановке.

Предположим, что преподавателю стали известны результаты тестирования знаний обучаемого. В ходе тестирования правильно решено m тестовых заданий. Каков реальный (истинный) уровень знаний этого обучаемого?

Другими словами, для преподавателя не представляет интереса то, каких успехов в процессе тестирования достиг обучаемый в «игре «Угадай-ка»». Его, безусловно, интересует тот результат, который показал бы обучаемый, не прибегая к этой игре. Однако отделить один результат от другого в рамках данной задачи можно, если использовать методы теории вероятностей и математической статистики. Правда, интерпретировать статистические выводы можно только к результатам тестирования всех обучаемых, не выделяя никого конкретно.

Итак, какова вероятность того, что обучаемый, правильно выполнивший в ходе тестирования m тестовых заданий, обладает реальными (истинными) знаниями,

позволяющими ему правильно решить ровно k тестовых задания, где $k = 0, 1, 2, \dots, (m - 1), m$?

Результаты расчета указанных условных вероятностей, определенных при помощи байесовского подхода, сведены в табл. 3.

Таблица 3

m	k										
	0	1	2	3	4	5	6	7	8	9	10
1	0,833	0,167	—	—	—	—	—	—	—	—	—
2	0,667	0,273	0,060	—	—	—	—	—	—	—	—
3	0,518	0,310	0,137	0,035	—	—	—	—	—	—	—
4	0,373	0,299	0,199	0,100	0,029	—	—	—	—	—	—
5	0,246	0,246	0,219	0,164	0,094	0,031	—	—	—	—	—
6	0,141	0,170	0,189	0,188	0,161	0,108	0,043	—	—	—	—
7	0,066	0,092	0,124	0,154	0,176	0,177	0,141	0,070	—	—	—
8	0,023	0,036	0,056	0,085	0,121	0,162	0,194	0,194	0,129	—	—
9	0,005	0,009	0,016	0,028	0,048	0,069	0,127	0,190	0,254	0,254	—
10	0,0005	0,0010	0,0020	0,0040	0,0080	0,0155	0,0311	0,0625	0,1251	0,2501	0,5002

Как следует из табл. 3, только тогда, когда обучаемый правильно решил все 10 тестовых заданий, вероятность того, что данный результат соответствует истинному уровню знаний испытуемого, будет больше 0,5, да и то незначительно. В остальных случаях эта вероятность ниже 0,5, и, чем меньше значение m , тем она ниже.

Элемент «игры «Угадай-ка»» приводит к тому, что при любом значении m математическое ожидание суммарного количества правильно выполненных тестовых заданий смещено относительно m в отрицательную сторону на величину $Z = Z(m)$, значения которой сведены в табл. 4.

Таблица 4

m	1	2	3	4	5	6	7	8	9	10
$Z(m)$	-0,833	-1,607	-2,326	-2,887	-3,293	-3,446	-3,273	-2,751	-1,928	-0,996

Таким образом, при значениях m , отличных от нуля, будет иметь место отрицательное смещение величины математического ожидания количества правильно выполненных обучаемым тестовых заданий относительно результата тестирования m . Это смещение отсутствует только при значении m , равном 0, да и то потому, что смещать математическое ожидание в отрицательную сторону некуда: ни одного тестового задания обучаемый правильно не выполнил. Средняя величина смещения по множеству значений m будет равна -2,12. Это среднее смещение результатов тестирования за счет «игры «Угадай-ка»», оцененное с позиции преподавателя.

По абсолютной величине смещения математические ожидания, оцененные с позиции обучаемого и с позиции преподавателя, различаются между собой незначительно. Но здесь более важным является качественное различие этих оценок:

- Максимум смещения, если оценивать его с позиции преподавателя, достигается при $m = 6$, т. е. на наибольшую выгоду от «игры «Угадай-ка»» может в среднем рассчитывать тот обучаемый, который обладает средним уровнем знаний.
- Наименьшую выгоду от «игры «Угадай-ка»», если оценивать ее результаты с позиции обучаемого, следует ожидать (в среднем) тем, у кого уровень знаний близок к «нулевому».

Помимо точечной оценки интерес представляет и интервальная оценка случайной величины. Здесь также возможны две постановки задачи: прямая и обратная.

Рассмотрим задачу интервальной оценки достоверности тестирования знаний обучаемого в прямой постановке.

Пусть задано значение доверительной вероятности α . Требуется определить доверительный интервал, который накроет истинное значение k с вероятностью α .

Особенность решения такой задачи состоит в следующем.

Во-первых, длина доверительного интервала будет зависеть от величины m .

Во-вторых, координата правой границы доверительного интервала будет равна m .

Все данные, необходимые для решения задачи в такой постановке, представлены в табл. 3. Необходимо лишь задать конкретное значение α . Обычно значение α выбирают из множества чисел отрезка $[0,8; 0,999]$. В табл. 5 представлены результаты решения данной задачи при $\alpha = 0,8$.

Таблица 5

m	Координаты границ доверительного интервала		Длина доверительного интервала
	левой	правой	
1	0	1	1
2	0	2	2
3	0	3	3
4	0	4	4
5	0	5	5
6	1	6	5
7	2	7	5
8	4	8	4
9	6	9	3
10	8	10	2

Как следует из данных табл. 5, наибольшая длина у доверительного интервала будет тогда, когда обучаемый в ходе тестирования показал средний результат, выполнив правильно 5, 6 или 7 тестовых заданий.

Теперь рассмотрим задачу интервальной оценки в обратной постановке.

Пусть задан размер доверительного интервала, накрывающего истинное значение k . Требуется определить доверительную вероятность, с которой этот доверительный интервал накроет истинное значение k .

Конечно, результат решения задачи будет зависеть от значений параметров: k , m и границ заданного доверительного интервала. Но главным является то, что модель позволяет определить искомую доверительную вероятность при любых значениях указанных параметров и границ доверительного интервала. В качестве примера в табл. 6 представлены значения доверительных вероятностей, с которыми доверительный интервал $[m - 1; m]$ накроет истинное значение случайной величины k .

Таблица 6

Количество правильно выполненных тестовых заданий m	Доверительный интервал $[m - 1; m]$	Доверительная вероятность
2	[1; 2]	0,333
3	[2; 3]	0,172
4	[3; 4]	0,129
5	[4; 5]	0,125
6	[5; 6]	0,152
7	[6; 7]	0,211
8	[7; 8]	0,323
9	[8; 9]	0,508

Как следует из данных табл. 6, наименьшая доверительная вероятность будет приходиться на доверительный интервал, соответствующий значению $m = 5$. Наибольшая — при $m = 9$.

Таким образом, одна из особенностей оценки знаний обучаемого тестированием заключается в том, что среднее значение такой оценки будет, как правило, смещено относительно истинного уровня знаний обучаемого.

В данной работе рассматривался случай, когда все тестовые задания являются однородными, а предлагаемые в процессе тестирования возможные варианты ответов на тестовое задание не содержат явной или замаскированной подсказки.

К сожалению, такая подсказка может иметь место в процессе тестирования, в том числе и тогда, когда составители тестовых заданий старались избежать этого. Например, обучаемый может не знать правильного ответа на тестовое задание, но, вчитавшись в условие и варианты ответов, вдруг находит в одном из них слова или фразы, которые когда-то слышал от преподавателя на занятии. Вполне возможно, что этого «воспоминания» будет достаточно для выбора правильного ответа. Но это другой механизм решения тестовых заданий, который в данной работе не рассматривался. Безусловно, он приведет к еще большему смещению оценки знаний обучаемого и снижению ее достоверности.

В заключение отметим, что условия рассмотренных выше задач подобраны так, чтобы показать значимость рассматриваемой особенности. Увеличение количества вариантов ответов в каждом тестовом задании, к примеру, будет способствовать тому, что превышение тестовой оценки относительно истинной, если оценивать его в среднем, может стать несколько меньшим. Однако исключить смещение тестовой оценки относительно истинной нельзя (теоретически возможно в случае, если количество вариантов ответов в каждом тестовом задании будет стремиться к бесконечности, что практически недостижимо). К этому следует добавить, что с увеличением количества вариантов ответов неизбежно будут возрастать затраты времени на тестирование, причем, скорее всего, в геометрической прогрессии.

Какие выводы можно сделать из вышеизложенного? Их два.

Во-первых, оценка знаний обучаемого по результатам тестирования не является точной и содержит в себе некоторую погрешность.

Во-вторых, *любая* оценка знаний, как представляется, не может быть точной и будет содержать некоторую погрешность. Преимущество тестирования в том, что указанную погрешность можно минимизировать, используя математическую модель тестирования. Оптимизационные расчеты по такой модели должны стать неотъемлемой частью любой тестовой разработки.

Литература

1. Сердюков В. И. О смещении оценки знаний и умений обучаемого при компьютерном тестировании // Сборник трудов международной научно-практической конференции «Информационные технологии в образовании, науке и производстве». Серпухов: Комитет по образованию, 2008.

2. Сердюков В. И. Сравнительная оценка результатов компьютерного тестирования знаний студентов вузов // Ученые записки ИИО РАО. 2007. Вып. 24.

3. Феллер В. Введение в теорию вероятностей и ее приложения: В 2-х т. М.: Мир, 1984.

Уважаемые читатели!

Приглашаем вас подписаться на журнал

«Информатика в школе»

Подписные индексы журнала в каталоге агентства «Роспечать»:

для индивидуальных подписчиков — 81407

для предприятий и организаций — 81408

в объединенном каталоге «Пресса России» — 45751



ПЕДАГОГИЧЕСКИЙ ОПЫТ

О. Ю. Заславская,

*доктор пед. наук, зам. зав. кафедрой информатизации образования
Московского городского педагогического университета,*

Н. Л. Галеева,

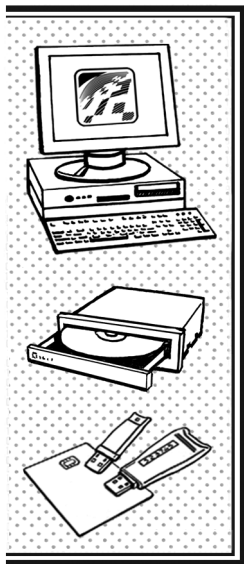
*зам. зав. кафедрой управления образовательными системами факультета
повышения квалификации и профессиональной переподготовки работников образования
Московского педагогического государственного университета*

ПОДХОДЫ К УПРАВЛЕНИЮ УЧЕБНОЙ ДЕЯТЕЛЬНОСТЬЮ УЧАЩИХСЯ НА УРОКАХ ИНФОРМАТИКИ

Содержание понятия «учебная деятельность» неоднократно подвергалось, с одной стороны, анализу в научно-педагогической теории и, с другой стороны, интуитивному переосмыслению со стороны учителей. И это имеет свое обоснование, если учесть, что учебное занятие, как открытая информационная и материальная система, не может не отражать изменения внешних социальных условий, прогресса науки и техники, приоритетов системы ценностей личности и государства.

Современный учитель информатики должен обладать базовыми качествами учителя-предметника, а также качествами учителя-новатора, учителя-экспериментатора, учителя-организатора, лидерскими, управленческими качествами, быть личностью, яркой индивидуальностью. В подготовке учителя информатики, с одной стороны, необходимо формировать способность к осуществлению обучения основам научного мировоззрения, а именно представлению об информации как одном из основополагающих понятий, на основе которого строится современная научная картина мира, навыкам жизни в информационном обществе, пониманию особой роли информации, информационных процессов и информационных технологий в развитии общества. С другой стороны, учителю информатики необходимо осуществлять развитие способностей к общению, коллективной деятельности, активизацию процессов самопознания, формирование у школьников нового типа мышления — информационного, оперативного, направленного на подготовку к выбору и принятию оптимальных решений.

Однако упомянутая выше открытость учебного занятия как системы требует от учителя информатики не только глубоких знаний основ дидактики, но и целостной системы умений: проанализировать потоки новой информации, оценить ее дидактический потенциал в реальных условиях, спрогнозировать результат ее использования, организовать ее материальное воплощение, а затем снова проанализировать, но уже результат воплощения новой информации в процесс обучения. При этом совершенно необязательно, что эта новая информация будет представлять собой новую методику или новую учебную программу. Учитель информатики, способный осуществить рефлексию собственной деятельности по всем правилам деятельност-



ного подхода, анализирует, оценивает потенциал содержания учебного материала, прогнозирует как его применение, так и применение учебных приемов и остального дидактического оснащения учебных занятий.

В методологии исследований выделяют два основных пути повышения эффективности обучения информатике:

- официально регулируемый, ориентированный на сформированные требования к уровням подготовки, подтверждаемые федеральными образовательными стандартами;
- инновационный, ориентированный на образование личностно ориентированного типа с приоритетной задачей развития.

Путь, основанный на регулировании учебного процесса образовательными стандартами, в настоящее время активно внедряется в школьную практику. Но двигаться только по этому пути недостаточно, так как возникает потребность осознанно формировать интерес к процессу получения знаний, рефлексии мыслительной и практической деятельности как обучаемых, так и обучающихся. Педагогическая рефлексия предполагает направленность на осмысление и осознание учебной деятельности, ее целей, содержания, результатов и методов их получения путем наблюдения и анализа своих действий субъектами обучения. Объективной основой этому может стать измерение характеристик обученности на всех этапах, сравнение ее с требуемыми нормами и определение путей их достижения. Рефлексия обучаемого базируется на его самоопределении при сопоставлении своих норм, потребностей и способностей с внешними нормами и приведения их затем в соответствие. Рефлексия обучающего основана на сопоставлении совокупных результатов обучения с социальными требованиями, анализе эффективности каждого из элементов дидактической системы, определении их действующих факторов и степени их дифференцированного и интегрального влияния на обученность. В результате такого анализа возникает возможность оперативного управления учебным процессом, общей, особенной и индивидуальной коррекции.

Из всех профессиональных компетенций современного учителя именно высокий уровень управленческих умений — анализировать, планировать, организовывать реализацию и оценивать результаты блока учебных занятий или одного урока как целостной системы — оснащает учителя «средствами навигации» в океане профессионально значимой информации.

И несомненно, главным из этих средств является умение учителя ставить цели учебного занятия или блока занятий (учебного модуля), так как именно цели являются системообразующим компонентом в любом социальном проекте, к которым относится и учебная деятельность. Правильно поставленные цели:

- объединяют содержание, средства, методы в единый процесс;
- обеспечивают учителю критерии для оптимального выбора способов и форм их достижения и диагностики;
- послужат основой для формирования и самоформирования учебно-познавательной мотивации, направленной не только на результат, но и в первую очередь на учебную деятельность.

Педагогически грамотно спроектированная цель учебной деятельности ученика очень важна еще и потому, что является для ученика опорой для развития организационных общеучебных навыков: умения планировать свою деятельность, умения отслеживать результаты собственной деятельности.

Все вышесказанное позволяет сделать вывод, что цели как целой учебной темы, так и каждого конкретного урока должны быть «открыты» для ученика и приняты им. Для этого можно и нужно не только представлять ученикам «готовые» цели, но и обсуждать цели и план вместе с учениками.

Данные рекомендации целесообразно использовать при изучении аппаратных основ компьютера, принципов работы и взаимодействия устройств компьютера, системы устройства компьютера. А для изучения темы «*История развития вычислительной техники*» этот алгоритм позволяет вскрыть иерархические связи в различных поколениях вычислительной техники: рассказывая ученикам об истории

вычислительной техники, необходимо акцентировать внимание учеников на роли физических открытий, исторических аспектов, влияния личности исследователей на развитие вычислительной техники. При этом исторические и технические связи осознаются учениками на уровне целостных образов, а не только на уровне теоретических понятий.

Если такую работу по «делегированию предметного целеполагания» с учениками проводить систематически, то через некоторое время учащиеся (особенно в старших классах) смогут самостоятельно ставить цели и планировать собственную деятельность не только на уроке, но и для целого учебного модуля.

Так, например, совместно с учащимися (с большей долей их самостоятельности) можно структурировать изучение темы «История развития вычислительной техники», выстроив логическую структуру целей учебного модуля по алгоритму изучения процессов (табл. 1).

Таблица 1

Логическая структура целей учебного модуля

<i>Алгоритм деятельности, предложенный учащимися</i>	Соотнесение выделенных целей с программными требованиями
<i>Объекты процесса развития вычислительной техники</i>	Элементная база. Физические и технические открытия. Понятие этапа развития вычислительной техники. Доказательства непрерывности процесса развития вычислительной техники
<i>Факторы и условия процесса развития вычислительной техники</i>	От простого к сложному. Технический прогресс. Производительность. Децентрализация. Совместимость и интеграция. Программное обеспечение
<i>Направления и этапы процесса развития вычислительной техники</i>	Теории о способах хранения, передачи и преобразования информации. Кодирование информации. Теории о построении вычислительных машин. Прогресс и регресс в развитии вычислительной техники. Хронологическая таблица
<i>Результаты процесса развития вычислительной техники</i>	Архитектуры и платформы вычислительной техники. Разнообразие вычислительной техники. Классификация как отражение эволюции развития вычислительной техники
<i>Эволюция процесса развития вычислительной техники</i>	Нанотехнологии и современное производство вычислительной техники. Этапы развития информатизации общества. Современные теории развития вычислительной техники

Эти цели ученики могут записать в рабочие тетради на первом уроке и/или представить в виде плаката. И каждый урок начинать с обращения к этим целям, которые определяют уже «пройденный путь» и мотивируют на следующий этап работы. Ученикам, пропустившим некоторые уроки, становится проще самостоятельно освоить материал, так как на вводном уроке рядом с последовательностью целей можно определить параграфы учебника, содержание которых соответствует поставленным целям. Здесь же можно предоставить список дополнительной литературы по изучаемой теме, адреса образовательных сайтов в Интернете, перечень электронных образовательных ресурсов и т. д. Эта работа на уроке занимает не более 5 минут, но ценность этих минут очень важна — формируется важнейший общеучебный навык работы с источниками информации.

Таблица 1 отражает одну из важнейших методологических управленческих функций учителя — последовательность действий при проектировании целей и планировании учебной деятельности:

- сначала выделяются цели, сопоставленные с алгоритмом изучения и зоной ближайшего развития учащихся (по их вопросам к теме);
- затем учитель соотносит эти цели с программными требованиями (в виде обязательных к изучению тем) и корректирует их;
- далее педагог планирует конкретную учебную работу, распределяя учебные задачи по урокам, используя имеющиеся дидактические средства и знания о возможностях учащихся в конкретных классах (так, работу по исследованию этапов развития вычислительной техники можно дать в виде домаш-

ней работы, а диагностику знаний о развитии вычислительной техники можно провести как этап занятия, не затрачивая для этого отдельный урок).

К сожалению, на практике очень часто приходится встречаться с ситуациями, когда учитель или пропускает этап постановки целей, или, в процессе подготовки к учебным занятиям, сначала распределяет содержание программы по урокам, а потом к каждому уроку «придумывает цели».

При подготовке педагогов в области методики обучения информатике необходимо предусматривать освоение учителем конкретного учебного предмета и методики его изложения с позиции актуального проявления в деятельности базовых и специальных компетенций, в числе которых теоретические и практико ориентированные знания, умения и навыки в области управления; приобретение теоретических знаний и практического опыта в области управления собственной деятельностью и учебно-познавательной деятельностью учащихся в процессе обучения информатике.

Можно предложить несколько правил-рекомендаций, использование которых позволит учителю информатики оптимизировать подготовку к урокам, действуя по законам теории управления:

- приступая к работе над проектированием учебной темы, сначала определить, **каких результатов** необходимо достичь ученику (по нормативным документам, в которых прописаны требования к знаниям, умениям и навыкам по данной теме или *предметные, личностные, метапредметные и интегрированные результаты*);
- учебные цели темы для ученика необходимо сформулировать **диагностично и операционально** (избегать формулировок «познакомиться с разными видами информации», так как такая формулировка может вызвать затруднения в способах осуществления данной цели и способах проверки ее достижения, например: *«знать и уметь определять различные виды информации»* или *«уметь отличать один вид информации от другого»*);
- учебные цели необходимо **сообщить ученику на первом же уроке** и сопроводить его в течение всей работы над темой (в идеале, если сформулированные цели будут для ученика *результатом коллективной и/или его индивидуальной работы*);
- содержание учебных целей для ученика должно отражать **реалии окружающего его мира** (необходимо конкретизировать учебные цели для ученика в зависимости от того, каким образом их можно использовать в качестве учебных ресурсов);
- в формулировке содержания учебных целей для ученика учителю необходимо отображать и учитывать **особенности образовательной программы** конкретной школы;
- на основе сформулированных целей для ученика **конструируется структура и содержание** итоговой диагностики по теме – каждая поставленная ученику цель должна быть проверена в зачетной работе.

Необходимо предостеречь учителя от часто встречающейся ошибки: учителя зачастую дифференцируют вопросы для зачета не по глубине содержания, а исходя из кажущейся сложности целей. Так цель *«знать типы и функции устройств компьютера»* проверяется в обязательной части зачета, а *знание сущности теории аппаратного построения компьютера* выносится в дополнительную часть, содержащую более сложные вопросы. В этом случае целесообразнее в обязательной части зачета предложить ученикам перечислить положения теории построения компьютерной техники, а в дополнительной части сделать выводы из положений этой теории, привести имена ее авторов.

Практика показывает, что не всем учителям удается легко выполнить такие требования к конструированию контрольных работ или других материалов для зачета по теме. Учителю информатики можно попробовать в качестве тренировки преобразовать цели для ученика, придав каждой цели разный уровень усвоения знаний умений и навыков по изучаемой теме (табл. 2).

Формулировка целей для учащихся по уровням обучаемости (учебных возможностей) на примере учебной темы «Аппаратное обеспечение компьютера»

Целеполагание для ученика	Уровень I (репродуктивный)	Уровень II (продуктивный)	Уровень III (творческий)
Знать названия и функции устройств компьютера	Знать названия, функции и принцип работы основных устройств компьютера (1)	Уметь различать устройства компьютера на схеме, рисунке и уметь назвать их составные части	Уметь нарисовать функциональную схему компьютера, объяснить принцип работы
Уметь различать назначение и типы компьютера по составу устройств	Уметь по составу компьютера определить назначение устройства и его тип (2)	Уметь определить тип, назначение компьютера по составу устройств	Уметь «сконструировать» заданный тип компьютера
Знать сущность принципов построения архитектуры компьютера	Перечислить принципы построения архитектуры компьютера и назвать авторов (3)	Уметь проиллюстрировать примерами принципы построения компьютера	Уметь сделать выводы, обобщения из принципов построения компьютера
Иметь представление о назначении, составе и характеристиках основных устройств компьютера	Знать правила работы с компьютером, уметь подключить и проверить его работоспособность (4)	Знать области применения компьютера	Уметь построить алгоритм исследования необходимого и достаточного набора устройств компьютера в соответствии с конкретной задачей

В дальнейшем останется только преобразовать каждую цель в конкретное задание и разнести эти задания в разные части зачетной работы.

Вариант зачетной работы по теме «Устройства ПК»

Обязательная часть.

1. Соотнесите названия устройств ПК с их функциями:

Устройства ПК	Функции
Дигитайзер	Ввод информации
CF-карта	Вывод информации
Процессор	Хранение информации
Графопостроитель	Обработка информации
ADSL	Передача информации

2. К какому типу относятся устройства, если они имеют в своем составе такие компоненты:

I	II	III
Регистры АЛУ УУ Сумматор	Экран Картридж «Перья»	Клавиши с символами Фотоприемный элемент Планшет

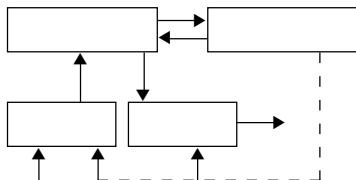
3. Перечислите принципы фон Неймана.

4. Расставьте последовательность этапов прохождения информации по функциональной схеме компьютера:

- устройства ввода информации;
- внешняя память;
- процессор;
- устройства вывода;
- внутренняя память компьютера.

Дополнительная часть.

1. Подпишите названия устройств компьютера в функциональной схеме компьютера.



2. Перечислите устройства, которые будут задействованы от момента нажатия буквенной клавиши клавиатуры до появления соответствующего символа на экране монитора в соответствии с магистрально-модульным принципом архитектуры ПК.

3. Назовите автора(ов) магистрально-модульного принципа построения компьютера и значение этого принципа.

4. Укажите общее и отличительные особенности в применении магнитных и лазерных дисков.

5*. С помощью стрелок установите взаимосвязь между блоками схемы:



6*. Составьте интеллект-карту или систему понятий из следующих понятий, дайте название составленной карте: компьютер, типы компьютеров, архитектура компьютера, принцип программного управления, принцип однородности памяти, принцип адресности памяти, магистрально-модульный принцип, устройства ввода, устройства вывода, устройства обработки, устройства передачи, шина адреса, шина управления, шина данных.

7*. Какой вывод из положений теории фон Неймана можно сделать о функционировании компьютера?

8*. Подберите и обоснуйте набор устройств, необходимый для построения компьютера для школьника.

Если ученик в процессе изучения темы не получил ни одной «тройки», ему можно предложить выполнить только четыре задания из дополнительной части. Остальные сначала выполнят задания обязательной части и, если претендуют на оценку выше, чем «3», могут выбрать задания из дополнительной части. Задания «со звездочкой» оцениваются по высшему баллу, выполнение других заданий дополнительной части зачета оценивается учителем с учетом полноты и точности каждого ответа, на «4 балла» ученик должен ответить минимум на четыре первых вопроса из дополнительной части.

Таким образом, показатели уровня обученности, воспитанности, развития познавательных интересов и другие заведомо носят субъективный характер, однако подобные измерения, производимые по научно обоснованной методике и в сходных условиях, позволяют проследить тенденции их изменения и способствовать повышению эффективности обучения информатике.

Информация о степени развития умений управлять деятельностью может служить аналитической основой для оценки реализации ведущей цели образования — перевода учащегося из режима реагирования в субъектную позицию.

Р. М. Газарян,

*ст. преподаватель лицея для одаренных детей
при Кабардино-Балкарском государственном университете, г. Нальчик,*

В. Г. Петросян,

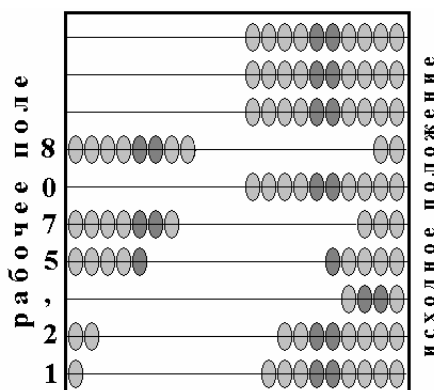
*канд. физ.-мат. наук, доцент, директор лицея для одаренных детей
при Кабардино-Балкарском государственном университете, г. Нальчик*

ОТ АБАКА ЧЕРЕЗ СЧЕТЫ К КОМПЬЮТЕРУ

При изучении темы «Системы счисления» у учащихся возникают вполне естественные психологические трудности. Например, трудно воспринять, как это после числа 7 идет не 8, а 10 (в восьмеричной системе). Еще труднее понимать, как это буквы А, В, С, D, Е и F вдруг оказываются среди цифр и с ними можно производить еще и какие-то действия (в шестнадцатеричной системе). Даже человеку, знакомому с разными системами счисления, трудно производить математические действия (в частности, сложение и умножение) над числами в недесятеричной системе. Здесь сказывается инерционность человеческого мышления, в силу которой при умножении чисел человек сразу пользуется таблицей умножения в привычной десятичной системе. А таблицу умножения человек изучает не на уровне разума, понимания — он ее учит, заучивает наизусть. Когда ученик учит таблицу умножения, ему никогда в голову не придет вопрос, почему пятью пять — двадцать пять, а семью восемь — пятьдесят шесть, он это воспринимает как данность. И при умножении в недесятеричных системах счисления эта заученная таблица умножения всплывает в сознании и создает дополнительные трудности.

Эти проблемы — психологического отторжения недесятеричных систем счисления и инерционного применения десятичной таблицы умножения, — на наш взгляд, легко решаются применением обычных бухгалтерских счетов при изучении указанных тем, по крайней мере в начале изучения темы.

Традиционные счета представляют собой ряд спиц с нанизанными на каждой из них десятью костяшками. Каждая спица соответствует разряду числа: первая спица — разряд единиц, вторая — разряд десятков, третья — сотен и т. д. Спица с малым количеством костяшек носит разделительный характер. На рис. 1 показано, как на спицах отложить (записать) число 8075,21.



*Рис. 1. Десятеричные счета
с отложенным числом 8075,21*

Возьмем теперь счета, в которых на каждой спице не десять, а, допустим, шесть костяшек, т. е. роль десятки на такой спице играет цифра 6 и добавлять одну костяшку на следующей спице надо в положении, когда все костяшки (шесть!) находятся на рабочем поле. Тогда, очевидно, что отложить, например, числа 8075 или 296 на таких счетах невозможно, ведь мы не сможем отложить на одной спице более шести костяшек. Значит, на таких счетах максимальная цифра, которой можно

оперировать, — цифра 5. Аналогично, если на одной спице будет восемь костяшек, то роль «десятки» будет играть восьмерка и оперировать можно будет только цифрами от 0 до 7, а если шестнадцать, то от 0 до 15. Но так как в используемой нами десятичной системе всего десять цифр, то для написания чисел, к примеру, в шестнадцатеричной системе надо «придумать» еще шесть цифр. Недостающие цифры от 10 до 15 принято обозначать первыми буквами латинского алфавита: А — 10, В — 11, С — 12 и так далее до F — 15.

Если из картона или пластмассы сделать «гребенку», то, отделяя гребенкой по две костяшки с каждой спицы, получим счеты для работы в восьмеричной системе счисления (рис. 2).

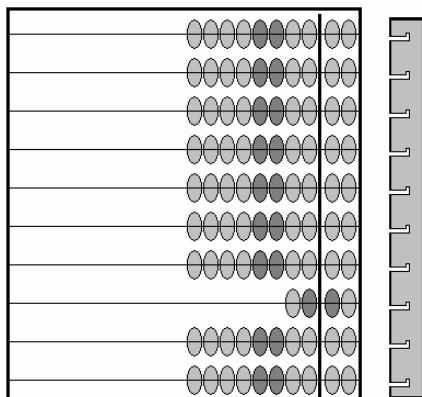


Рис. 2. Счеты для вычислений в восьмеричной системе счисления

Сложим теперь на счетах с восемью костяшками (в восьмеричной системе счисления), например, числа 543 и 607.

Отложим сначала на соответствующих спицах первое слагаемое — три, четыре и пять костяшек соответственно (рис. 3).

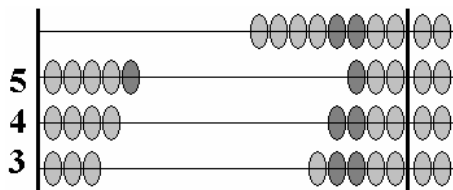


Рис. 3. Фрагмент восьмеричных счетов с отложенным числом 543

Складываем сначала на первой спице единицы. К трем имеющимся на рабочем поле костяшкам надо прибавить семь, но так как в исходной позиции всего осталось пять, то, добавив пять костяшек, получим одну «десятку». Значит, к четырем костяшкам следующей спицы (разряда) добавляем одну костяшку (одну «десятку»). А на первой спице (единиц), оставив на рабочем поле недостающие две костяшки, остальные шесть возвращаем на исходное поле (рис. 4).

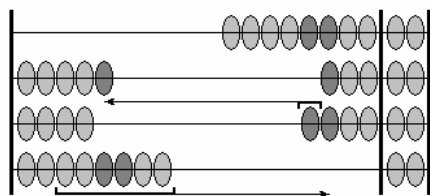


Рис. 4. Операция сложения единиц (3 + 7)

Таким образом, мы сложили единицы. Эта операция соответствует обычному (привычному для учащихся) сложению «столбиком» — к семи прибавили три, получили двенадцать, два пишем, один в уме, и даже еще и распорядились этой «единицей в уме», прибавив ее к четырем десяткам.

Перейдем теперь к следующей спице — разряду «десятков». К имеющимся на рабочем поле пяти костяшкам добавлять нечего — у второго слагаемого десятков нуль.

Перейдем теперь к разряду «сотен». К имеющимся на спице пяти костяшкам надо добавить еще шесть, но на рабочем поле имеются только три. Добавляем к имеющимся пяти три костяшки. Рабочее поле переполнено, поэтому перемещаем на следующей спице (разряде «тысяч») одну костяшку на рабочее поле, а на предыдущей спице оставляем недостающие до шести три костяшки, а остальные возвращаем в исходное положение (рис. 5).

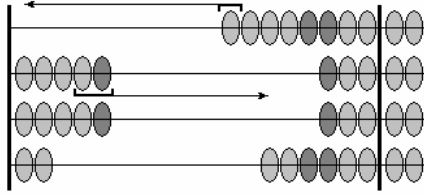


Рис. 5. Операция сложения «сотен»

На рабочем поле получается результат — 1352 (рис. 6). Мы видим, что никакой разницы в *правилах* вычислений в различных системах нет.

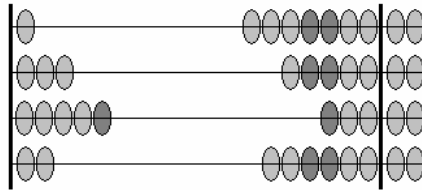


Рис. 6. Результат сложения чисел 543 и 607 в восьмеричной системе

Если взять счеты с 16 костяшками, то, отделяя гребенкой любое количество костяшек, можно сконструировать счеты для любой, меньшей 16, системы счисления.

Такой способ подачи недесятичных систем счисления не вызывает вопросов по поводу количества костяшек, так как визуально психологически между 10, 8, 2 или 16 костяшками на одной спице принципиальных различий нет. И при выполнении арифметических действий опять же не происходит психологической фиксации на числе 10, так как алгоритм вычислений оперирует не понятием «переход через десятку», а понятием «переход к следующей спице, когда предыдущая заполнена».

Умножение чисел на счетах осуществляется многократным сложением или рядом специфических приемов. Например, при умножении на 10 переносят все число одной спицей выше (операция «сдвиг»). Выполнять деление с помощью счетов гораздо труднее, чем умножать, для этого нужно запомнить целый ряд особых приемов, подчас довольно замысловатых. Мы на этом не будем останавливаться, интересующиеся могут обратиться к специальной литературе [1].

Литература

1. Голубкина Т. С. Торговые вычисления: Учебное пособие. М.: Академия, 1986.

О. В. Щербакова,

*зам. директора по информатизации образовательного процесса,
учитель информатики средней общеобразовательной школы № 1908, Москва*

К ВОПРОСАМ ПРОПЕДЕВТИЧЕСКОЙ ПОДГОТОВКИ ПО ЛОГИКЕ ШКОЛЬНИКОВ ВТОРОЙ СТУПЕНИ

Проблема возможностей усвоения младшими школьниками формальной логики как учебного предмета имеет сегодня особую актуальность в связи с ориентацией современной системы образования на реализацию идей и положений концепции развивающего обучения. Одно из главных направлений этой концепции — развитие у учащихся теоретического мышления в процессе преподавания различных учебных предметов и, прежде всего, основ формальной логики.

Известно, что в дореволюционной российской школе был учебный предмет «Классическая логика», специально направленный на развитие мышления. К сожалению, в современной школе логика как учебный предмет не нашла достойного места в учебных планах. Отсутствуют в настоящее время и конкретные программы развития логического мышления учащихся. Между тем еще К. Д. Ушинский утверждал, что логика должна стоять в преддверии всех изучаемых наук. Особенно большое внимание он уделял роли логики в начальной школе и указывал на ее важное значение в умственном развитии ребенка.

Имеющиеся исследования и анализ практического опыта показывают, что при наличии необходимых дидактических условий (учебных пособий, заинтересованности учителя, правильной организации хода урока) дети младшего школьного возраста способны к усвоению основ формальной логики. По мнению многих ученых, у младших школьников есть определенные потенциальные возможности, актуализация которых может обеспечить успешное изучение этого предмета.

Справедливо считается, что овладение знаниями в области формальной логики может в значительной степени способствовать развитию теоретического мышления учащихся начальной и средней школы и подготавливать их к более сложному обучению на последующих этапах школьного образования (в среднем и старшем звеньях). Однако решение этой задачи в настоящее время сильно затрудняется в силу того, что не разработаны специальные программы и учебные планы по основам формальной логики, отсутствуют учебники и учебные пособия, адаптированные для среднего звена школьного обучения.

Анализ психолого-педагогической и методической литературы показывает, что младшие школьники обладают определенными интеллектуальными резервами и возможностями, которые в полной мере не используются в процессе обучения. Актуализация этих возможностей (любопытности, умственной активности, потребности усваивать и закреплять знания, высокой отзывчивости на учебные ситуации, повышенной внешнеречевой активности, легкости в овладении терминологией, стремления делать всё по правилам и т. д.) с помощью изучения формальной логики может стать важным средством формирования у младших школьников теоретического мышления и их успешного умственного развития.

В процессе обучения школьников информатике немалую роль играет учитель, но главная роль принадлежит учебнику или тому учебному пособию, с которым ученик имеет возможность самостоятельно работать либо повторять пройденное. В УМК по информатике для VI класса Л. Л. Босовой изучению азов логики посвящена глава 2 «Человек и информация». Последовательность изучения выстроена следующим образом:

- Информация и знания.
- Чувственное познание окружающего мира.
- Понятие как форма мышления.
- Как образуются понятия.

- Содержание и объем понятия.
- Отношения между понятиями.
- Определение понятия.
- Классификация.
- Суждение как форма мышления.
- Умозаключение как форма мышления.

Для закрепления материала по этим темам помимо упражнений после каждого параграфа в учебнике предложено еще 38 упражнений в рабочей тетради. Все задания отобраны с учетом возраста детей (11—13 лет), представлены в разных формах (загадки, стихи, головоломки), т. е. направлены на то, чтобы заинтересовать ребят изучением данного материала.

При введении основных логических понятий «суждение» и «умозаключение» мы считаем методически правильным предложить учащимся опорный конспект и, используя групповую работу, с помощью учителя внести в него необходимые записи.

В начале урока ребята получают опорный конспект на листе формата А4 (см. приложение) и под руководством учителя выполняют в нем следующие действия:

1. Найти в учебнике определение суждения как формы мышления и внести его в опорный конспект.

2. Вычленить из текста правильное представление о языковой форме суждения и записать его в конспект.

3. Схема в опорном конспекте показывает, что суждения бывают простыми и сложными. Ученикам по образцу приведенных в учебнике общеутвердительных, общепризнавательных, частноутвердительных и частнопризнавательных высказываний предлагается придумать свои высказывания и вписать их в форму. Образцы придуманных детьми высказываний проговариваются вслух и коллективно обсуждаются.

4. Приступаем к работе со сложными суждениями. Учащиеся должны вписать словосочетание «логические связки», которые и конкретизируют, чем отличаются простые суждения от сложных. Сами же логические связки в опорном конспекте уже присутствуют.

5. Формы для заполнения определений необходимого, достаточного, необходимого и достаточного условий устроены таким образом, что определения этих условий дети должны построить сами на основе ранее изученного материала с помощью формулы вида:

$$\underbrace{\text{Видовое понятие}}_{\text{Определяемое понятие}} = \underbrace{\text{родовое понятие}}_{\text{Определяющее понятие}} + \underbrace{\text{видовое отличие}}$$

6. Следующий вид работы — вписать в соответствующий блок определение умозаключения и привести два примера умозаключений. В учебнике приводятся две формы получения умозаключений. Как показывает практика, вторая форма более трудна для детского восприятия, а вот первую ребята воспринимают достаточно легко. Для поддержания интереса учащимся можно сообщить, что первая форма получения умозаключений носит название транзитивности.

В результате данной работы заполняется вся форма, прорабатываются два параграфа из учебника и ненавязчиво закрепляется достаточно сложный материал.

Литература

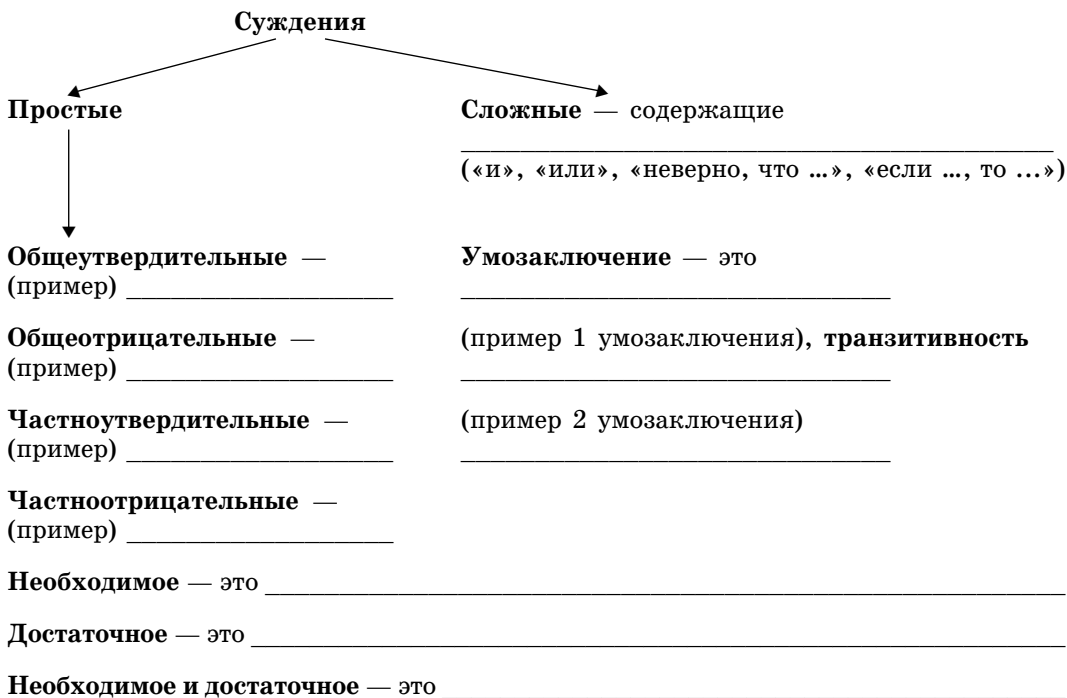
1. Босова Л. Л. Информатика: Учебник для 6 класса. М.: БИНОМ. Лаборатория знаний, 2005.
2. Лыскова В. Ю. Логика в информатике. М.: БИНОМ. Лаборатория знаний, 2006.

Опорный конспект

Тема: Суждение как форма мышления. Умозаключение как форма мышления.

Суждение — это _____

Языковая форма суждения — это _____



Уважаемые читатели!

Приглашаем вас на наш сайт www.infojournal.ru,
на котором вы можете познакомиться
с новыми учебниками по информатике
и задать вопросы авторам (ШКОЛА МАСТЕРСТВА),
узнать об условиях конкурса ИНФО и принять в нем участие.
Наша постоянная рубрика ГОРИЗОНТЫ ЦИФРОВОГО БУДУЩЕГО
регулярно пополняется новыми материалами от ведущих IT-компаний.
Ждем вас на нашем сайте. Пишите, задавайте вопросы,
предлагайте новые рубрики. Нам дорого мнение каждого из вас.
Сайт — это прямая связь между вами, уважаемые читатели, и редакцией.

В. Д. Кильдишов,

*канд. техн. наук, доцент, профессор РМАТ, доцент кафедры прикладной информатики
Одинцовского гуманитарного института*

МОДЕЛИРОВАНИЕ РОБОТОВ С ИСПОЛЬЗОВАНИЕМ ЭЛЕКТРОННОЙ ТАБЛИЦЫ MS EXCEL

Табличный процессор MS Excel позволяет создавать простейших роботов, двигающихся в различных условиях. Рассмотрим моделирование робота-футболиста, который жонглирует мячом с помощью ноги. При этом введем некоторые ограничения. Робот-футболист стоит на месте и при жонглировании ногу не сгибает в колене. Мяч в момент удара ногой поднимается вверх и под силой тяжести падает вниз до соприкосновения с ногой робота. Скорость движения ноги можно регулировать и задавать скорость мяча в момент удара. Так как при создании данной модели в первую очередь интересует кинематика, то при моделировании опустим вычислительные изменения скорости мяча от масс и скоростей ноги и мяча. Начальное положение мяча определяется координатами ноги робота во время удара.

Моделирование робота будем осуществлять последовательно по следующим этапам:

1. Определение размеров фигуры робота.
2. Разработка модели движения ударной ноги робота.
3. Моделирование вертикального движения мяча.
4. Организация временных циклов моделирования полета мяча, движения ноги и синхронизация их для обеспечения нанесения удара по мячу.

Этап 1. Создаем неподвижную фигуру робота-футболиста из простейших линий (рис. 1).

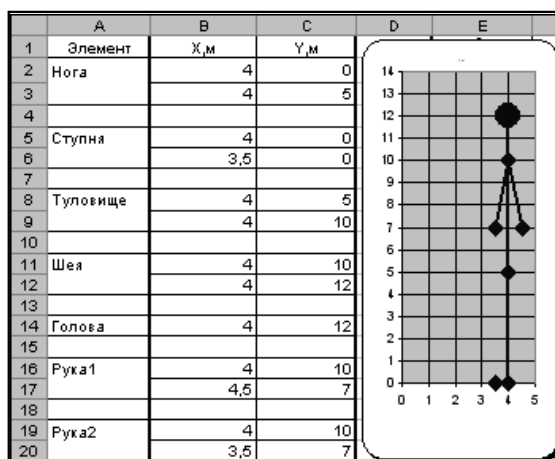


Рис. 1. Фигура робота-футболиста

Координаты концов отдельных элементов фигуры задаем в соседних парах ячеек. Чтобы отдельные элементы не соединялись, между парами строк с ячейками, содержащими координаты, пропускаем по одной строке. Голову задаем одной координатой. После задания координат можно обратиться к мастеру построения диаграмм, выбрать пункт *Точечная диаграмма*, на которой значения соединены отрезками, и откорректировать размеры отдельных элементов фигуры робота.

После исправления размеров элементов фигуры можно через контекстное меню выбрать пункт *Формат рядов данных*, установив приемлемую толщину линий и маркер концов элементов фигуры робота.

Для отображения головы робота необходимо навести указатель на маркер головы, щелкнуть левой клавишей мыши для выделения точки и через контекстное меню и пункт *Формат точки данных* задать подходящий размер, цвет и фон точки.

Целесообразно установить максимальные и минимальные значения шкал диаграммы, а также растянуть диаграмму для установки одинаковых размеров цен деления шкал для обеспечения неискаженного изображения робота по горизонтали и вертикали.

Этап 2. Теперь можно перейти к заданию координат ударной ноги робота. Исходное положение ударной ноги совпадает с неподвижной ногой. Пусть нога движется с постоянной заданной угловой скоростью до точки встречи с мячом и обратно в исходное положение (рис. 2). Ступня всегда перпендикулярна ноге. Нужно вычислить угол, на который поворачивается нога $\alpha_{\text{удара}}$.

$$\alpha_{\text{удара}} = \alpha - \alpha_s, \quad (1)$$

где α — угол, на который поворачивается крайняя точка ступни, $\alpha_s = \arctg(S/N)$ — угловая величина ступни, S — размер ступни, N — длина ноги.

Угол α находим из условия равенства координаты X крайней точки ступни нулю в момент удара по мячу

$$\begin{aligned} X_0 - (N^2 + S^2)^{0.5} \cdot \sin \alpha &= 0. \\ \alpha &= \arcsin (X_0 - (N^2 + S^2)^{0.5}). \end{aligned} \quad (2)$$

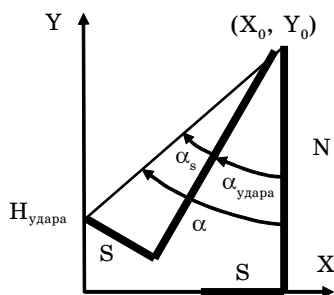


Рис. 2. Положение ноги в исходном положении и при ударе по мячу

При заданной угловой скорости ноги ω и $\alpha_{\text{удара}}$ можно найти время движения ноги до удара:

$$T_{\text{удара}} = \alpha_{\text{удара}} / \omega. \quad (3)$$

Это время в дальнейшем используем для организации цикла жонглирования мячом.

Для моделирования движения ноги определим выражения для изменения координат отрезков, имитирующих ногу и ступню, в зависимости от угла поворота $\alpha_{\text{тек}}$, который изменяется в диапазоне от нуля до $\alpha_{\text{удара}}$. Одна из координат отрезка ноги (X_0, Y_0) остается неизменной, а координаты нижнего конца ноги определяются по формулам:

$$\begin{aligned} X_{N_{\text{тек}}} &= X_0 - N \cdot \sin \alpha_{\text{тек}}, \\ Y_{N_{\text{тек}}} &= Y_0 - N \cdot \cos \alpha_{\text{тек}}. \end{aligned} \quad (4)$$

Одни координаты пятки соответствуют координатам движущегося конца ноги, а другие координаты ступни рассчитываются по формулам с учетом перпендикулярности ступни к ноге:

$$\begin{aligned} X_{S_{\text{тек}}} &= X_{N_{\text{тек}}} - S \cdot \cos \alpha_{\text{тек}}, \\ Y_{S_{\text{тек}}} &= Y_{N_{\text{тек}}} + S \cdot \sin \alpha_{\text{тек}}. \end{aligned} \quad (5)$$

Этап 3. Остается определить выражения для расчета вертикального движения мяча, который в момент удара начинает движение с заданной начальной скоростью V_0 . Вектор скорости в момент удара направлен вертикально вверх.

Высота мяча $Y_{\text{мяча}}$ вычисляется по формуле

$$Y_{\text{мяча}} = H_{\text{удара}} + V_0 \cdot T - g \cdot T^2 / 2, \quad (6)$$

где $H_{\text{удара}}$ — начальная высота полета, $g = 9,8 \text{ м/с}^2$ — ускорение свободного падения, T — текущее время.

Текущая скорость $V_{тек}$ определяется выражением

$$V_{тек} = V_0 - g \cdot T. \tag{7}$$

Эта скорость понадобится для получения текущих характеристик полета мяча.

Для определения в диаграмме максимального значения шкалы по Y целесообразно иметь значения максимальной высоты подъема мяча $Y_{макс}$, которая вычисляется по формуле с учетом высоты удара по мячу $H_{удара}$:

$$Y_{макс} = H_{удара} + V_0^2 / (2 \cdot g). \tag{8}$$

Этап 4. При моделировании нужно знать полное время полета мяча $T_{пол}$, которое рассчитывается по формуле

$$T_{пол} = 2 \cdot V_0 / g. \tag{9}$$

При известном $T_{пол}$ мяча уже можно организовать циклы жонглирования мячом.

Заносим в таблицу выражения для вычисления координат центра тяжести мяча. Координата мяча по оси X равна нулю. Сформировав счетчик текущего времени T, можем проверить правильность моделирования полета мяча, задавая различные значения счетчика текущего времени. Через контекстное меню точки мяча и пункт *Формат точки данных* можно задать подходящий размер, цвет и фон мяча (рис. 3).

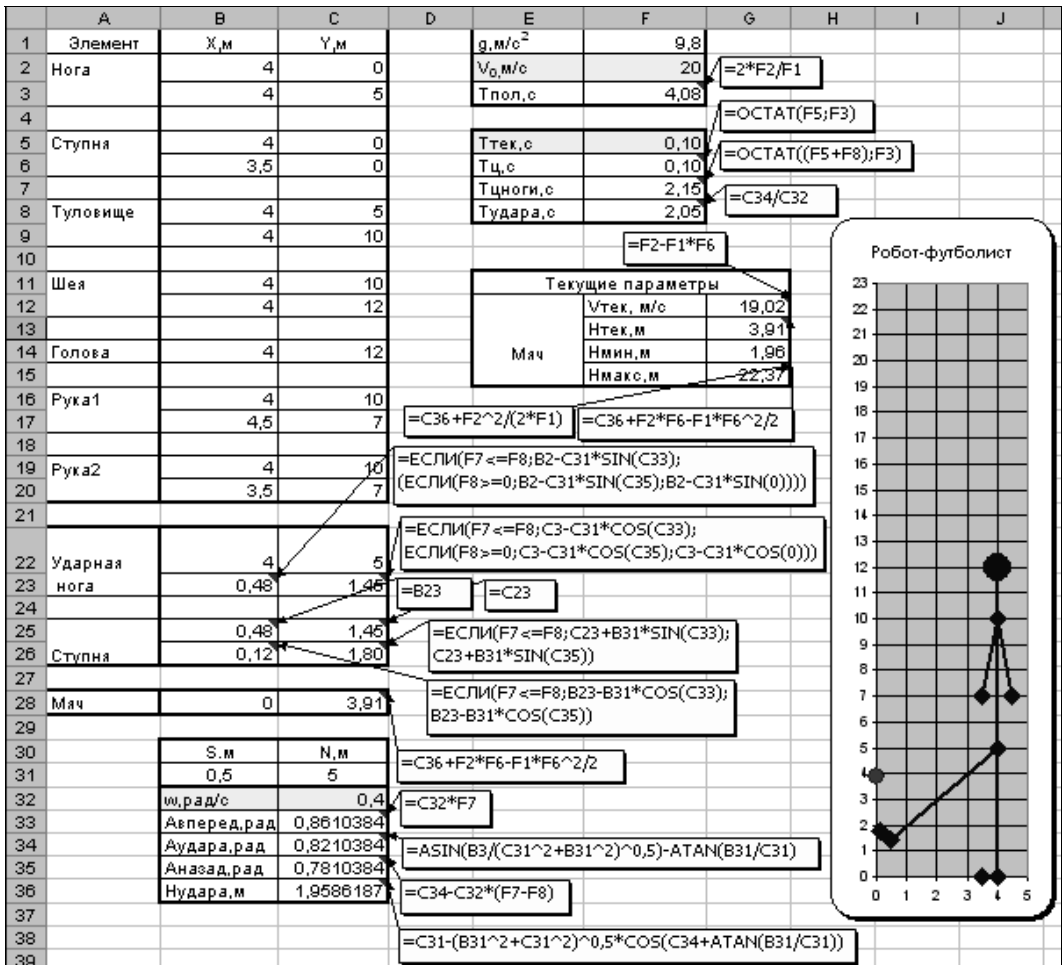


Рис. 3. Модель работа-футболиста с мячом после удара

После проверки моделирования полета мяча приступаем к моделированию движения ноги робота-футболиста с применением формул (4), (5).

Однако полет мяча и движение ноги не будут согласованы для моделирования удара по мячу.

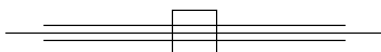
Остается согласовать движение ноги для нанесения удара по мячу в каждом цикле. Время цикла зависит от времени полета $T_{\text{пол}}$ мяча от точки удара $H_{\text{удара}}$. Во время цикла наносится один удар по мячу. Таким образом, для формирования счетчика времени внутри цикла ($T_{\text{ц}}$) используем функцию $\text{ОСТАТ}(T; T_{\text{пол}})$. Вопрос точности вычисления координат мяча и положения ноги опускаем. С этого момента для моделирования полета мяча используется $T_{\text{ц}}$. Убеждаемся в правильности моделирования цикла полета мяча, который движется от точки $H_{\text{удара}}$ до $Y_{\text{макс}}$ и обратно. Теперь в момент достижения мяча точки удара нужно обеспечить совпадение положения ноги с мячом. Так как нога движется с определенной скоростью, то ее движение начинается за интервал времени $T_{\text{удара}}$ до момента, равного $T_{\text{пол}}$, когда мяч достигает точки удара. Причем после удара нога должна вернуться в исходное положение. В этом случае угол $\alpha_{\text{тек}}$ изменяется от $\alpha_{\text{удара}}$ до нуля. Сразу нужно отметить, что при моделировании необходимо выполнить условие $T_{\text{пол}} > 2 \cdot T_{\text{удара}}$, которое обеспечивает возврат ноги в исходное положение. Если условие не выполняется, то необходимо изменить угловую скорость ноги ω .

Для моделирования движения ноги также организуем счетчик времени цикла $T_{\text{цноги}} = \text{ОСТАТ}(T + T_{\text{удара}}; T_{\text{пол}})$, который смещен вперед относительно $T_{\text{ц}}$ на $T_{\text{удара}}$.

Внеся соответствующие выражения и начальные данные, можно приступить к моделированию жонглирования мячом путем изменения счетчика текущего времени T . На рис. 3 в примечаниях представлены реализованные выражения формул, которые были рассмотрены выше. Следует отметить, что для упрощения расчетов при движении ноги предварительно вычисляется угол поворота при подъеме ноги $A_{\text{вперед}} = \omega \cdot T_{\text{цноги}}$. При условии $T_{\text{цноги}} \leq T_{\text{удара}}$ в формулах (4) и (5) $\alpha_{\text{тек}} = A_{\text{вперед}}$. Одновременно вычисляется угол поворота ноги назад в исходное положение $A_{\text{назад}} = \alpha_{\text{удара}} - \omega \cdot (T_{\text{цноги}} - T_{\text{удара}})$. При условии $T_{\text{цноги}} > T_{\text{удара}}$ и $A_{\text{назад}} \geq 0$ в формулы 4 и 5 $\alpha_{\text{тек}} = A_{\text{назад}}$. При отрицательном значении $A_{\text{назад}}$, когда исходное положение уже пройдено, $\alpha_{\text{тек}} = 0$.

Изменения счетчика времени можно автоматизировать с помощью макросов и создания дополнительных функций.

При разработке других моделей робота-футболиста интересно установить ворота на некотором расстоянии и смоделировать попадание в них мяча. При этом нужно направить вектор скорости мяча под определенным углом. Особых трудностей не составит разработать модель шагающего или бегущего робота. Представленная методика разработки модели робота-футболиста позволит расширить область применения MS Excel.



УВАЖАЕМЫЕ ЧИТАТЕЛИ!

**Напоминаем вам подписные индексы
журнала «Информатика и образование»:**

Каталог агентства «Роспечать»:

70423 — для индивидуальных подписчиков;

73176 — для предприятий и организаций.

Каталог «Пресса России» — 26097.

Е. В. Невзорова,

учитель информатики гимназии № 4, г. Электросталь, Московская область

ПОСТРОЕНИЕ ТРЕУГОЛЬНИКА ПАСКАЛЯ С ПРИМЕНЕНИЕМ СРЕДСТВ ЯЗЫКА ПРОГРАММИРОВАНИЯ QBasic

Тема урока: «Построение треугольника Паскаля с применением средств языка программирования QBasic».

Цель урока: научить школьников использовать подпрограммы на примере построения треугольника Паскаля.

Тип урока: объяснение нового материала.

Ход урока

1. Организационный момент

Учитель. Сегодня на уроке мы с вами построим специальную таблицу, которая в дальнейшем поможет вам увидеть интересные закономерности и свойства чисел, входящих в нее. Также вы познакомитесь с новыми возможностями языка программирования — с подпрограммами, которые вы будете применять для решения задач.

Таблица, заполнением которой мы займемся, состоит из сочетаний, записываемых в определенном порядке.

2. Повторение

Ответьте на следующие **вопросы:**

1. Что такое сочетание?

2. Назовите сочетания, которые можно составить из цифр 1, 2, 3 по 2 элемента в каждом сочетании, и количество таких сочетаний.

Ответ. 12, 13, 23. Всего 3 сочетания.

3. Почему вы не использовали число 21, ведь это другое число?

Ответ. 21 и 12 — одно и то же сочетание, различные сочетания должны отличаться хотя бы одним элементом.

4. По какой формуле можно вычислить число сочетаний?

Ответ. $C_n^k = \frac{n!}{k!(n-k)!}$, где n — 0 и натуральное число, $0 \leq k \leq n$.

5. Какие свойства числа сочетаний вам известны?

Ответ. 1) $C_n^n = 1$;

2) $C_n^0 = 1$ (из любого количества объектов ничего не выбрать можно одним способом);

3) $C_n^k = C_n^{n-k}$. Где используется это свойство? Ответ: «При вычислении большого числа сочетаний»;

4) $C_n^1 = n$.

Однако это не все свойства сочетаний.

3. Изучение нового материала

Для знакомства с другими свойствами числа сочетаний запишем числа сочетаний построчно, начиная с $n = 0$ — на первой строчке, $n = 1$ — на второй, $n = 2$ — на третьей, ..., $n = 7$ — на восьмой в виде пирамидки. Посмотрите, какие клетки

таблицы мы можем уже заполнить, применяя свойства числа сочетаний. Заполните их. А остальные мы будем заполнять, вычисляя число сочетаний с использованием языка программирования:

$n = 0$								C_0^0											
$n = 1$								C_1^0		C_1^1									
$n = 2$								C_2^0		C_2^1		C_2^2							
$n = 3$								C_3^0		C_3^1		C_3^2		C_3^3					
$n = 4$								C_4^0		C_4^1		C_4^2		C_4^3		C_4^4			
$n = 5$								C_5^0		C_5^1		C_5^2		C_5^3		C_5^4	C_5^5		
$n = 6$								C_6^0		C_6^1		C_6^2		C_6^3		C_6^4	C_6^5	C_6^6	
$n = 7$								C_7^0		C_7^1		C_7^2		C_7^3		C_7^4	C_7^5	C_7^6	C_7^7

$n = 0$																						
$n = 1$										1												
$n = 2$										1		2		1								
$n = 3$										1		3		3		1						
$n = 4$										1		4		6		4		1				
$n = 5$										1		5		10		10		5		1		
$n = 6$										1		6		15		20		15		6	1	
$n = 7$										1		7		21		35		35		21	7	1

Прежде чем составить программу, поговорим о подпрограмме, которую мы будем использовать. Запишите тему урока: «Подпрограммы в QBASIC».

Готовый алгоритм можно использовать в другом алгоритме. Подпрограмма — это вспомогательный алгоритм, записанный на языке программирования. Вызов подпрограммы в QBASIC осуществляется в основной программе по команде `сosув n`, где n — номер первой строки подпрограммы, он должен быть больше, чем номера, встречающиеся в основной программе. Подпрограмма записывается после основной программы. Возврат осуществляется по команде `RETURN`, эта последняя команда в любой подпрограмме.

А теперь вернемся к вычислению числа сочетаний.

Вычисление числа сочетаний сводится к решению задачи: вычислить значение

$$C_n^k = \frac{n!}{k!(n-k)!}, \text{ где } n \text{ — количество объектов, из которых делается выбор, } k \text{ —}$$

количество элементов в комбинации.

Этап 1. Постановка задачи.

Дано: n, k — значения аргументов.

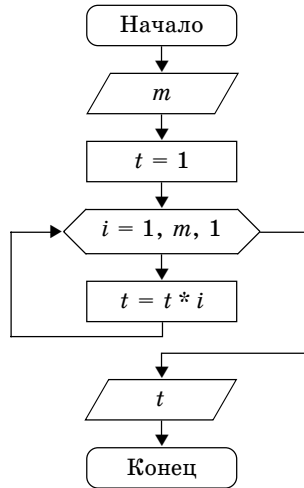
Найти: C_n^k — ?

Этап 2. Математическая формализация.

Сразу вычислить C_n^k нельзя, так как нам даны значения n и k , а в формуле используются $n!, k!, (n-k)!$, следовательно, надо ввести промежуточные переменные для этих значений. Обозначим $a = n!$, тогда $a = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$; $b = k!$, тогда $b = 1 \cdot 2 \cdot 3 \cdot \dots \cdot k$; $c = (n-k)!$, тогда $c = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-k)$. Теперь формула примет вид: $C_n^k = a/(b \cdot c)$. При составлении алгоритма будем использовать вспомогательный алгоритм. Из математической модели видно, что факториал вычисляется трижды, поэтому очевидно, что вспомогательный алгоритм — это вычисление факториала. Вам известен алгоритм вычисления $m!$. Будем использовать его в качестве вспомогательного алгоритма.

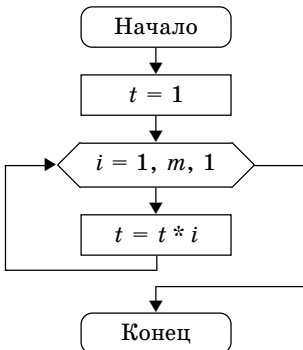
Этап 3. Построение блок-схемы.

Постройте блок-схему вычисления факториала.

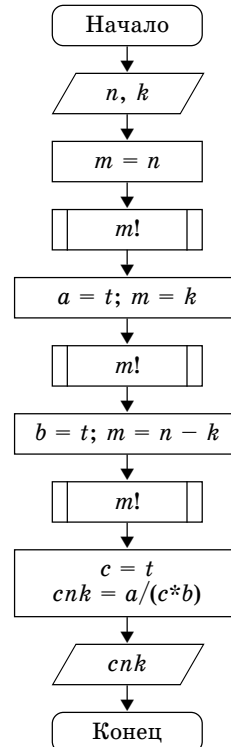


Сначала во вспомогательном алгоритме необходимо вычислить $n!$. Поэтому перед вызовом вспомогательного алгоритма необходимо в основном алгоритме выполнить команду присваивания $m = n$. Результат работы вспомогательного алгоритма будет находиться в переменной t , а в математической модели значение $n!$ должно быть присвоено переменной a , следовательно, в основном алгоритме надо присвоить a значение t . Исходные данные для вспомогательного алгоритма задаются в основном алгоритме, результат работы вспомогательного алгоритма используется в основном алгоритме, поэтому в блок-схеме вспомогательного алгоритма становятся ненужными блоки ввода и вывода.

Вспомогательный алгоритм принимает вид:



Основной алгоритм:



Этап 4. Составление программы.

Составим программу:

```
CLS
INPUT "Введите N и K"; N, K
M=N
COSUB 100
A=T: M=K
COSUB 100
B=T: M=N-K
COSUB 100
C=T
CNK=A/(C*B)
PRINT C; "из N"; "по K"; "=" ;CNK
END
100 T=1
FOR I=1 TO M
T=T*I
RETURN
```

Этап 5. Заполнение таблицы.

А теперь выполните программу, вычислите значения и заполните таблицу до 7-й строки включительно (1-й вариант заполняет строки 4, 7; 2-й вариант — 5, 6). В результате получается:

0								1								
1							1		1							
2						1		2		1						
3					1		3		3		1					
4				1		4		6		4		1				
5			1		5		10		10		5		1			
6			1		6		15		20		15		6		1	
7		1		7		21		35		35		21		7		1

Таблицу можно продолжать и дальше. Полученная таблица называется треугольником Паскаля. Подробную информацию о треугольнике вы можете найти по адресу: http://arbuz.uzpak.uz/u_treug.html.

4. Подведение итогов

Не случайно арифметическому треугольнику уделяется большое внимание, он этого заслуживает. Еще не раз мы будем обращаться к нему, к закономерностям и числам, из которых он составлен.

Сегодня вы еще раз увидели, что на компьютере можно решать задачи разного рода и различного уровня сложности. При этом главное — правильно использовать ресурсы языка программирования. Вспомогательный алгоритм — это удобное средство, которое облегчает решение трудных задач. При этом алгоритм становится лаконичным, компактным, наглядным.

5. Домашнее задание

Составить алгоритм решения задачи: вывести на экран пиктограмму времени года в зависимости от введенного с клавиатуры времени года (зима — снежинка, весна — листик, лето — солнце, осень — тучка, дождик).



ЗАДАЧИ

Р. Р. Сулейманов,

*канд. пед. наук, доцент, директор Центра информатизации образования
Башкирского института развития образования,
член-корреспондент Академии информатизации образования*

ЗАНИМАТЕЛЬНЫЕ ЗАДАЧИ С ИСТОРИЧЕСКИМ МАТЕРИАЛОМ ПО ИНФОРМАТИКЕ

Говоря о занимательности при решении задач, нельзя не упомянуть включение исторического материала в задачи по информатике.

Знакомство учеников с фрагментами истории науки имеет вполне определенные цели:

- сведения из истории повышают интерес школьников к изучению предмета;
- осуществляются межпредметные связи;
- ознакомление с историческими фактами и персоналиями расширяет кругозор учеников и повышает их общую культуру.

Ниже приведены задачи, при решении которых можно использовать исторический материал.

Теорема Ферма—Эйлера о двух квадратах

Взгляните на несколько первых простых чисел: 3, 5, 7, 11, 13, 17, 19...

Числа 5, 13 и 17 представимы как суммы двух квадратов:

$$5 = 1^2 + 2^2,$$

$$13 = 2^2 + 3^2,$$

$$17 = 1^2 + 4^2,$$

а остальные записанные числа (3, 7, 11, 19) так представить нельзя. Можно ли дать какое-нибудь объяснение этому наблюдению? Оказывается, можно, а именно справедлива следующая

Теорема. Для того чтобы простое число было представимо в виде суммы двух квадратов, необходимо и достаточно, чтобы оно при делении на четыре давало в остатке единицу.

Действительно:

$$5 = 4 \cdot 1 + 1,$$

$$13 = 4 \cdot 3 + 1,$$

$$17 = 4 \cdot 4 + 1,$$

тогда как

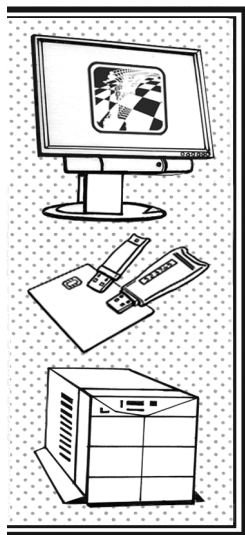
$$3 = 4 \cdot 0 + 3,$$

$$7 = 4 \cdot 1 + 3,$$

$$11 = 4 \cdot 2 + 3.$$

Говоря о доказательстве теорем, мы будем говорить о компьютерном (вычислительном) эксперименте, подтверждающем данное предположение.

Приведенная ниже программа позволяет удостовериться в справедливости приведенного утверждения.



```

program fi;

var
  n, k, i, j, pr: longint;

procedure prost(x: longint; var pr: longint);
var
  k: longint;
begin
  k:=1;
  repeat
    k:=k+1;
  until (k>round(sqrt(x))) or (x mod k=0);
  if k>round(sqrt(x)) then pr:=x;
end;

begin
  readln(n);
  for k:=1 to n do
    begin
      prost(k, pr);
      for i:=1 to trunc(sqrt(pr)) div 2+1 do
        for j:=trunc(sqrt(pr)) div 2 to trunc(sqrt(pr))+1 do
          if pr=i*i+j*j then
            if pr mod 4=1 then
              writeln('prostoe', pr, 'i=', i, 'j=', j);
            end;
          end;
        end;
      end;
    end.

```

Немного истории.

Кто и когда впервые обнаружил это явление? Есть свидетельство, что этот результат в конце прошлого года мог праздновать свое трехсотпятидесятилетие. Ибо на Рождество 1640 г. (в письме от 25.12.1640) великий Пьер Ферма (1601—1665) извещал знаменитого Марена Мерсенна, верного друга Рене Декарта и главного посредника в переписке ученых своего времени, о том, что «всякое простое число, которое при делении на четыре дает единицу, единственным способом представимо как сумма двух квадратов».

В ту пору математических журналов еще не существовало, информацией обменивались в письмах, и, как правило, результаты лишь анонсировались и не сопровождались доказательствами.

Спустя почти 20 лет после письма Мерсенну в письме к Пьеру Каркави, отправленном в августе 1659 г., Ферма приоткрывает замысел доказательства сформулированной выше теоремы. Он пишет, что основная мысль доказательства состоит в методе спуска, позволяющем из предположения, что для какого-то простого числа вида $4n + 1$ заключение теоремы неверно, получить, что оно неверно и для меньшего числа и т. д., пока мы не доберемся до числа 5, когда окончательно приходим к противоречию.

Первые доказательства были найдены Леонардом Эйлером (1707—1783) между 1742 и 1747 гг. Причем, желая утвердить приоритет Ферма, к которому он испытывал чувство глубокого уважения, Эйлер придумал доказательство, отвечающее описанному выше замыслу Ферма. Воздавая должное обоим великим ученым, теорема называется теоремой Ферма—Эйлера.

Проблема Гольдбаха

В 1724 г. Петром I была основана Петербургская академия наук. Для работы в ней были приглашены многие иностранные ученые: математики Даниил и Николай Бернулли, астроном Ж. Н. Делиль, физик Г. В. Вильфингер и другие. Без приглашения прибыл на открытие академии очень интересный человек — большой выдумщик и непоседа Христиан Гольдбах (1690—1764). Благодаря своей энергии и эрудиции, Гольдбах вскоре занял пост неперменного секретаря академии, стал ее

почетным членом. Видимо, именно его настойчивости обязана академия приезду в 1727 г. великого математика Леонарда Эйлера.

В 1742 г. Гольдбах пишет Эйлеру в Берлин: «По крайней мере кажется, что каждое натуральное число, которое больше двух, представляется в виде суммы трех простых чисел». Нужно отметить, что в то время единица считалась простым числом, поэтому числа

$$3 = 1 + 1 + 1, \quad 4 = 2 + 1 + 1, \quad 5 = 2 + 2 + 1$$

удовлетворяли сделанному предположению.

В ответном письме Эйлер написал по этому поводу: «Но что каждое четное число есть сумма двух простых, я полагаю совершенно определенной теоремой, хотя и не могу ее доказать». Именно это утверждение Эйлера для четных чисел, больших 4, и стало называться «проблемой Гольдбаха», хотя многие ученые считают проблемой Гольдбаха задачу о возможности представления каждого натурального числа, большего 6, в виде суммы трех простых. (Единицу давно уже не считают простым числом, и оснований для этого достаточно много.) Очевидно, что из решения проблемы в формулировке Эйлера следует и решение проблемы во второй постановке. Действительно, отняв от четного числа 2 и представив полученное число в виде суммы двух простых чисел, получим требуемое представление в виде трех простых; от нечетного числа нужно отнять не 2, а 3.

Об эту проблему «обломало зубы» немало крупных ученых. Выдающийся немецкий математик Эдмунд Ландау на V Международном математическом конгрессе в 1912 г. заявил, что при современном состоянии знаний проблема Гольдбаха недоступна для решения.

В 1922 г. английские математики Годфри Харолд Харди и Джон Идензор Литлвуд, опираясь на так называемую обобщенную гипотезу Римана, вывели, что каждое достаточно большое нечетное число представимо в виде суммы трех простых чисел. В 1937 г. И. М. Виноградову удалось доказать это утверждение, не опираясь на гипотезу Римана, которая не доказана до сих пор.

В этой постановке, казалось бы, проблема Гольдбаха уже практически решена, осталось лишь проверить ее справедливость для конечного числа нечетных чисел, что при современной технике не составляет большого труда. И действительно, к концу 1960-х гг. гипотеза Гольдбаха в формулировке Эйлера была проверена для всех четных чисел до $1\,000\,000\,000 = 10^9$, но число N , начиная с которого теория гарантирует выполнение гипотезы, поистине огромно: $N = e^{e \cdot 16,3} > 10^{5643232}$.

Опишем алгоритм, на основе которого составлена программа проверки проблемы Гольдбаха.

Пусть задано четное число ch . Берем первое нечетное число del ($del = 3$). Проверяем на простоту число del с помощью соответствующей подпрограммы. Находим разность $ch - del$. Проверяем на простоту это число. Если число del — простое и $ch - del$ — простое, выводим эти числа на экран. Если хотя бы одно из этих чисел составное, переходим к следующему нечетному числу $del + 2$ и вновь повторяем указанные действия до тех пор, пока соответствующее нечетное число не превзойдет $ch \div 2$.

```

program goldbach;

uses crt;

var
  ch, del, x, y, v, a, b, l: longint;

procedure prost(x: longint; var y: longint);
var
  k: longint;
begin
  k:=1;
  repeat
    k:=k+1;
  
```

```

until (k>round(sqrt(x))) or (x mod k=0);
if k>round(sqrt(x)) then y:=1;
end;

begin
  readln(ch);l:=0;
  del:=3;
  while del<=ch div 2 do
    begin
      prost(del, y); a:=y; v:=ch; ch:=ch-del; y:=0;
      prost(ch, y); b:=y;
      if (a=1) and (b=1) then
        begin
          writeln(del, ' ', ch);
          l:=l+1;
        end;
      ch:=v;
      del:=del+2;
    end;
  writeln('l=', l);
end.

```

Совершенные числа

Число называется *совершенным*, если оно равно сумме своих делителей.

Древних математиков удивляло особое свойство двух чисел — 6 и 28, каждое из которых равно сумме всех своих собственных делителей:

$$6 = 1 + 2 + 3,$$

$$28 = 1 + 2 + 4 + 7 + 14.$$

До Евклида были известны только эти два совершенных числа.

Евклид доказал, что всякое число, которое может быть представлено в виде произведения множителей 2^{p-1} и $2^p - 1$, где $2^p - 1$ и p — простые числа, является совершенным.

Благодаря своей формуле Евклид сумел найти еще два совершенных числа:

496 и 8128.

Следующее, пятое, совершенное число было обнаружено лишь в XV в. Оказалось, что и пятое совершенное число также подчиняется условию Евклида. Не удивительно, что его так долго не могли найти. Гораздо более поражает то, что в XV в. вообще смогли его обнаружить.

Пятое совершенное число равно

33 550 336,

ему соответствует значение $p = 13$ в формуле Евклида.

Пьетро Антонио Катальди, бывший профессором математики во Флоренции и Болонье и давший первым способ извлечения квадратных корней, для спасения своей души тоже занимался поисками совершенных чисел. В его записках были указаны значения шестого и седьмого совершенных чисел:

8 589 869 056 (шестое число),

137 438 691 328 (седьмое число).

Для этих чисел p равно 17 и 19 соответственно:

$$8\,589\,869\,056 = 2^{16} \cdot (2^{17} - 1),$$

$$137\,438\,691\,328 = 2^{18} \cdot (2^{19} - 1).$$

Но оставалось еще не доказанным, действительно ли эти числа являются совершенными; для этого необходимо, чтобы множители $2^{17} - 1$ и $2^{19} - 1$ были простыми.

Петербургский академик, основатель современной математики, непревзойденный вычислитель, друг М. В. Ломоносова, великий Леонард Эйлер сумел найти новую теорему о таинственных и загадочных совершенных числах. Он доказал, что все четные совершенные числа имеют вид, указанный Евклидом. Но какой вид

должны иметь нечетные совершенные числа и могут ли они вообще существовать — осталось неизвестным и до нашего времени.

Восьмое совершенное число, которому соответствует $p = 31$ в формуле Евклида:
2 305 843 008 139 952 128.

Девятое совершенное число было вычислено только в 1883 г. В нем оказалось 37 значащих цифр. Этот вычислительный подвиг совершил сельский священник из-под Перми Иван Михеевич Первушин. Он сумел вычислить для того времени самое большое простое число вида $2^p - 1$ при $p = 61$:

2 305 843 009 213 693 951

и соответствующее ему совершенное число

2 305 843 009 213 693 951 · 2^{60} .

И. М. Первушин, вычислив девятое совершенное число, совершил настоящий подвиг. Он считал, по существу, без всяких вычислительных приборов.

В начале двадцатого столетия появились первые механические счетные машины. Их появление ускорило поиски новых совершенных чисел.

Десятое было найдено в 1911 г., в нем оказалось 54 цифры:

618 970 019 642 690 137 449 562 111 · 2^{88} .

Одиннадцатое совершенное число, имеющее 65 цифр, открыли в 1914 г.:

162 259 276 829 213 363 391 578 010 288 127 · 2^{106} .

Двенадцатое совершенное число нашли также в 1914 г., оно состоит уже из 77 цифр.

Вычисленное на ЭВМ совершенное число $2^{11212} \cdot (2^{11213} - 1)$ имеет 3376 цифр.

История поисков совершенных чисел показывает, как сильно увеличивает машина возможности человека. Однако, по словам Эдмунда Ландау, одного из крупнейших специалистов в области теории чисел,

«Две проблемы остаются нерешенными до сих пор:

Имеется ли бесконечное множество четных совершенных чисел? — Не знаю.

Имеется ли бесконечное множество нечетных совершенных чисел? — Я даже не знаю, существует ли одно такое число».

Следующие задачи связаны с совершенными числами. Приведем их компьютерное решение.

Задача 1.

Докажите, что число вида $2^{k-1} \cdot (2^k - 1)$, где $2^k - 1$ — простое, является совершенным.

«Доказать» в нашем случае — это значит убедиться в правильности утверждения для вполне достаточного количества входных данных.

Решение.

```
program ZADACHA1;

uses crt;

var
  a, n, k, x, y: word;
  st, pr, sv, v, w, g: longint;

procedure stepen(x, y: word; var st: longint);
var
  i: integer;
begin
  st:=1;
  for i:=1 to y do
    st:=st*x;
end;
```

```

procedure prost(x: longint; var pr: longint);
var
  k: longint;
begin
  k:=1;
  repeat
    k:=k+1;
  until (k>round(sqrt(x))) or (x mod k=0);
  if k>round(sqrt(x)) then pr:=x;
end;

procedure sover(i: longint; var sv: longint);
var
  j, k, s: integer;
begin
  s:=1;
  j:=1;
  repeat
    j:=j+1;
    k:=i div j;
    if(i=k*j) and (j<=k) then
      begin
        s:=s+j;
        if j<k then s:=s+k;
      end;
  until j>=k;
  if s=i then sv:=i;
end;

begin
  a:=2;
  readln(k);
  stepen(a, k, st);
  v:=st-1;
  stepen(a, k-1, st);
  w:=st;
  g:=v*w;
  prost(v, pr);
  if pr<>0 then
    begin
      sover(g, sv);
      writeln('sv=', sv);
    end;
end.

```

Задача 2.

Обозначим через $\Phi(n)$, где n — натуральное число, сумму всех делителей числа n . Докажите, что если числа n_1 и n_2 — взаимно просты, то $\Phi(n_1 \cdot n_2) = \Phi(n_1) \cdot \Phi(n_2)$.

Решение.

```

program ZADACHA2;

uses crt;

var
  n1, n2, b, sum, k1, k2, k3: word;

procedure vzprost(m, n: word; var b: word);
var
  x, y: word;
begin
  x:=m;
  y:=n;

```

```

while x<>y do
  if x>y then x:=x-y else y:=y-x;
  if x=1 then b:=1;
end;

procedure sumd(ch: word; var sum: word);
var
  del: word;
begin
  sum:=0;
  for del:=1 to ch div 2 do
    if ch mod del = 0 then sum:=sum+del;
  sum:=sum+ch;
end;

begin
  readln(n1, n2);
  vzprost(n1, n2, b);
  if b=1 then
    begin
      sumd(n1, sum); k1:=sum; writeln(k1);
      sumd(n2, sum); k2:=sum; writeln(k2);
      sumd(n1*n2, sum); k3:=sum; writeln(k3);
      if k3=k1*k2 then writeln('verno');
    end;
end.

```

Задача 3.

Пусть n — четное совершенное число. Тогда $\Phi(n) = 2 \cdot n$. Представьте n в виде $2^{k-1} \cdot b$, где $k \geq 2$, а b — нечетно, и докажите, что $b = (2^k - 1) \cdot c$. Докажите далее, что $c = 1$ и $2^k - 1$ — простое число.

Решение.

```

program ZADACHA3;

uses crt;

label aa,bb;
var
  n, sv, sum, l, st, k, b, y, m: integer;

procedure sover(i: integer; var sv: integer);
var
  j, k, s: integer;
begin
  s:=1;
  j:=1;
  repeat
    j:=j+1;
    k:=i div j;
    if(i=k*j) and (j<=k) then
      begin
        s:=s+j;
        if j<k then s:=s+k
      end;
  until j>=k;
  if s=i then sv:=i;
end;

procedure sumd(ch: word; var sum: integer);
var
  del: integer;
begin
  sum:=0;

```



```

    for del:=1 to ch div 2 do
        if ch mod del = 0 then sum:=sum+del;
    sum:=sum+ch;
end;

procedure stepen(x, y: integer; var st: integer);
var
    i: integer;
begin
    st:=1;
    for i:=1 to y do
        st:=st*x;
    end;

procedure prost(x: integer; var y: integer);
var
    k: integer;
begin
    k:=1;
    repeat
        k:=k+1;
    until (k>round(sqrt(x))) or (x mod k=0);
    if k>round(sqrt(x)) then y:=x;
end;

begin
    readln(n);
    sover(n, sv);
    n:=sv;
    writeln('sv=', sv);
    if (n>0) and (n mod 2=0) then
        begin
            sumd(n, sum);
            writeln('n=', n, ' sumdel=', sum);
            l:=n;
            k:=0;
aa:    if (l>=2) and (l mod 2=0) then
                begin
                    l:=l div 2;
                    k:=k+1;
                    goto aa;
                end;
            goto bb;
bb:    writeln('k=', k);
            stepen(2, k, st);
            writeln(st);
            b:=n div st;
            writeln('b=', b);
            stepen(2, k+1, st);
            m:=st;
            writeln('m=', m);
            b:=m-1;
            prost(b, y);
            writeln('y=', y);
            if (y>1) and (y<>0) then writeln('verno');
        end;
end.

```

Литература

1. Денман И. Совершенные числа // Квант. 1991. № 5. http://kvant.mirror1.mccme.ru/1991/05/sovershennye_chisla.htm
2. Тарасенко Б. Проблема Гольдбаха и программирование // Квант. 1992. № 6. http://kvant.mirror1.mccme.ru/1992/06/problema_goldbaha_i_programmir.htm

И. Г. Овчинникова,

доктор пед. наук, профессор кафедры информатики, начальник отдела образовательной политики и мониторинга качества Магнитогорского государственного университета (МаГУ), преподаватель лицея при МаГУ,

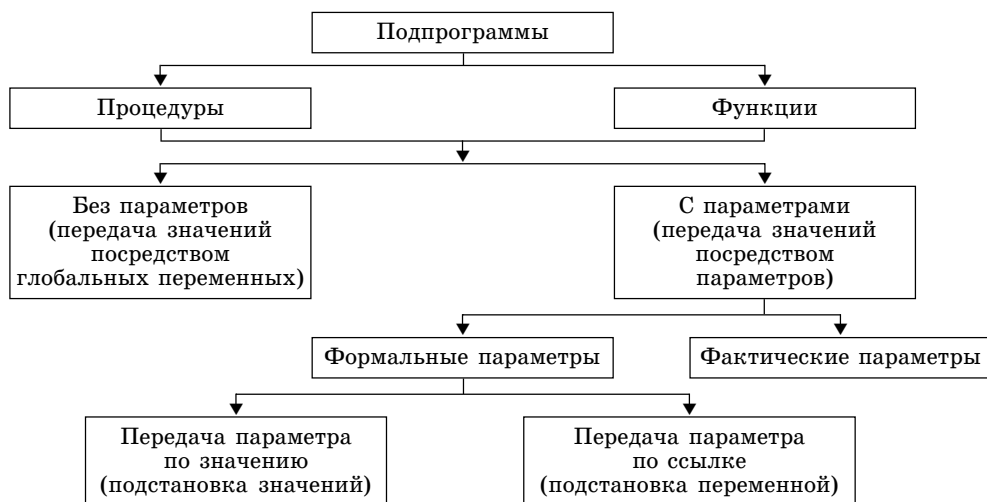
Т. Н. Сахнова,

канд. пед. наук, зам. декана факультета информатики, доцент кафедры информатики МаГУ, преподаватель лицея при МаГУ

ОСОБЕННОСТИ ИЗУЧЕНИЯ ПОДПРОГРАММ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Подпрограммы являются важным инструментом в любом языке программирования, так как позволяют писать структурированные программы. Именно поэтому при изучении данной темы в школьном курсе информатики необходим методически грамотный подход. От этого зависит приверженность учеников процедурному стилю, который является основой всех методологий программирования.

В Паскале подпрограммы реализованы через **процедуры** и **функции**. Описываются процедуры и функции в декларативной части программы.



При изучении процедур у учащихся часто возникают проблемы с пониманием **механизмов передачи данных из основной программы в процедуру**. Только последовательное рассмотрение решения одной и той же задачи на основе использования всех возможных механизмов приводит к пониманию и дальнейшему осознанному использованию процедурного стиля программирования.

Рассмотрим использование процедур как с параметрами, так и без них на примере представленной ниже задачи.

Задача 1.

Пусть даны значения x, y . Нужно вычислить:

$$u = \min(xy, x^2 + y^2),$$

$$v = \min(2u, 2(x + y)).$$

Решение.

В данной задаче нужно описать процедуру, так как само условие задачи диктует необходимость использования процедуры нахождения минимального значения из двух имеющихся.

Исходные данные для решения:

x, y — задаваемые значения.

Для решения задачи воспользуемся следующим алгоритмом:

1. Введем значения x и y .
2. Подготовим значения для поиска u .
3. Вызовем процедуру поиска минимального.
4. Повторим 2-е и 3-е действия для нахождения v .
5. Выведем результат на печать.

```
var
  x, y, u, v: real;
  temp1, temp2, min: real;

procedure min_element;
begin
  if temp1>temp2
    then min:=temp1
    else min:=temp2
end;

begin
  readln(x, y);
  temp1:=x*y;
  temp2:=sqr(x)*sqr(y);
  min_element;
  u:=min;
  temp1:=2*u;
  temp2:=2*(x+y);
  min_element;
  v:=min;
  writeln(u, v)
end.
```

В рассмотренном примере значения из основной программы в процедуру и обратно передаются посредством **глобальных переменных**. Данный способ имеет ряд неудобств:

- необходимо определять начальные значения перед вызовом процедуры;
- необходимо запоминать результат после вызова процедуры.

Для устранения этих недостатков в Паскале реализован **механизм передачи значений посредством параметров**.

Параметры процедуры делятся на формальные и фактические.

Формальные параметры локализованы в процедуре и служат для обозначения тех данных, с которыми будет работать процедура после ее вызова. При каждом вызове процедуры формальные параметры должны быть конкретизированы.

Параметры, указанные в строке вызова процедуры, называются **фактическими параметрами**.

Соответствующие формальные и фактические параметры должны быть одного типа, кроме того, должен быть указан способ подстановки. Существуют следующие способы подстановки:

- подстановка значений (передача параметра по значению);
- подстановка переменной (передача параметра по ссылке).

Рассмотрим первый способ подстановки на примере представленной выше задачи:

```
var
  x, y, u, v: real;
  min: real;

procedure min_element(temp1, temp2: real);
begin
  if temp1>temp2
    then min:=temp1
    else min:=temp2
end;
```

```

begin
  readln(x, y);
  min_element(x*y, sqr(x)*sqr(y));
  u:=min;
  min_element(2*u, 2*(x+y));
  v:=min;
  writeln(u, v)
end.

```

В описанной процедуре переменные `temp1` и `temp2` являются формальными параметрами. Значения этих параметров конкретизируются при вызове процедуры. Однако параметры-значения могут использоваться только при входе в процедуру, по окончании выполнения процедуры они прекращают свое существование и воспользоваться результатом вне процедуры невозможно.

Для описания данных, являющихся результатом, используются **параметры-переменные**. Перед параметрами-переменными в заголовке процедуры записывается служебное слово `var`. При таком способе подстановки можно получить результат в основную программу и использовать его в дальнейшем.

Оптимизируем рассматриваемую задачу, используя второй способ подстановки.

```

var
  x, y, u, v: real;

procedure min_element(temp1, temp2: real; var min: real);
begin
  if temp1>temp2
  then min:=temp1
  else min:=temp2
end;

begin
  readln(x, y);
  min_element(x*y, sqr(x)*sqr(y), u);
  min_element(2*u, 2*(x+y), v);
  writeln(u, v)
end.

```

Особенности организации процедур:

- количество формальных параметров в заголовке процедуры определяется пользователем; чем больше формальных параметров, тем гибче процедура, но использование большого количества параметров-значений замедляет работу программы;
- при использовании параметров-значений в памяти создаются копии этих значений, что замедляет работу при использовании параметров структурированного типа;
- при использовании параметров-переменных происходит ссылка на соответствующую область памяти;
- служебное слово `var` записывается перед каждым параметром-переменной;
- параметры-значения и параметры-переменные используются для передачи данных в процедуру, но параметры-переменные еще и возвращают данные в основную программу;
- тип каждого фактического параметра должен строго соответствовать типу формального параметра;
- фактический параметр, соответствующий параметру-значению, может быть представлен в виде переменной, константы или выражения, фактический параметр, соответствующий параметру-переменной, может быть только переменной;
- формальные параметры в заголовке подпрограммы перечисляются в любом порядке, но при вызове процедуры фактические параметры должны строго соответствовать заведенному порядку;
- в заголовке процедур для указания типов формальных параметров можно использовать только имена типов.

В Паскале допустимы только такие функции, значения которых относятся к простым типам. Правила организации функций совпадают с правилами организации процедур. При описании типа возвращаемого значения может использоваться только имя типа, но не его задание.

Рассмотрим использование функций на примере задачи, представленной ниже.

Задача 2.

Найти значение $f = (2k + 1)!/5! + 3k!$.

Решение.

Исходные данные для решения:

k — задаваемое значение.

Для решения задачи воспользуемся следующим алгоритмом:

1. Введем значение k .
2. Подсчитаем значение f . Для этого трижды обратимся к функции *factorial*, вычисляющей значение факториала.
3. Выведем результат на печать.

```
var
  k: byte;
  f: real;

function factorial(n: integer): real;
var
  p: real;
  i: integer;
begin
  p:=1;
  for i:=1 to n do
    p:=p*i;
  factorial:=p
end;

begin
  writeln('Введите значение k:');
  readln(k);
  f:=factorial(2*k+1)/factorial(5)+factorial(3*k);
  writeln('factorial = ', f:0:0);
end.
```

Особенности организации функций:

- тип возвращаемого функцией значения может быть только простым;
- тело функции может содержать любые операторы, но в нем обязательно должен быть оператор присваивания вида: <идентификатор функции>:=<выражение>;
- результат последнего по времени выполнения оператора такого вида и принимается в качестве значения функции;
- при описании типа возвращаемого функцией значения может использоваться только имя типа, но не его задание.

Задача 3.

Даны ежедневные температуры за три летних месяца.

Найдите, в каком месяце была минимальная среднемесячная температура.

Решение.

Анализ постановки задачи.

Исходными данными для решения задачи являются три массива температур:

temper1 — температуры за июнь (одномерный массив);

temper2 — температуры за июль (одномерный массив);

temper3 — температуры за август (одномерный массив).

Введем обозначения объектов программы:

i — переменная для организации цикла;

$sr1, sr2, sr3$ — переменные для хранения среднемесячной температуры;

s — сумма элементов массива.

Для решения задачи воспользуемся следующим алгоритмом:

1. Ввод элементов массива.
2. Нахождение среднемесячной температуры за каждый месяц.
3. Нахождение минимального значения.
4. Вывод результата на печать.

Текст программы:

```

type
  mass=array[1..31] of real;

var
  temper1, temper2, temper3: mass;
  sr1, sr2, sr3: real;

procedure vvod(var a: mass; n: byte);
var
  i: integer;
begin {ввод элементов массива}
  for i:=1 to n do
    readln(a[i]);
end;

function sum(var a: mass; n: byte): real;
var
  s: real;
  i: integer;
begin
  s:=0;           {обнуление счетчика}
  for i:=1 to n do {суммирование элементов массива}
    s:=s+a[i];
  sum:=s
end;

begin
  vvod(temp1, 30);
  vvod(temp2, 31);
  vvod(temp3, 31);
{нахождение среднего значения}
  sr1:=sum(temp1, 30)/30;
  sr2:=sum(temp2, 31)/31;
  sr3:=sum(temp3, 31)/31;
{нахождение минимального}
  if (sr1<sr2) and (sr1<sr3)
    then writeln('июнь')
    else if (sr2<sr1) and (sr2<sr3)
      then writeln('июль')
      else writeln('август')
end.

```

Задачи для самостоятельного решения.

1. Задано целое число N . Найти ближайшее к нему совершенное число. Поиск совершенного числа оформить в виде функции.

2. Определить количество различных цифр числа N . Использовать функцию для установления того, имеется ли цифра в числе.

3. Среди элементов целочисленного массива найти те числа, сумма квадратов цифр которых делится на 13. Использовать процедуру для нахождения суммы цифр в числе.

4. Вывести на печать сто простых чисел. Использовать функцию для установления того, является ли число простым.

5. Известны цены на 10 одинаковых товаров в каждом из 8 магазинов. Для каждого магазина напечатать список 3 наиболее дорогих товаров. Сортировку оформить в виде процедуры.

6. Задан текст. Напечатать слова из данного текста, которые являются симметричными. Определение симметричности слова оформить в виде процедуры.

Литература

1. *Аляев Ю. А., Козлов О. А.* Алгоритмизация и языки программирования Pascal, C++, Visual Basic: Учебно-справочное пособие. М.: Финансы и статистика, 2007.

2. *Ахо А. В. и др.* Структуры данных и алгоритмы: Пер. с англ.: Учеб. пособие. М.: Вильямс, 2000.

3. *Голицина О. Л.* Основы алгоритмизации и программирования : Учеб. пособие. 3-е изд., испр. и доп. М.: Форум, 2008.

4. *Зыков С. В.* Введение в теорию программирования. Курс лекций: Учеб. пособие. М.: ИНТУИТ.РУ «Интернет-Университет ИТ», 2004.

5. *Овчинникова И. Г., Сахнова Т. Н.* Задачник-практикум по структурному программированию на языке Паскаль: Учеб.-метод. пособие: 2-е изд., перераб. и доп. Магнитогорск: МаГУ, 2001.

6. *Сахнова Т. Н., Овчинникова И. Г.* Пособие для подготовки к ЕГЭ и ЦТ по информатике: Учеб. пособие: В 2 ч. Ч. 2. Базовый курс. Магнитогорск: МаГУ, 2006.

7. *Сахнова Т. Н., Овчинникова И. Г.* Пособие для подготовки к ЕГЭ и ЦТ по информатике: Учеб. пособие: В 2 ч. Ч. 1. Алгоритмизация и программирование. Магнитогорск: МаГУ, 2006.

8. *Хьюз Дж., Митчелл Дж.* Структурный подход к программированию: Пер. с англ. / Под ред. В. Ш. Кауфман. М.: Мир, 1980.

НОВОСТИ ЦИФРОВОГО МИРА

Google Go продолжает привлекать сторонников

Новый язык программирования Go, представленный Google в начале ноября 2009 г., продолжает привлекать сторонников, утверждающих, что он предоставляет целый ряд уникальных возможностей.

Напомним: объектно-ориентированный Go сочетает скорость разработки динамических языков (наподобие Python) с безопасностью компилирующихся языков вроде (Си и C++). При этом Go располагает неявной типизацией и реальными замыканиями (true closures), а возможности многопоточного программирования и параллелизма позволяют эффективно нагружать современные многоядерные процессоры. Есть мнение, что Go прекрасно подходит для системного программирования веб-серверов, систем хранения и баз данных. Внешне исходный код Go-приложений напоминает синтаксис C++ и Python.

Несмотря на то что Go — еще очень незрелый продукт, девелоперам импонирует мгновенная реакция Google на замечания и пожелания: корпорация прислушивается к мнению сообщества и быстро внедряет в Go необходимые исправления и улучшения. Кстати, особых критических ошибок и неожиданного поведения языка не замечено.

Несомненно, Go пока не подходит для создания критических бизнес-приложений, однако на нем вполне успешно реализуются прочие проекты, например сервер почтовых списков рассылки. Восторженные отзывы разработчиков касаются в основном упрощенной модели параллельного программирования, ведь в других отраслевых языках многопоточная и многопроцессорная обработка достигается с трудом. Параллельных приложений было бы куда больше, если бы Go появился раньше: на нем легко создавать, координировать и осуществлять взаимодействие потоков и процессов.

В целом в Google довольны высоким интересом к языку со стороны девелоперов.

(По материалам международного компьютерного еженедельника Computerworld Россия)



ИНФОРМАТИКА В НАЧАЛЬНОЙ ШКОЛЕ

Т. П. Быкова,

канд. пед. наук, доцент кафедры естественных наук и методики их преподавания
Борисоглебского государственного педагогического института

ФОРМИРОВАНИЕ У МЛАДШИХ ШКОЛЬНИКОВ НАВЫКА СТРУКТУРИРОВАНИЯ ИНФОРМАЦИИ С ПОМОЩЬЮ ТАБЛИЦ

Линия информации является одной из основных содержательных линий школьного курса информатики. Это справедливо и для программ по информатике для начальной школы. На пропедевтическом уровне младшие школьники знакомятся с видами и свойствами информации, формами представления информации, а также с информационными процессами.

Одним из основных информационных процессов является *обработка информации*. Все ситуации, связанные с обработкой информации, можно условно разделить на два типа.

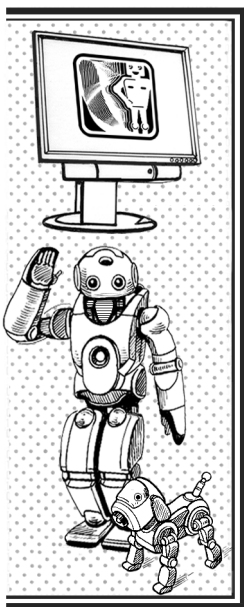
Первый тип — это *обработка, связанная с получением новой информации*. Примерами такой обработки информации являются решение математических задач, принятие решения на основе анализа ситуации и т. д.

Второй тип — это *обработка, связанная с изменением формы представления информации, не меняющая ее содержания*. Примерами обработки информации этого типа является перевод с одного языка на другой, преобразование информации из одной формы представления в другую. Формирование у учащегося навыка выполнения данного информационного процесса — важнейшая задача школы, так как умение преобразовывать информацию в форму, удобную для ее дальнейшего использования, — это неотъемлемая часть умения учиться.

Одним из важных видов обработки информации второго типа является *структурирование данных*. Этот вид преобразования информации связан с внесением определенного порядка в хранилище информации. Расположение в алфавитном порядке, группировка по некоторым признакам, использование табличного или графового представления — всё это примеры структурирования информации. С этим видом преобразования информации дети сталкиваются при изучении любого школьного предмета. Они выполняют структурирование информации, когда составляют план пересказа текста, оформляют краткую запись условия текстовой задачи и т. д. Однако при этом никак не говорится, что, выполняя подобные задания, дети совершают информационный процесс преобразования (в частности, структурирования) информации. Целенаправленно формировать умение структурировать информацию необходимо на уроках информатики. Одним из направлений этой работы может быть *обучение представлению текстовой информации в табличной форме*. Приведем примеры подобных заданий для IV класса.

Пример 1.

Расположи страны, о которых идет речь в тексте, в порядке убывания их площади.



«Знаешь ли ты, что наша страна — самая большая в мире? Ее площадь — 17 075 400 квадратных километров. Она располагается в двух частях света — Европе и Азии. Ее площадь почти в 31 раз превышает площадь одного из наиболее крупных государств Европы — Франции. Даже государство Австралия, занимающее практически весь континент Австралия, имеет площадь на 9 375 400 квадратных километров меньшую, чем Россия.

Одним из крупнейших государств Азии является Китай. На его территории может уместиться чуть больше 17 Франций.

Другим крупнейшим государством мира является Канада. Она расположена в Северной Америке и занимает площадь на 2 миллиона 270 тысяч квадратных километров большую, чем Австралия. Почти такую же площадь имеет государство Соединенные Штаты Америки. Его территория всего на 606 800 квадратных километров меньше территории Канады.

Крупнейшим государством Южной Америки является Бразилия. На ее территории может уместиться более 15 Франций.

В Африке нет государств с очень большой площадью. Крупнейшим из африканских государств является Алжир, площадь которого меньше площади Бразилии более чем в 3 раза, но менее чем в 4 раза».

В ходе обсуждения данного задания дети придут к выводу, что для его выполнения текстовое представление информации неудобно. С числовыми сведениями будет гораздо легче работать, если представить информацию в табличной форме.

Цель задания — расположить страны в порядке убывания их площади. Поэтому из всей информации, имеющейся в тексте, актуальными будут только сведения о площадях государств и названия этих государств. Следовательно, столбцов в таблице будет два. Количество строк таблицы равно количеству государств, о которых идет речь в тексте. Читаем текст еще раз и подчеркиваем все встречающиеся в нем названия стран. Вместе с нашей страной в тексте упоминаются еще семь государств. Поэтому строк в таблице будет 8. Еще нужно добавить одну строку, в которой будут записаны названия столбцов. Создаем такую таблицу и заполняем первый столбец:

Название государства	Площадь государства
Россия	
Франция	
Австралия	
Китай	
Канада	
США	
Бразилия	
Алжир	

Далее с помощью соответствующих арифметических действий находим площади указанных стран.

При выполнении задания важно обратить внимание детей на то, что не для всех стран могут быть получены точные числовые данные об их площади.

Так, например, площадь Франции равна:

$$17\,075\,400 : 31 = 550\,819 \text{ (ост. 11).}$$

Следовательно, площадь Франции больше 550 819 км², но меньше 550 820 км².

Площадь Китая примерно равна:

$$550\,820 \cdot 17 = 9\,363\,940 \text{ км}^2$$

(из значений 550 819 и 550 820 выбрано большее, так как сказано, что площадь Китая более чем в 17 раз больше площади Франции).

И т. д. Однако указанные погрешности не существенны по сравнению с численными значениями площадей государств. Поэтому, например, площадь территории Франции можно считать приблизительно равной 550 819 км², и т. д.

В таблицу информацию следует заносить в том порядке, в котором она предлагается в тексте, попутно выполняя необходимые вычисления. Когда второй столбец

таблицы будет заполнен численными значениями, станет легко расположить страны в порядке убывания их площади.

Данное задание — для работы в тетради. Его логическим продолжением будет предложить детям создать составленную таблицу на компьютере в ходе компьютерного практикума.

При выполнении задания осуществляется межпредметная связь с математикой: дети выполняют арифметические действия с многозначными числами и сравнение таких чисел.

В качестве необязательного домашнего задания можно предложить детям найти еще какие-нибудь сведения о странах, упоминавшихся в задаче, и составить небольшое устное сообщение об одной из них.

Задание способствует формированию информационных умений и навыков: работа с текстом, представление информации в табличной форме, упорядочение информации по числовым значениям.

Составление сообщения об одной из стран будет способствовать воспитанию любознательности, формированию познавательного интереса, формированию умения самостоятельно добывать информацию, работать с дополнительной литературой, воспитанию информационной потребности, формированию навыка самостоятельно создавать информационный объект, в частности устное сообщение.

Приведем еще один пример.

Пример 2.

«Вокруг самой близкой к нам звезды — Солнца — вращается 9 планет. Земля — это третья планета Солнечной системы. Расстояние от Солнца до Земли примерно 150 000 000 километров. Расстояние от Солнца до Нептуна почти в 30 раз больше, чем до Земли. Температура на Нептуне очень низкая, ведь он расположен так далеко от Солнца. Зато Венера почти в 45 раз ближе к Солнцу, чем Нептун. А Меркурий примерно на 40 миллионов километров ближе к Солнцу, чем Венера. Из-за близости к Солнцу поверхность Меркурия раскалена. Марс на 80 миллионов километров дальше от Солнца, чем Земля. Сатурн на 1 200 000 000 километров дальше Марса, а Юпитер на 650 000 000 километров ближе Сатурна. Сатурн окружен кольцами, состоящими из глыб и мелких частиц льда и пыли. Если расстояние от Нептуна до Солнца разделить на 2, а к результату прибавить 620 миллионов километров, то получим примерное расстояние до еще одной планеты Солнечной системы — Урана. Она ближе к Солнцу на 3 030 000 000 километров, чем Плутон — самая дальняя планета Солнечной системы.

Найди примерное расстояние от каждой планеты до Солнца и нарисуй модель Солнечной системы».

Важной функцией данного задания является формирование умения работать с информацией, представленной в виде достаточно объемного текста.

Учителю очень важно правильно организовать работу с этим заданием. Прежде всего необходимо, чтобы дети поняли, что информация, представленная в таком виде, трудна для обработки. Не все сведения, имеющиеся в тексте, актуальны для решения поставленной задачи. Поэтому прежде всего в тексте необходимо выделить главное. Так как в задании требуется найти расстояние от Солнца до каждой из планет, то именно эта информация и будет главной. Предложите ученикам подчеркнуть сведения, имеющие отношение к поставленной задаче. Далее работа организуется аналогично предыдущему заданию. Актуальная для поставленной задачи информация представляется в табличной форме, выполняются необходимые расчеты, и полученные сведения упорядочиваются по возрастанию расстояния от каждой планеты до Солнца.

Выполнение задания позволяет формировать такие важные компоненты информационной культуры современного человека, как умение работать с текстом, оценивать информацию с точки зрения ее актуальности и полезности для решения поставленной задачи, выделять в имеющейся информации главное, представлять информацию в табличной форме, упорядочивать ее по заданному принципу.

После работы над заданием в классе учитель может предложить детям в качестве **необязательного домашнего задания** найти другие сведения о планетах Солнечной системы и составить сообщение об одной из планет.

Подобные задания помимо основной цели — формирования умения структурировать информацию и представлять текстовую информацию в табличной форме — способствуют развитию любознательности учащихся, формированию интереса к учебной деятельности, процессу получения новых знаний. Они также помогают формировать умение работать с дополнительной литературой, искать информацию в энциклопедиях, справочниках, создавать собственные информационные объекты, в частности устные и письменные сообщения, а также способствуют развитию речи учащихся.

НОВОСТИ ЦИФРОВОГО МИРА

Работаем со шрифтом — программы-менеджеры

Несмотря на то что многие приложения для работы с изображениями и видео, да и просто текстовые редакторы отображают название шрифта, используя его начертание, в отдельных случаях бывают весьма полезны программы, которые позволяют не только просматривать шрифты, но и выполнять с ними определенные действия. Среди них есть как бесплатные, так и коммерческие продукты. Хорошим примером свободно распространяемой и при этом весьма мощной программы является Font Xplorer (www.moonsoftware.com/xfplorer.asp). Она довольно старая, ее последний релиз вышел в 2001 г. Однако данная утилита отлично справляется со своими обязанностями во всех версиях операционной системы Windows, включая Windows XP. А вот для владельцев компьютеров, на которых установлена Windows Vista, она уже не подходит, и это очень серьезный ее недостаток.

В остальном же этот продукт весьма удачен и очень удобен. Даже английский язык интерфейса практически не осложняет его использование. Главное окно программы содержит список шрифтов активной папки. Здесь могут находиться шрифты как из стандартной папки Windows, установленные в системе, так и из любого другого источника. В каждой строке приводится название шрифта, а также отображается определенный пользователем текст. Это позволяет быстро отыскивать шрифты с нужным начертанием. Кроме того, можно просмотреть таблицу символов, полные метрические свойства шрифта и другую информацию.

Имеющиеся в Font Xplorer дополнительные возможности делают более удобной работу с этой программой. В частности, предусмотрены такие функции, как фильтрация списка по набору или типу шрифтов (символьные, балтийские, кириллические, декоративные, романские, модерн и т. д.), а также создание избранных папок, обеспечивающих быстрый доступ к не установленным в системе шрифтам.

В этой программе реализованы интересные функции управления, например, предварительная загрузка шрифтов, их установка и удаление из системы. А еще Font Xplorer поможет быстро найти все дубликаты шрифтов, а также шрифты, размещенные на жестком диске, но не установленные в Windows.

Также можно осуществить автоматическое присвоение файлам шрифтов названий, совпадающих с их полными именами. Иногда это бывает очень удобно. Впрочем, и этим возможности программы не ограничиваются: она выполняет поиск и удаление ссылок в системном реестре Windows на несуществующие шрифты, в том числе и находящиеся на съемных дисках. Это очень полезно, поскольку наличие таких записей может привести к замедлению работы и даже сбоям операционной системы. Кстати, порадуют пользователей печать выбранного шрифта на бумаге, сохранение его в виде графического файла формата BMP и проч.

(Обзор подготовлен по материалам журнала «Мир ПК»)

Т. В. Баракина,

канд. пед. наук, доцент Омского государственного педагогического университета,

С. В. Поморцева,

канд. пед. наук, доцент Омского государственного педагогического университета

ИЗУЧЕНИЕ ЭЛЕМЕНТОВ ЛОГИКИ И ТЕОРИИ МНОЖЕСТВ В НАЧАЛЬНОМ КУРСЕ ИНФОРМАТИКИ*

3. Высказывания и операции над ними в обучении информатике в начальной школе

На уроках информатики в начальной школе по любой действующей программе необходимо:

- познакомить младших школьников с понятием высказывания (на уровне представления);
- научить отличать высказывания от других предложений;
- научить оценивать истинность высказываний;
- научить строить составные высказывания с помощью логических связок «и», «или», «не» и определять их истинность.

3.1. Изучение высказываний в курсе информатики Е. П. Бененсон, А. Г. Паутовой

3.1.1. II класс

Понятие высказывания вводится в рамках темы «Устройство компьютера», причем с истинными и ложными высказываниями дети знакомятся не одновременно.

Сначала учащимся дается задание отметить предложения, в которых высказана истина, буквой И [1, с. 39].

Задача 1.

- 7 — двузначное число.
- Солнце вращается вокруг Земли.
- Телевизор используется для хранения информации.
- Компьютер предназначен для обработки информации.
- Телефон нужен для передачи информации.

Затем сообщается, что «предложение, в котором высказана истина, назовем **истинным высказыванием**» [1, с. 39].

Далее предлагаются абсолютно одинаковые рисунки, но с разными подписями: «Осень в лесу» и «Зима в лесу». Требуется раскрасить рисунки так, чтобы подписи под ними были истинными.

Упражнения на распознавание истинных высказываний постепенно усложняются посредством включения условий «если ..., то».

Задача 2.

Отметь буквой И истинные высказывания.

- Если число записано двумя цифрами, то оно двузначное.
- Если число двузначное, то оно записано двумя цифрами.
- Если Вася старше Пети, а Петя старше Коли, то Коля моложе Васи.
- Если Даша учится во втором классе, то Даша — школьница.
- Если Витя школьник, то он учится во втором классе [1, с. 43].

* Продолжение. Начало см.: Информатика и образование. 2010. № 1.

Ложные высказывания вводятся в процессе выполнения упражнения на их распознавание среди высказываний со словами-кванторами («все», «каждый», «некоторые», «ни один»).

Задача 3.

Отметь буквой И предложения, в которых высказана истина, а буквой Л — ложь.

- Все птицы хорошо летают.
- Все деревянные дома ниже, чем каменные.
- Некоторые дома — одноэтажные.
- Каждый луговой цветок — белый.
- Ни один луговой цветок не может быть белым.
- Некоторые луговые цветы — желтые [1, с. 46].

Затем обобщается: «Предложение, в котором высказана ложь, назовем **ЛОЖНЫМ высказыванием**» [1, с. 46].

В дальнейшем упражнения на распознавание истинных и ложных высказываний помогают проверить усвоение учащимися ранее изученного материала, например устройства компьютера.

Задача 4.

Истинные высказывания отметь буквой И, ложные — Л.

- Информатика изучает числа и фигуры.
- Информатика изучает только устройство компьютера.
- Человек получает информацию благодаря органам чувств.
- Компьютер получает информацию благодаря устройствам ввода.
- При выключении компьютера информация на жестком диске пропадает.
- Монитор — устройство ввода.
- При выключении компьютера информация в оперативной памяти пропадает.
- Разных людей интересует разная информация [1, с. 68].

Дети используют понятия истинного и ложного высказываний при выполнении упражнений на кодирование информации, в процессе решения логических задач.

Задача 5.

Каждую букву своего имени мальчик заменил порядковым номером этой буквы в алфавите. Имей в виду, что буква «а» имеет номер 1. Буквы с номером 0 нет. Получился код: 510141.

а) Как звали мальчика?

510141 ⇒ _____

б) Девочку зовут Светлана. Зашифруй ее имя тем же способом.

Светлана ⇒ _____

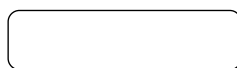
в) Подчеркни истинное высказывание:

Использованный шифр является шифром перестановки.

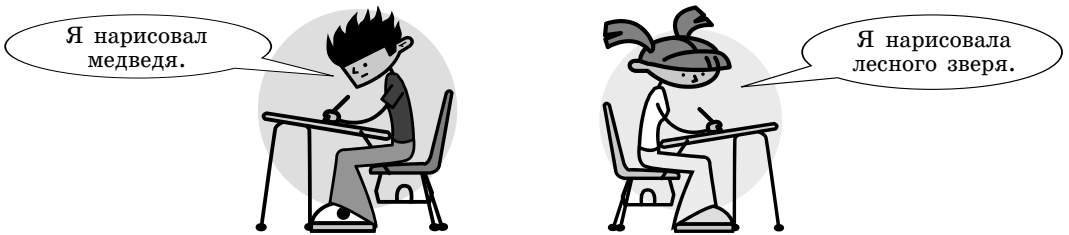
Использованный шифр является шифром замены [1, с. 70].

Задача 6.

Миша и Маша нарисовали двух зверей. Вот они:



Когда их спросили, кто какого зверя нарисовал, они ответили:



Одно высказывание было истинным, а другое — ложным. Подпиши под рисунками имена авторов [1, с. 70].

Решение.

Предположим, что высказывание Миши истинное, т. е. он действительно нарисовал медведя. Но тогда высказывание Маши тоже истинное, так как нарисованная ею белка — лесной зверь. Это противоречит условию о том, что одно из высказываний истинное, а другое — ложное.

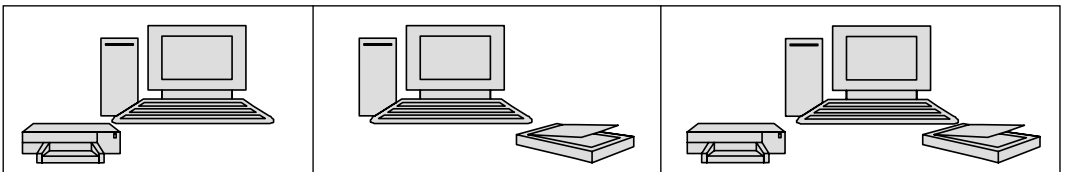
Если высказывание Миши ложное, то в действительности он нарисовал не медведя, а белку. При этом получается, что Маша нарисовала медведя. Это лесной зверь. Поэтому высказывание девочки истинное.

Ответ. Миша нарисовал белку, а Маша — медведя.

Для подготовки к введению понятия **логических связей** используются следующие упражнения.

Задача 7. Назначение логических связей «и», «или».

Ире важно, чтобы у компьютера был принтер, Вере — сканер. Лене нужен компьютер, который имеет принтер **и** сканер. Сою устраивает любой компьютер с принтером **или** сканером. Подпиши имя каждой девочки под теми компьютерами, которые ей подходят [2, с. 13].



Задача 8. Построение отрицания.

а) Петя и Света — близнецы. Они родились 8 января. Вот они со своей семьей:



Каждому, кто моложе Пети, дай шарик (нарисуй).

Каждому, кто старше Пети, дай цветок (нарисуй).

Кто такого же возраста, как Петя? Раскрась его одежду.

Обведи красной рамкой каждого, кто **не** моложе Пети.

Обведи зеленой рамкой каждого, кто **не** старше Пети [2, с. 24].

б) Заполни пропуски в высказываниях так, чтобы высказывания были истинными.

Шкаф выше, чем _____.

Первоклассник ниже, чем _____.

Катя ниже Пети, а Маша одного с ним роста. Катя и Маша **не** _____, чем Петя.

Рекс выше, чем Бим, а Мухтар одного роста с Бимом. Рекс и Мухтар **не** _____, чем Бим [2, с. 25].

Во II классе учащиеся закрепляют умение распознавать истинные и ложные высказывания, опираясь на свои знания по математике.

Равенства и неравенства — это тоже высказывания. Они записаны не на русском языке, а на языке математики.

Задача 9.

Отметь буквой И истинные высказывания, буквой Л — ложные [2, с. 36].

$18 > 81$

$30 + 18 < 19$

$20 + 40 = 60$

$37 < 73$

$70 - 5 = 80 - 5$

$5 \times 3 = 5 + 3$

Задача 10.

Поставь знаки $<$, $>$ или $=$ так, чтобы получилось истинное высказывание [2, с. 40].

$17 \square 12$

$21 \square 21$

$48 \square 50$

$10 \square 0$

$18 - 18 \square 8$

$47 + 3 \square 40$

$70 - 70 \square 0$

$0 + 0 \square 0$

В процессе обучения основам алгоритмизации учащиеся также оперируют термином «истинное высказывание».

Игра «Перемещайка».

На поле 4 знака: 3 цифры (2, 8, 9) и знак «больше» ($>$). Придумай алгоритмы.

Цель: получить истинное высказывание. Внутри неравенства не должно быть пустых клеток.

Исходное положение:

9	8	2	>	
---	---	---	---	--

Конкурс: кто составит больше истинных высказываний [2, с. 55].

3.1.2. III класс

Как уже отмечалось, во II классе осуществляется подготовка к введению логических связок «и», «или». В III классе изучается тема «**Простые и сложные высказывания**» [4, с. 26].

Детям сообщается, что высказывания бывают простыми и сложными. Определить истинность простых высказываний несложно.

Например, легко оценить истинность высказываний:

$10 > 5$

Москва — столица России.

$3 + 2 < 10$.

Если два простых высказывания соединить с помощью действия логического сложения или логического умножения, то получится одно **сложное высказывание**.

Сложное высказывание, полученное с помощью **логического умножения**, истинно, если все простые высказывания, из которых оно состоит, истинны.

Логическое умножение будем обозначать буквой И.

Задача 11.

Прочти сложное высказывание $0 < 9$ И $0 > 1$.

Из каких простых высказываний оно состоит? Оцени их истинность и истинность сложного высказывания.

Учащихся знакомят с таблицей истинности логического умножения.

Таблица истинности логического умножения

Высказывание 1	Высказывание 2	Высказывание 1 И Высказывание 2
Истина	Истина	Истина
Истина	Ложь	Ложь
Ложь	Истина	Ложь
Ложь	Ложь	Ложь

Дети определяют истинность сложных высказываний:

$$10 > 5 \text{ И } 10 = 3.$$

Москва — столица России **И** В Москве есть Кремль.

Буратино — герой сказки «Колобок» **И** Буратино сделан из глины.

Сложное высказывание, полученное с помощью **логического сложения**, истинно, если хотя бы одно из простых высказываний, входящих в него, истинно.

Логическое умножение будем обозначать словом **ИЛИ**.

Предлагается оценить истинность сложного высказывания:

$$0 < 9 \text{ ИЛИ } 0 > 1$$

Это проще сделать, воспользовавшись таблицей истинности логического сложения.

Таблица истинности логического сложения

Высказывание 1	Высказывание 2	Высказывание 1 ИЛИ Высказывание 2
Истина	Истина	Истина
Истина	Ложь	Истина
Ложь	Истина	Истина
Ложь	Ложь	Ложь

Далее учащиеся закрепляют умение определять истинность сложных высказываний с помощью упражнений следующих видов:

- 1) определение истинности сложных высказываний по рисунку;
- 2) исполнение алгоритмов, содержащих условия в форме сложных высказываний, по их блок-схемам;
- 3) оценка истинности сложного высказывания с переменной, если задано ее конкретное значение.

Задача 12.

Оцени истинность высказывания при $D = 70$ [4, с. 89].

$$D > 40 \text{ И } D < 70$$

$$D > 40 \text{ ИЛИ } D < 70$$

$$D > 40 \text{ И } D < 90$$

$$D > 40 \text{ ИЛИ } D < 90$$

3.1.3. IV класс

В IV классе дети закрепляют свои умения строить сложные высказывания и оценивать их истинность при изучении последующих разделов курса информатики.

3.2. Тема «Логические рассуждения»

в курсе «Информатика в играх и задачах» А. В. Горячева

Эта тема входит в содержание всех лет обучения в начальной школе, так как содержание и объем рассматриваемых логических понятий возрастают от класса к классу.

3.2.1. I и II классы

В I и II классах дети учатся распознавать заведомо ложное высказывание, подбирать противоположное по смыслу слово. Учащиеся должны отличать высказывания от других предложений, приводить примеры высказываний, по рисунку определять истинность или ложность высказываний.

Учащимся предлагаются задания:

- на сравнение предметов с противоположными свойствами («высокий — низкий», «большой — маленький» и т. д.);
- на рисование предметов с противоположными свойствами;
- на попарное соединение слов с противоположными значениями.

Например, для выделенного свойства одного предмета ученики должны подобрать противоположное свойство другого предмета:

- жираф высокий — барашек низкий;
- мяч большой — воланчик маленький;
- лента узкая — другая лента широкая;
- одна дорога прямая — другая кривая (или извилистая).

Смысл понятий «истина» и «ложь» в I и II классах объясняют на примерах, которые знакомы детям из их личного опыта. Это может быть картинка и подпись под ней. Ребенок знает, что если изображение и подпись совпадают, то это истина, если нет, то ложь. Наряду с заданиями, в которых нужно определить истину, есть задания, в которых требуется исправить изображение или подпись, чтобы они стали истинными (рис. 1):



Рис. 1 [10]

Некоторые упражнения включают в себя условие «если», которое усложняет задание, но в то же время делает его более интересным (рис. 2).

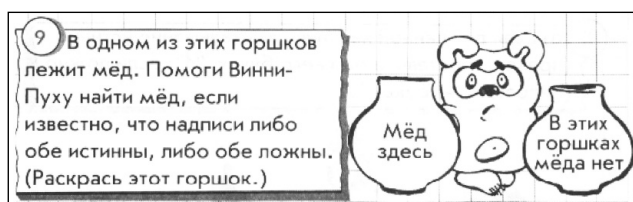


Рис. 2 [10]

Во II классе вводится понятие **высказывания**.

Учитель подчеркивает, что математика тоже имеет дело с предложениями. Но только в математике мы интересуемся, правдиво ли это предложение (т. е. верно, истинно) или неправдиво (т. е. неверно, ложно). И такое предложение в математике называется высказыванием. То есть мы не просто говорим какое-то предложение, а высказываем правду или ложь.

Итак, высказывание бывает истинным (правда) или ложным (неправда).

При построении отрицаний высказываний особое внимание уделяется употреблению слов, взаимно отрицающих друг друга: «всегда», «иногда», «никогда», «все», «некоторые», «никто». Учитель должен научить ученика употреблять эти слова при отрицании текста.

Необходимо помочь детям образовывать правильное отрицание высказывания. Для этого достаточно подчеркнуть, что отрицание высказывания — это полная противоположность самого высказывания («совсем наоборот»).

Для самоконтроля рекомендуется отрицание высказывания начинать со слова «нет» («иногда идет дождь» — «нет, дождь никогда не идет»).

После освоения детьми понятий «отрицание», «истина», «ложь» вводятся связи «и», «или», «не» и рассматривается их связь с операциями над множествами. Необходимо, чтобы дети понимали, что у пересекающихся множеств есть области, соответствующие данным логическим союзам.

Если употребляется союз «и», то речь идет об области пересечения.

Если употребляется союз «или», то речь идет об элементах всех рассматриваемых областей.

Если употребляется частица «не», то речь идет об элементах, находящихся вне рассматриваемой области.

Задача 13 (рис. 3).

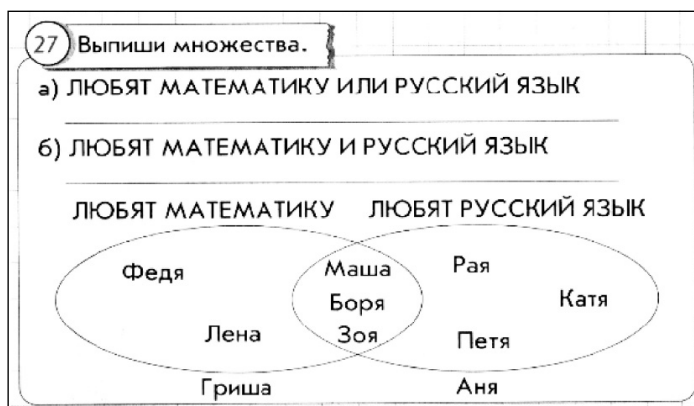


Рис. 3 [12]

3.2.2. III класс

В III классе осуществляется повторение и закрепление у учащихся представлений об истинности высказываний, в том числе со связками «и», «или», «не».

Учитель обсуждает с детьми, что бывают вопросы, на которые можно ответить «да» или «нет», и вопросы, на которые нельзя так ответить. Просит нескольких человек привести примеры вопросов на «да — нет». Проводит фронтальный опрос: каждый ребенок должен придумать свой вопрос.

Затем учитель предлагает нескольким детям привести примеры вопросов, на которые нельзя ответить «да» или «нет». Как правило, такие вопросы начинаются с вопросительных слов либо предполагают выбор из перечисленного.

Например:

- Сколько весит слон?
- Это город Москва или Петербург?

Сначала надо дать время детям самостоятельно придумать три вопроса, на которые можно ответить «да» или «нет», и три вопроса, на которые нельзя так ответить. А через некоторое время следует проверить задание вслух.

Примеры вопросов, на которые можно ответить «да» или «нет»:

- Знайка сидит в машине?
- Самоделкин держит кисточку?
- Незнайка держит кисточку?

Примеры вопросов, на которые нельзя ответить «да» или «нет»:

- Что в руках у Чиполлино?
- Где лежит молоток?
- Откуда выглядывает Знайка?

При определении истинности высказываний со связками «не», «и», «или» вместо термина «связка» используется термин «слово».

В III классе задания на эту тему сгруппированы попарно.

Первое задание представляет собой рисунок и таблицу, в которой приведены высказывания — простые или сложные. Ученик должен решить, истинное высказывание или ложное, и проставить слова «да» или «нет» (рис. 4).

19 Рассмотрите рисунки и впишите в таблицу слова «ДА» и «НЕТ».

Номера рисунков:	1	2	3	4
Рубашка белая.				
У рубашки больше пяти пуговиц.				
У рубашки меньше пяти пуговиц.				
У рубашки есть карманы.				
У рубашки все пуговицы застёгнуты.				

Рис. 4 [14]

Следующее задание связано с предыдущим. Чтобы справиться с ним, ребенок должен проанализировать результаты выполнения предыдущего задания (рис. 5).

20 Впишите в фигуры номера рисунков.

Множества:	
	– рубашек на рисунках
	– белых рубашек
	– рубашек с карманами

Рис. 5 [14]

Особую сложность для учащихся представляют задания, где в простых высказываниях используется слово «не» (рис. 6).

21 Рассмотрите рисунки и впишите в таблицу слова «ДА» и «НЕТ». Найдите рисунок, для которого все высказывания истинны. Обведите цветным карандашом его рамку.

Номера рисунков:	1	2	3	4
На рисунке НЕ белая скатерть.				
На рисунке НЕТ вазы с цветами.				
На рисунке НЕ больше трёх свечек.				
На рисунке НЕ все свечки горят.				

Рис. 6 [14]

Далее предлагаются задания со сложными высказываниями, в которых употребляются слова «и», «или» (рис. 7, 8).

23 Рассмотрите рисунки и впишите в таблицу слова «ДА» и «НЕТ».

Номера рисунков:	1	2	3	4
НЕ нарисована кошка.				
Нарисована кошка И собака.				
Нарисован зверь (кошка ИЛИ собака).				
НЕ нарисован зверь				

Рис. 7 [14]

24 Впиши в фигуры номера рисунков. Какими должны быть рисунки 5 и 6? (Опиши их словами.)
Закрась жёлтым цветом область множества рисунков, на которых есть кошка **И** собака.

Множества:	
<input type="checkbox"/>	– рисунков
<input type="radio"/>	– рисунков с кошкой
<input type="checkbox"/>	– рисунков с собакой
<input type="radio"/>	– рисунков с облаками

Рис. 8 [14]

При закреплении умения строить отрицания простых высказываний особое внимание следует уделить использованию кванторов.

Учитель произносит высказывания с кванторами:

- всегда, везде, все, всякий;
- никогда, нигде, никто, никакой;
- иногда, кое-где, кое-кто, некоторый,

а ученики должны сформулировать отрицания высказываний учителя.

Необходимо обратить внимание детей, что в данном случае речь не идет о правильности высказываний, нужно назвать высказывание (фразу), опровергающее высказывание (фразу) учителя вне зависимости от того, правильно это или нет.

А. В. Горячев рекомендует рассмотреть следующую таблицу:

Высказывание	Отрицание
Винтик и Шпунтик сделали автомобиль.	Винтик и Шпунтик не сделали автомобиль.
Все дети любят хорошие отметки.	Не все дети любят хорошие отметки. (Некоторые дети любят плохие отметки.)
Иногда во время дождя бывает радуга.	Во время дождя никогда не бывает радуги.
Пончик и Сиропчик любят сладкое.	Пончик и Сиропчик не любят сладкое.
Все дети любят мультики.	Не все дети любят мультики.
Не все птицы улетают зимой на юг. (Некоторые птицы не улетают зимой на юг.)	Все птицы улетают зимой на юг.

3.2.3. IV класс

В IV классе учащиеся впервые рассматривают **правило построения выводов «если ..., то ...»**.

Цель данной темы — научить составлять схемы таких правил, определять ситуации, в которых можно (нельзя) сделать вывод с помощью правила «если ..., то ...». Под термином «правило» понимается инструкция к определению истинности высказывания для каждого конкретного случая.

Правило «если ..., то ...» состоит из двух высказываний: условия и заключения. Стрелка в краткой записи этого правила идет от условия к заключению. Иногда правило «если ..., то ...» действует и в обратном направлении.

Например:

- а) «Если птица — страус, то она не умеет летать».

Правило будет записано так:

СТРАУС → НЕ ЛЕТАЕТ

Это правило работает только в одну сторону, мы не можем сказать: «Если птица не летает, то эта птица — страус».

- б) «Если буква стоит первой в алфавите, то это буква А».

Правило будет записано так:

ПЕРВАЯ БУКВА АЛФАВИТА ↔ БУКВА «А»

Дети рассматривают примеры на это правило из математики или других предметов.

ЧЕТНОЕ ЧИСЛО ↔ ДЕЛИТСЯ НА 2
КВАДРАТ → СТОРОНЫ РАВНЫ

Вопросы. Как вы думаете, работает ли последнее правило в обратном направлении? Какое высказывание надо добавить и каким союзом его соединить, чтобы это правило работало и в обратном направлении?

Условие правила «если ..., то ...» может состоять не только из одного высказывания, но и из двух и более, которые должны быть соединены союзами «и» либо «или».

Игра «Назови условие».

Учитель произносит правило, пропуская его условие, и просит детей назвать пропущенное условие, например:

- Если зверь ..., то это бобр.
- Если предмет ..., то это карандаш.
- Если человек ..., то это клоун.

В условии правила учащиеся должны назвать одно или несколько свойств объекта (признаки, составные части, действия). Поэтому условие может быть простым («умеет строить плотины») или сложным («в нем есть грифель и им можно рисовать»). Ученики формулируют такое условие, чтобы в нем объект однозначно определялся набором представленных свойств.

В результате выполнения этих заданий дети должны сделать следующие выводы:

- Правило «если ..., то ...» будет истинным, если истинны все высказывания, соединенные союзом «и», которые входят в условие этого правила.
- Если высказывания, входящие в условие, соединены союзом «или», то правило «если ..., то ...» истинно при истинности хотя бы одного из этих высказываний.

Цель изучения темы «**Схема рассуждений**» — научить детей составлять схемы рассуждений из правил «если ..., то ...» и делать выводы с их помощью.

Схема рассуждений отвечает на вопрос: почему из факта можно сделать вывод? Для этого в нее включаются все правила, которые помогают сделать правильный вывод.

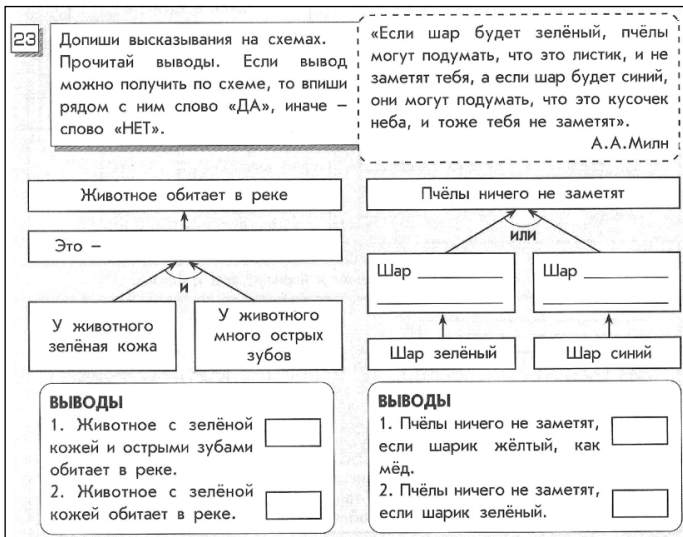


Рис. 9 [16]

Схему рассуждений можно составить из нескольких правил, для этого заключение одного правила должно быть условием другого. Схему рассуждений принято составлять снизу вверх. Важно, что правильный вывод можно сделать по схеме рассуждений только тогда, когда в схему включены соответствующие правила (рис. 9).

Игра «Составьте цепочку правил».

Учитель читает отрывок из сказки или стихотворения С. Маршак.

Дуйте, дуйте, ветры в поле!
Чтобы мельницы мололи,
Чтобы завтра из муки
Испекли нам пирожки!

Дети должны сформулировать правило «если ..., то ...» по тексту этого стихотворения:

Если дует ветер, то мельницы мелют.
Если мельницы мелют, то есть мука.
Если есть мука, то завтра будут пирожки.

Далее учащиеся должны построить цепочку рассуждений (рис. 10).

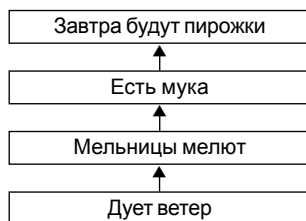


Рис. 10

Затем учитель предлагает учащимся несколько выводов, а они должны определить, можно ли сделать такие выводы по полученной схеме рассуждений. Например:

- В ветреный день есть мука.
- В ветреный день есть пирожки.
- Если мельницы не работают, то пирожков нет.

Таким образом, тема «Логические рассуждения» изучается по программе А. В. Горячева «Информатика в играх и задачах» концентрически на протяжении всех лет обучения информатике в начальной школе.

Учащиеся выясняют смысл понятий «истина» и «ложь»; получают представление о понятии высказывания; учатся определять его истинность, распознавать высказывания среди множества предложений, приводить примеры высказываний, строить сложные высказывания со словами «и», «или», «не» и определять их истинность, устанавливать связи между этими словами и операциями над множествами, строить простейшие логические цепочки рассуждений по правилу «если ..., то ...».

Учебники А. В. Горячева содержат достаточно полную, разнообразную, хорошо иллюстрированную систему практических упражнений по теме «Логические рассуждения», позволяющих акцентировать внимание на всех необходимых аспектах изучаемых понятий.

3.3. Примеры упражнений, которые можно использовать

при изучении высказываний и операций над ними в начальной школе
Задание 1.

Под **высказыванием** понимают всякое повествовательное предложение, относительно которого имеет смысл говорить, истинно оно или ложно. Вопросительные и восклицательные предложения, а также предложения, выражающие приказание или сожаление, не являются высказываниями, так как не имеет смысл говорить, истинны они или ложны.

Среди перечисленных предложений укажи те, которые являются высказываниями.

- Подойди ко мне!
- Ты вчера был в школе?
- Сумма чисел 2 и 3 равна 5.
- Земля вращается вокруг Луны.
- Сегодня по телевизору хорошая программа.
- Цыплят по осени считают.
- 3 — число нечетное.

Ответ. Высказываниями являются следующие повествовательные предложения:

- «Сумма чисел 2 и 3 равна 5»,
- «Земля вращается вокруг Луны»,
- «3 — число нечетное».

Задание 2.

Высказывания могут быть **истинными** и **ложными**.

Определи истинность следующих высказываний:

- A: «Москва — столица России».
- B: «Лед — твердое состояние воды».
- C: «Солнце — спутник Земли».
- D: «Кошки умеют летать».
- E: «10 — число однозначное».

Ответ. Истинные высказывания — A, B; ложные высказывания — C, D, E.

Задание 3.

В предложение вставь частицу «не» так, чтобы его смысл изменился на противоположный.

Кошки любят есть сено. → Кошки **не** любят есть сено.

- Лошади летают по небу.
- Дети должны играть спичками.
- 2 — число четное.
- Море впадает в реку.
- Вашингтон — столица Украины.

Ответ.

- Лошади **не** летают по небу.
- Дети **не** должны играть спичками.
- Море **не** впадает в реку.
- Вашингтон — **не** столица Украины.

Задание 4.

Высказывания, в которых нет слов-связок («не», «и», «или», «если ..., то ...»), называются **элементарными**. Высказывания, полученные из элементарных с помощью слов-связок, называют **составными**.

Из двух элементарных высказываний построй составное, используя связки «и», «или».

Марина старше Светы. → Марина и Оля старше Светы.
Оля старше Светы.

- Квадрат — геометрическая фигура. Круг — геометрическая фигура.
- Синий кубик меньше красного. Синий кубик меньше зеленого.
- Дельфин — рыба. Дельфин — млекопитающее.
- Луна — планета. Луна — спутник Земли.
- Буква «а» — первая буква в слове «аист». Буква «т» — первая буква в слове «аист».

Ответ.

Квадрат и круг — геометрические фигуры.

Синий кубик меньше красного и зеленого.

Дельфин — рыба или млекопитающее.

Луна — планета и спутник Земли.

В слове «аист» первая буква «а» или «т».

Задание 5.

Лена сказала, что пойдет в кино, если пойдут Оля и Женя.

- Оля идет в кино, а Женя нет. Пойдет ли Лена?
- Ни Оля и ни Женя не пойдут в кино. Пойдет ли Лена?

Ответ. Лена не пойдет в кино ни в первом, ни во втором случае.

Задание 6.

Для урока рисования ученики должны были принести краски или карандаши.

- Готов ли Саша к уроку, если он принес только краски?
- Готов ли Женя к уроку, если он не принес ни красок, ни карандашей?
- Готова ли Оля к уроку, если она принесла краски и карандаши?

Ответ. Саша и Оля к уроку готовы, а Женя — нет.

Задание 7.

Назови число, которое ...

- а) больше 3 и больше 5;
- б) больше 3 или больше 5;
- в) больше 3 и меньше 5;
- г) меньше 3 и больше 5;
- д) не меньше 3.

Ответ.

- а — все числа, большие 5;
- б — все числа, большие 3;
- в — 4;
- г — такого числа нет;
- д — числа, большие или равные 3.

Задание 8.

Таня и Ира играли в шашки. Как закончилась игра, если известно, что ...

- а) Таня не выиграла и Ира не проиграла?
- б) Таня не выиграла или Ира не проиграла?
- в) проиграла не Ира?
- г) выиграла не Таня?

Ответ.

- а — ничья;
- б — ничья;
- в — проиграла Таня;
- г — проиграла Таня.

Задание 9.

Записаны числа 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Какие из приведенных ниже предложений относительно этих чисел будут верными?

- а) Все числа имеют по два соседних числа.
- б) Существует число, имеющее только одно соседнее число.
- в) Любое из чисел — однозначное.
- г) Каждое число больше предыдущего на 1.

Ответ. Верными будут предложения б, г.

Продолжение следует

Литература

1. *Бененсон Е. П., Паутова А. Г.* Информатика: Учебник для 2 класса (Первый год обучения): В 2 ч. Ч. 1. М.: Академкнига/Учебник, 2002.
2. *Бененсон Е. П., Паутова А. Г.* Информатика: Учебник для 2 класса (Первый год обучения): В 2 ч. Ч. 2. М.: Академкнига/Учебник, 2002.
3. *Бененсон Е. П., Паутова А. Г.* Информатика: 3 класс: Учеб.-тетрадь: В 2 ч. Ч. 1. 2-е изд., испр. и доп. М.: Академкнига/Учебник, 2005.
4. *Бененсон Е. П., Паутова А. Г.* Информатика: 3 класс: Учеб.-тетрадь: В 2 ч. Ч. 2. 2-е изд., испр. и доп. М.: Академкнига/Учебник, 2005.
5. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 1 класс: Методические рекомендации для учителя. М.: Баласс, 2006.
6. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 2 класс: Методические рекомендации для учителя. М.: Баласс, 2007.
7. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 3 класс: Методические рекомендации для учителя. М.: Баласс, 2006.
8. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 4 класс: Методические рекомендации для учителя. М.: Баласс, 2004.
9. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 1-й класс: Учеб.-тетрадь: В 2 ч. Ч. 1. М.: Баласс, 2005.
10. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 1-й класс: Учеб.-тетрадь: В 2 ч. Ч. 2. М.: Баласс, 2004.
11. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 2-й класс: Учеб.-тетрадь: В 2 ч. Ч. 1. М.: Баласс, 2003.
12. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 2-й класс: Учеб.-тетрадь: В 2 ч. Ч. 2. М.: Баласс, 2006.
13. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 3-й класс: Учеб.-тетрадь: В 2 ч. Ч. 1. М.: Баласс, 2008.
14. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 3-й класс: Учеб.-тетрадь: В 2 ч. Ч. 2. М.: Баласс, 2008.
15. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 4-й класс: Учеб.-тетрадь: В 2 ч. Ч. 1. М.: Баласс, 2008.
16. *Горячев А. В., Горина К. И., Суворова Н. И.* Информатика в играх и задачах. 4-й класс: Учеб.-тетрадь: В 2 ч. Ч. 2. М.: Баласс, 2008.
17. *Залогова Л. А., Плаксин М. А., Русаков С. В. и др.* Информатика: Задачник-практикум: В 2 т. / Под ред. И. Г. Семакина, Е. К. Хеннера: Т. 1. 2-е изд. М.: БИНОМ. Лаборатория знаний, 2005.
18. *Керова Г. В.* Нестандартные задачи по математике. 1—4 классы. М.: ВАКО, 2006.
19. *Матвеева Н. В., Челак Е. Н., Конопцова Н. К., Панкратова Л. П., Нурова Н. А.* Информатика: Учебник для четвертого класса. М.: БИНОМ. Лаборатория знаний, 2006.
20. *Матвеева Н. В., Челак Е. Н., Конопцова Н. К., Панкратова Л. П., Нурова Н. А.* Информатика и ИКТ: Рабочая тетрадь для 4 класса: В 2 ч. М.: БИНОМ. Лаборатория знаний, 2008.
21. *Солнышко С. В.* Использование комбинаторных задач при обучении первоклассников математике // Начальная школа. 1996. № 12.
22. *Тур С. Н., Бокучава Т. П.* Первые шаги в мире информатики: Методическое пособие для учителей 1—4 классов. СПб.: БХВ-Петербург, 2002.
23. *Тур С. Н., Бокучава Т. П.* Первые шаги в мире информатики. Рабочая тетрадь. СПб.: БХВ-Петербург, 2002.
24. *Целищева И. И., Румянцева И. Б., Ермакова Е. С.* Обучение решению комбинаторных задач детей 4—10 лет // Начальная школа. 2005. № 11.



ТЕОРИЯ И ПРАКТИКА ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ

К. И. Мацуца,

*науч. сотрудник Института содержания и методов обучения
Российской академии образования*

МЕДИАОБРАЗОВАТЕЛЬНЫЙ АНАЛИЗ ЦЕЛЕЙ ОБУЧЕНИЯ ИНФОРМАТИКЕ В ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ

Рассмотрим изменение целей обучения информатике с момента зарождения учебного предмета до наших дней для выявления медиаобразовательной составляющей. Сразу отметим, что в полном объеме цели медиаобразования, интегрированного в гуманитарные и естественнонаучные дисциплины начального общего и среднего общего образования, ни у одного автора не были реализованы [1]. Также характерной чертой рассматриваемых концепций является стопроцентная реализация четвертой цели стандарта медиаобразования — формирования умений находить, готовить, передавать и принимать требуемую информацию, в том числе с использованием различного технического инструментария (компьютеры, модемы, факсы, мультимедиа и др.). В связи с этим большой интерес представляет рассмотрение степени реализации первых трех целей стандарта медиаобразования.

При проведении анализа целей обучения информатике необходимо ответить на следующие ключевые вопросы:

1. В чем заключаются отличия или что общего в подхо-

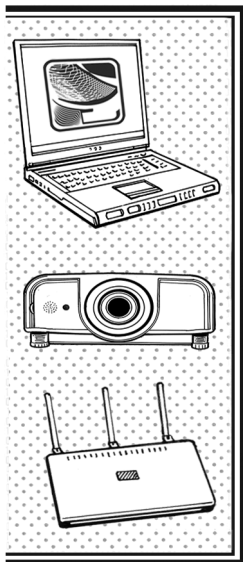
дах разных авторов, а также авторских коллективов при постановке целей обучения информатике с момента зарождения учебного предмета до сегодняшнего дня с медиаобразовательных позиций?

2. Какие медиаобразовательные компоненты были включены авторами, авторскими коллективами в цели обучения информатике и в каких пропорциях?

Прежде чем проводить анализ изменения целей обучения информатике в контексте медиаобразования, изложим цели медиаобразования, сформулированные в различных концепциях [7].

Впервые цели медиаобразования разных научных школ были сформулированы и изложены в общем виде Л. С. Зазнобиной в «Стандарте медиаобразования, интегрированного в гуманитарные и естественнонаучные дисциплины начального общего и среднего общего образования», [1] следующим образом:

- обучение восприятию и переработке информации, передаваемой по каналам СМИ (в широком толковании);
- развитие критического мышления, умений понимать скрытый смысл того или иного сообщения, противостоять манипулированию сознанием индивида со стороны СМИ;
- включение внешкольной информации в контекст общего базового образования, в систему формируемых в



Уважаемые читатели! В рубрике «Теория и практика информатизации образования» публикуются результаты диссертационных исследований в области информатизации образования и методики преподавания информатики. Для их размещения объем журнала, в котором публикуются данные статьи, специально увеличивается по сравнению с обычным (96 страниц) на 32 страницы. Обращаем внимание, что это увеличение объема не влияет на стоимость журнала, которая одинакова для всех номеров текущего полугодия. Материалы в данную рубрику просим присылать заблаговременно.

Цели медиаобразования в различных концепциях

Автор концепции	Цели медиаобразования
Ю. Н. Усов, Г. А. Поличко, С. Н. Пензин	Развитие личности при использовании материала художественных медиатекстов
О. А. Баранов	Эстетическое, аудиовизуальное, эмоционально-интеллектуальное, этическое образование/воспитание аудитории, развитие личности при использовании материала художественных медиатекстов
А. В. Шариков	Развитие медиакommunikативных способностей учащихся, критического мышления [8, с. 28], социокультурное развитие личности при использовании материала средств массовой коммуникации
Л. С. Зазнобина	Подготовка школьников к жизни в новых информационных условиях (информационная компетентность и свободное обращение с информационным потоком)
А. В. Спичкин	Аудиовизуальная грамотность и развитие критического мышления учащихся
Е. Л. Вартанова, Я. Н. Засурский	Формирование у молодежи творческого и критического отношения к медиа, превращение ее в креативного пользователя СМИ в дальнейшей жизни после учебного заведения (школы, вуза)
Н. А. Леготина	Формирование готовности студентов университетов к реализации медиаобразования в общеобразовательных учреждениях
Л. Мастерман	Обучение аудитории пониманию того, как медиа переосмысляет/представляет реальность, «декодировать», критически анализировать медиатексты, ориентироваться в информационном/идеологическом потоке современного общества
А. Силверблэт	Развитие критической автономии личности — умения, которое дает возможность школьникам/студентам быть осознанно независимыми от однообразных предпочтений медиатекстов
К. Бээлэгэт, Д. Букингэм, Э. Харт	Подготовка людей к жизни в демократическом медийном обществе, опираясь на шесть ключевых понятий медиаобразования: «агентство медиа» (agency), «категория медиа» (category), «язык медиа» (language), «технология медиа» (technology), «репрезентация медиа» (representation), «аудитория медиа» (audience), в трактовке Д. Букингэма ключевые понятия «агентство», «категория» и «технология» объединены в одно — «производство»
Дж. Поттер	Развитие понимания медиа и их сообщений; усиление контроля над процессом интерпретации и, таким образом, повышение способности к оценке медиа/медиатекстов
Н. Андерсен, Б. Дункан, Дж. Пандженте	Опираясь на семь ключевых положений медиаобразования, подготовка людей к жизни в демократическом медийном обществе
А. В. Федоров	Формирование у аудитории культуры общения с медиа, творческих, коммуникативных способностей, развитие творческого, критического мышления [8, с. 27]

предметных областях знаний и умений;

- формирование умений находить, готовить, передавать и принимать требуемую информацию, в том числе с использованием различного технического инструментария (компьютеры, модемы, факсы, мультимедиа и др.).

Итак, на основании представленных позиций медиаобразования проанализируем цели обучения информатике.

В Концепции содержания обучения информатике в 12-летней школе [2, с. 17–22] основная цель изучения информатики в школе была сформулирована без учета

большей части медиаобразовательных возможностей. Она рассматривалась в свете необходимости обеспечить прочное и сознательное овладение учащимися основами знаний о процессах получения, преобразования, передачи и использования информации и на этой основе раскрыть учащимся значение информационных процессов в формировании современной научной картины мира, роль информационной технологии и вычислительной техники в развитии современного общества, привить учащимся навыки рационального использования компьютеров в своей учебной, а затем в профессиональной деятельности.

В научном направлении, развиваемом в Российской академии образования (В. С. Леднев, А. А. Кузнецов, С. А. Бешенков) [3], выявляется и включается в содержание общего образования *базовое кибернетическое образование*, направленное на изучение феномена самоуправляемости, в основе которого лежат процессы передачи, восприятия, хранения и переработки информации. Авторским коллективом рассматривались четыре цели:

1. Формирование основ научного мировоззрения.

2. Формирование общеучебных и общекультурных навыков работы с информацией.

3. Подготовка школьников к последующей профессиональной деятельности.

4. Овладение информационными и коммуникационными технологиями как необходимое условие перехода к непрерывному образованию.

Первые три цели обучения информатики по В. С. Ледневу и другим направлены на формирование общекультурных умений работы с информацией. Тем не менее первые три цели медиаобразования необходимо выделить, усилить и выразить более конструктивно.

Е. А. Ракина [5, с. 113] сформулировала основную цель курса информатики с двух позиций: с точки зрения интересов общества — формирование информационной культуры на уровне, соответствующем информационному этапу развития общества, и с точки зрения личностного развития и самоопределения — формирование культуры работы с информацией как неотъемлемого компонента общей культуры индивида. Бинарный характер определения цели — интересный подход, но в то же время считаем, что необходимо конкретизировать поставленную цель в медиаобразовательном аспекте. Перечисленные линии Е. А. Ракиной по воспитанию качеств в информатике [5, с. 90] носят медиаобразовательный характер и могут быть взяты в качестве ориентиров при определении воспитательных целей медиаобразования, интегрированного в курс информатики. Кроме того, представляют чрезвычайный интерес выводы [5, с. 129], в том числе вывод о том, что курс информатики призван сделать процесс формирования информационной культуры целенаправленным, осмысленным и управляемым.

В стандарте основного общего образования 2004 г. по информатике [6, с. 124] определены следующие цели:

- освоение знаний, составляющих основу научных представлений об информации, информационных процессах, системах, технологиях, моделях;
- овладение умениями работать с различными видами информации с по-

мощью компьютера и других средств информационных и коммуникационных технологий (ИКТ), организовывать собственную информационную деятельность и планировать ее результаты;

- развитие познавательных интересов, интеллектуальных и творческих способностей средствами ИКТ;
- воспитание ответственного отношения к информации с учетом правовых и этических аспектов ее распространения, избирательного отношения к полученной информации;
- выработка навыков применения средств ИКТ в повседневной жизни, при выполнении индивидуальных и коллективных проектов, в учебной деятельности, при дальнейшем освоении профессий, востребованных на рынке труда.

Очевидно, что первая (косвенно) и четвертая цели медиаобразования отражены. Вторая цель медиаобразования частично, но недостаточно отражена в воспитательной цели стандарта. Третья цель медиаобразования также частично заложена в последней цели стандарта. И, как итог, цели медиаобразования не нашли достаточного, полного и логичного изложения в стандарте основного общего образования, хотя и присутствуют в той или иной степени.

В стандарте среднего (полного) общего образования 2004 г. по информатике были определены следующие цели [6]:

- освоение и систематизация знаний, относящихся к математическим объектам информатики, построению описаний объектов и процессов, позволяющих осуществлять их компьютерное моделирование, средствам моделирования, информационным процессам в биологических, технологических и социальных системах;
- овладение умениями строить математические объекты информатики, в том числе логические формулы и программы на формальном языке, удовлетворяющие заданному описанию, создавать программы на языке программирования по их описанию, применять общепользовательские инструменты и настраивать их для нужд пользователя;
- развитие алгоритмического мышления, способностей к формализации, элементов системного мышления;
- воспитание чувства ответственности за результаты своего труда; формирование установки на позитивную социальную деятельность в информационном обществе, на недопустимости действий, нарушающих правовые, этические нормы работы с информацией;

- приобретение опыта проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств, построения компьютерных моделей, коллективной реализации информационных проектов, информационной деятельности в различных сферах, востребованных на рынке труда.

Анализ позволяет сделать вывод о том, что первые три цели стандарта медиаобразования, интегрированного в гуманитарные и естественнонаучные дисциплины начального общего и среднего общего образования, реализуются в не полной мере, но вместе с тем сформулированные цели стоит принять во внимание при постановке целей медиаобразования, интегрированного в курс информатики.

Сильной стороной целеполагания Н. В. Макаровой [4, с. 13] является системный культурологический подход, при котором уделено внимание проблемам формирования умения применять, анализировать, преобразовывать информационные модели реальных объектов и процессов, развития логического мышления, творческого и познавательного потенциала школьника, его коммуникативных способностей, а также воспитания ответственного отношения к соблюдению этических и правовых норм информационной деятельности. Кроме того, необходимо принять во внимание справедливое мнение автора о том, что развитием мышления занимаются практически при изучении всех школьных предметов, но системно нигде. Таким образом, можно сказать, что частично первая и вторая цели медиаобразования нашли отраже-

ние в сформулированной системно-информационной концепции Н. В. Макаровой.

Итак, проведенный анализ подходов к формированию целей обучения информатике свидетельствует о необходимости уточнения целей обучения информатике и ИКТ с учетом целей медиаобразования, интегрированного в данный курс.

Литература

1. *Зазнобина Л. С.* Стандарт медиаобразования, интегрированного в гуманитарные и естественнонаучные дисциплины начального общего и среднего общего образования // Стандарты и мониторинг в образовании. 1998. № 2
2. Концепция содержания обучения информатике в 12-летней школе // ИНФО. 2000. № 2.
3. *Леднев В. С., Кузнецов А. А., Бешенков С. А.* О теоретических основах содержания обучения информатике в общеобразовательной школе // ИНФО. 2000. № 2.
4. *Макарова Н. В.* Программа по информатике и ИКТ (системно-информационная концепция). СПб.: Питер, 2008.
5. *Ракитина Е. А.* Построение методической системы обучения информатике на деятельностной основе: Дис. ... д-ра пед. наук. М., 2002.
6. Федеральный компонент государственного стандарта общего образования. Ч. II. Среднее (полное) общее образование / Министерство образования Российской Федерации. М., 2004.
7. *Федоров А. В.* Развитие медиакомпетентности и критического мышления студентов педагогического вуза. М.: Изд-во МОО ВПП ЮНЕСКО «Информация для всех», 2007.
8. *Чельшева И. В.* Теория и история российского медиаобразования. Таганрог: Изд-во Кучма, 2006.

Е. А. Васенина,

*канд. пед. наук, доцент кафедры информатики и методики обучения информатике
Вятского государственного гуманитарного университета, г. Киров*

ЭКСПЕРИМЕНТ И ИССЛЕДОВАТЕЛЬСКАЯ ДЕЯТЕЛЬНОСТЬ НА КОМПЬЮТЕРНОМ УРОКЕ ИНФОРМАТИКИ

В современных условиях, когда компьютер и другие средства ИКТ прочно вошли в образовательный процесс, на первый план выходят вопросы эффективности их применения для повышения качества результатов обучения.

Выделим как один из основных показателей эффективности обучения высокую познавательную активность учащихся, которая предполагает:

- прочную мотивацию и осознание ценности изучаемого;

- включенность ученика в деятельность по разрешению проблем и противоречий;
- формирование нового знания как итог осмысления выполненной деятельности — как самого процесса, так и полученных результатов;
- самостоятельность в познании не только на уровне выполнения тех или иных действий, но и в выборе средств и методов деятельности, ее планировании и оценке ее результа-

тов, т. е. движение от самостоятельности действия к самостоятельности мысли.

При этом познавательная активность учеников инициируется и поддерживается усилиями учителя, задачей которого является не просто информирование ученика, передача ему определенного количества содержательных единиц материала, но в первую очередь создание условий для понимания и усвоения (присвоения) изучаемого содержания. При этом требуется достижение баланса между самостоятельностью ученика в познании и педагогическим руководством его познавательной деятельностью.

Безусловно, применение на уроке компьютера и других средств ИКТ увеличивает возможности для реализации каждого из перечисленных факторов, однако сам только факт их применения еще не означает обязательного повышения познавательной активности учеников.

Рассмотрим такую важную составляющую урока, как формирование нового знания. Как правило, учителя как информатики, так и других школьных дисциплин применяют средства ИКТ для иллюстрирования своего рассказа, чтобы сделать его более ярким, наглядным. При этом они используют статические или динамические информационные объекты, несущие информацию о реальных объектах и явлениях, которую ученикам требуется усвоить. В этом случае основная функция средств ИКТ состоит в визуализации изучаемого материала, что позволяет разнообразить формы его предъявления и осуществлять более мощное воздействие на органы чувств и эмоциональную сферу учащихся. Яркие образы, без сомнения, привлекают внимание учащихся, но интерес их пассивен, это, скорее, интерес потребителя знаний, но не их творца. Кроме того, возможно и отрицательное влияние на качество усвоения знания. Излишние наглядность и яркость могут привести к перегрузке канала восприятия, отвлечь от сути, переключив внимание ученика на форму представления учебной информации, а излишняя визуализация выключает воображение, мешает формированию собственного «внутреннего образа» изучаемого материала, продлевает фазу конкретно-образного мышления в ущерб формированию способности к абстракции. Всё это требуется учитывать при выборе или создании демонстрационных информационных объектов.

Большую самостоятельность в познании ученики проявляют в ситуации включенности в информационный поиск. Для этого используются информационно-поисковые системы, через которые можно получить доступ (в том числе и удаленный) к сведениям, составляющим содержание изучаемой дисциплины. Чаще всего такой метод ис-

пользуется для организации самостоятельной работы учеников вне урока, однако и на уроке учащиеся могут выполнить поиск и отбор информации по интересующему вопросу, которую можно использовать для описания того или иного явления, для аргументации той или иной позиции, для нахождения ответа на поставленный вопрос.

Однако наилучшего результата можно достичь в случае, когда на уроке организована экспериментальная деятельность с использованием компьютера. Тогда новое знание формируется не из пассивного восприятия и запоминания информации, но как результат исследования, проводимого учеником в сотрудничестве с учителем. При этом соотношение между самостоятельностью и педагогическим руководством познавательной деятельностью можно подобрать индивидуально для каждого ученика.

Такой компьютерный эксперимент может быть организован при изучении различных предметов, преимущественно естественнонаучного цикла. В *компьютерных лабораториях* физики, химии, биологии и других учащиеся работают с компьютерными программами, которые моделируют явления и процессы с целью выявления характеризующих их закономерностей. При этом в распоряжение ученика предоставляется не сам изучаемый объект, а его компьютерная модель, в рамках которой ученик выполняет воздействия на «виртуальный образ» изучаемого объекта с целью выявления закономерностей его поведения. Однако компьютерную модель ученик получает, как правило, в готовом виде, она уже построена, закономерности поведения объектов заданы, хотя и неважно для ученика. Экспериментируя, ученик, понимает факт «виртуальности» объекта эксперимента, той самой заданности его поведения; отсюда некоторая ограниченность в ощущении реальности происходящего и в осознании самостоятельности познания. Тем не менее даже такие ограничения не снижают ценности данного вида учебной работы, поскольку самостоятельная экспериментальная деятельность учеников в значительной мере активизирует процесс познания.

В этом отношении позиция информатики как учебного предмета намного предпочтительнее, ибо предметом изучения и экспериментального исследования являются собственно информационные объекты — разрабатываемая программа, исследуемая информационная среда или создаваемый в этой среде информационный продукт (текстовый документ, электронная таблица, электронная база данных и др.).

На уроке информатики компьютер выступает не только как средство обучения, но и как объект изучения, соединяющий аппаратную и программную составляющие и являющийся универсальным инструмен-

том реализации информационных процессов, в работе которого проявляются закономерности их протекания и находят воплощение методы работы с информацией. При этом формирование теоретического знания возможно в процессе реальной экспериментальной деятельности с перечисленными информационными объектами, в ходе которой ученик может манипулировать ими, что в значительной мере повышает эффективность обучения и качество его результатов, поскольку приближает процесс обучения к реальному процессу познания.

Замечательно то, что результат экспериментирования во многом неизвестен не только ученику, но порой и учителю — достаточно часты ситуации, когда не самые разумные действия ученика приводят к непредсказуемым результатам, совместный поиск учителя и ученика направлен на понимание причины создавшейся нештатной ситуации и определение пути выхода из нее. Однако опыт и знание, извлеченное из такого поиска, имеют значительно большую ценность.

Важно также новое качество наглядности, визуализации изучаемого в процессе компьютерного эксперимента. Когда ученик на экране наблюдает воплотившийся результат его усилий, имеет место визуализация *результата мыслительной деятельности*, которая выражается в построении образа усвоенного, выявленной структуры, установленных связей и т. д., в отличие от рассмотренной выше визуализации предъявления материала.

К сожалению, изучение нового материала в режиме исследования и эксперимента на уроках информатики встречается вовсе не так часто, как это можно предположить, исходя из возможностей компьютера как качественно нового средства обучения. Как правило, учитель в информационно-инструктивном ключе объясняет основные теоретические положения, показывает образцы выполнения заданий или решения задач. Затем ученики самостоятельно выполняют задания на уровне воспроизведения образца или реконструктивно-вариативном уровне, руководствуясь при этом инструкцией той или иной степени подробности, которая часто не оставляет пространства для раздумий. Отдельным ученикам предлагаются задания эвристического характера, но всё это в рамках закрепления преподанного и формирования заданного круга умений.

В основном применяются школьная лекция, фронтальная лабораторная работа в режиме комментирования — в ходе ее учитель комментирует действия учеников и результаты, к которым эти действия должны привести, дает точные указания, которые тут же непосредственно и синхронно выполняются учениками, а также самостоя-

тельная лабораторная (практическая) работа с применением инструкции.

Рассмотрим, каким образом может быть представлена структура изучения нового материала.

1. Мотивация (мотивационная задача).

Изучение нового непременно начинается с мотивации, и наилучшей мотивацией будет служить задача, которую невозможно решить без этого нового знания. Это может быть модификация уже решенной ранее задачи, когда новые условия делают ее неразрешимой без новых средств. Но может быть и так называемая *ситуативная задача*, описывающая некоторую бытовую, практическую ситуацию, в которой может быть применено новое знание.

2. Сообщение начального знания.

Оно протекает в форме беседы, в ходе которой учитель и ученики обсуждают *основные идеи* изучаемой темы. При этом говорится далеко не всё: в стороне оставлены факты, которые можно извлечь из справочных материалов, и закономерности, которые предстоит выявить экспериментально. Важно, чтобы в сознании ученика наметились логически связанные между собой опорные точки, которые определяют направление его деятельности.

3. Экспериментальная работа.

Учитель готовит и предлагает цепочку задач и заданий, в ходе решения которых ученики сталкиваются с рядом проблем и затруднений, экспериментируют и, направляемые учителем, приходят к их разрешению. При этом предусмотрена более или менее длительная самостоятельная экспериментальная работа, за ходом которой учитель внимательно наблюдает и оперативно реагирует на все изменения, происходящие как с отдельными учениками, так и с системой, которую образует класс (учебная группа) в целом.

Такую деятельность целесообразно организовать в форме *фронтальной лабораторной работы в исследовательском режиме*. Основное отличие *фронтальной* лабораторной работы в том, что ученики работают с компьютером синхронно, эта работа направляется учителем по заранее подготовленному плану. Указание на *исследовательский* режим говорит о том, что учебная деятельность будет носить частично-поисковый, и даже исследовательский характер. Важно продумать систему мини-проблем, которые ученики будут разрешать в ходе работы.

Хотя работа в основном синхронизирована и управляется учителем, следует предоставить ученикам определенную самостоятельность. Ее доля поначалу невелика, но по мере роста интереса и способности учеников к исследовательской деятельности эта доля также будет расти. Фронтальная рабо-

та под руководством учителя дает ученикам образец для будущего самостоятельного исследования — пример того, какие надо задавать вопросы, как строить последовательность опытов и т. д.

Чтобы учесть индивидуальный темп работы учеников, в составе класса (учебной группы) может быть выделено несколько групп учеников, одна из которых работает синхронно с учителем, другая «бежит» чуть впереди, т. е. успевает выполнить еще несколько заданий, а третья выполняет только обязательный минимум и пользуется результатами экспериментов, добытыми первой и второй группами.

4. Обобщение результатов эксперимента, формулирование выводов.

Экспериментальная работа непременно должна завершаться обсуждением результатов и формулированием выводов, составляющих основу нового теоретического знания, полученного на уроке.

Важно, чтобы это новое знание нашло достаточно выразительную форму для своего представления. С одной стороны, это краткие, точные, понятные формулировки, достаточно полно описывающие изучаемую тему, т. е. выражение знания в вербальной форме, которое является неотъемлемым атрибутом формирования понятий. С другой стороны, это построение целостного визуального образа этой темы, т. е. забота о хорошем гештальте темы, облегчающая ее восприятие, запечатление в памяти, последующую обработку и длительное хранение.

Записи в тетради целесообразно сделать именно на данном этапе. Если на этапе начального знакомства можно было наметить определенный план записи (можно использовать прием оставления свободного места), то сейчас этот план детализируется, оставленное место заполняется. На первых порах эта работа выполняется вместе с учителем, под его непосредственным руководством. При этом следует обращать внимание на оформление, но не с позиций художественности, а с позиций структурированности. Хорошо, если учитель имеет четко очерченный мысленный образ тетради ученика, который затем воплощается в реальную запись в тетради, а она, в свою очередь, составляет основу мысленного образа изучаемой темы в сознании ученика.

В дальнейшем выполнение записей в тетради можно постепенно организовать в качестве самостоятельной работы учеников. Результат необходимо обязательно проверять, обращая внимание на качество работы с информацией — отбор, структурирование, представление. При этом аккуратность может отойти на второй план.

В сознании ученика формируется некий схематический каркас, который при

дальнейшем изучении будет упорядоченно заполняться и обогащаться новыми фактами и сведениями. Это уже не будет хаотичный набор информационных единиц — каждая такая единица найдет свое место в сформированной схеме, в системе связей с другими единицами. Причем основой этого каркаса станут те опорные точки, которые были намечены при обсуждении основных идей темы.

5. Применение нового в соединении с обогащением знаний, формированием умений и навыков. Методика этого этапа заслуживает отдельного, серьезного, длительного и подробного рассмотрения. Поскольку сейчас обсуждается формирование нового знания, скажем только, что дальнейшая деятельность учеников и учителя будет направлена на его закрепление, применение в различных ситуациях и обогащение выстроенного схематического каркаса, размещение в его узлах новой информации, нового опыта, приобретенного в ходе этого применения.

Безусловно, эта деятельность связана с решением задач. Практически в каждой задаче можно увидеть минимум два возможных уровня ее решения, что позволяет исследовать проблему с разной степенью глубины. Если для первого уровня можно подобрать образец из тех, что были разобраны в совместном обсуждении с учителем, то движение в глубь задачи постепенно лишает ученика подобных «подпорок», заставляя мыслить самостоятельно. Такое самостоятельное осмысление вполне конкретной задачи ведет к формированию абстрактного теоретического знания, обогащает представление ученика о проблеме, которая исследовалась на уроке, позволяет взглянуть на нее с разных сторон. Каркас начальных сведений обрастает живой тканью знания, причем каждый новый увиденный факт, отмеченное явление или закономерность находят свое место в общей картине изучаемой темы.

6. Формирование целостного образа темы.

Здесь, на новом витке, мы обращаемся к синтезу, последовательное развитие способности к которому (обучение синтезу) ведет к формированию целостного образа конкретной темы, а в дальнейшем содержания учебного предмета и далее целостной картины мира, системного подхода к ее осмыслению.

Однако для реализации описанных выше возможностей и преимуществ, которые открываются на компьютерном уроке информатики, требуется, чтобы формы, методы и сама структура урока были ориентированы на проведение эксперимента, организацию исследовательской деятельности, в ходе которой ученик активно познает и усваивает новое.

С. А. Тарнаева,

Нижегородский государственный технический университет им. Р. Е. Алексеева

КОМПЛЕКС ПРОГРАММНО-ПЕДАГОГИЧЕСКИХ И ТЕЛЕКОММУНИКАЦИОННЫХ СРЕДСТВ ПО ИНФОРМАТИКЕ КАК СРЕДСТВО ФОРМИРОВАНИЯ ИНФОРМАЦИОННОЙ КОМПЕТЕНТНОСТИ БУДУЩИХ ИНЖЕНЕРОВ В ТЕХНИЧЕСКОМ ВУЗЕ

Основной задачей, стоящей перед современным высшим образованием, является подготовка высококвалифицированных инженерных кадров. В нормативных документах РФ в области образования задачам информатизации уделяется большое значение. В связи с этим возникает необходимость компетентного подхода в образовании будущих инженеров. Одним из путей решения этой проблемы является формирование в процессе обучения информатике информационной компетентности, как важнейшей компоненты профессиональной компетентности.

Информационная компетентность инженера — качественная характеристика субъекта профессиональной деятельности, которая определяется как совокупность информационных знаний, умений и навыков, необходимых для решения поставленных профессиональных задач. Информационная компетентность современного инженера предполагает, что кроме знаний технологии своего основного производства он должен владеть несколькими информационными и телекоммуникационными системами, встроенными в технологию и производство, поддерживать свой уровень квалификации в условиях постоянно изменяющихся информационных и телекоммуникационных средств и быть готовым оперативно освоить новый или не применявшийся им ранее информационный ресурс.

В соответствии с концептуальными положениями реформы системы образования в высшей школе, декларируемыми в Доктрине национального образования в России, подготовка будущего специалиста в условиях модернизации образования должна отражать перспективные тенденции развития информационных педагогических технологий. Актуальными являются вопросы использования программно-педагогических и телекоммуникационных средств (ППТС) обучения информатике студентов — будущих инженеров, оптимизации различных форм организации обучения.

Для формирования информационной компетентности студентов — будущих инженеров на кафедре «Прикладная математика» НГТУ им. Р. Е. Алексеева был разработан комплекс ППТС преподавания информатики, реализованный с помощью

информационно-методической системы интерактивного обучения информатике DEIS. Данная система включает:

- электронный учебник «Информатика для всех» по дисциплине «Информатика»;
- комплекс анимационных и интерактивных моделей;
- тестирующий комплекс, интегрированный с базой данных задач;
- тренирующий блок, имеющий примеры решения задач;
- справочный комплекс, содержащий список основных формул по всем темам раздела «Численные методы», используемых в численных расчетах постоянных, а также глоссарий по всем терминам электронного учебника «Информатика для всех»;
- предметный поиск по ключевым словам;
- систему помощи с путеводителем по курсу и инструкциями по работе;
- обзор интернет-ресурсов по информатике;
- рекомендуемый список литературы по различным темам курса;
- систему методической поддержки курса.

При создании интерактивного курса «Информатика для всех» особое внимание уделялось выполнению курсовых работ, связанных с выбранной специальностью. Для каждой курсовой работы были созданы интерактивные модели, позволяющие, варьируя заданные параметры, изучать процессы в изучаемых системах. Например, при выполнении курсовой работы «Численное моделирование и анализ переходных процессов в электрической цепи» студенты-электрики, выбирая метод решения, изменяя значения заданных параметров, могут наблюдать изменение возникающих в цепи силы тока и напряжения, а также определять количество теплоты, выделяемой на определенном резисторе за определенный промежуток времени. Интерактивная модель позволяет выполнить численные расчеты на любом изучаемом языке программирования.

С целью подтверждения эффективности выбранного направления совершенствования системы преподавания информатики и формирования информационной компе-

тентности студентов — будущих инженеров на протяжении 2006—2008 гг. проводился педагогический эксперимент, в результате которого:

1) предложены новые интерактивные модели обучения, использующие инновационные информационные и телекоммуникационные технологии, учитывающие вариативность и индивидуализацию профессионального образования и направленные на повышение уровня информационной компетентности;

2) сделан вывод о том, что комплексное применение новых ППТС обучения информатике способствует достижению более высокого уровня информационной компетентности студентов.

Анализ результатов педагогического эксперимента подтверждает гипотезу о том, что существует связь между применением новых информационных и телекоммуникационных технологий и повышением уровня сформированности информационной компетентности студентов — будущих инженеров.

М. Б. Суханов,

канд. техн. наук, доцент кафедры математического моделирования и оптимизации химико-технологических процессов Санкт-Петербургского государственного технологического института (технического университета),

А. Г. Суханова,

канд. техн. наук, ст. преподаватель кафедры математического моделирования и оптимизации химико-технологических процессов Санкт-Петербургского государственного технологического института (технического университета)

ОПРЕДЕЛЕНИЕ КОЭФФИЦИЕНТОВ НЕЛИНЕЙНЫХ ЗАВИСИМОСТЕЙ МЕТОДОМ НАИМЕНЬШИХ КВАДРАТОВ В OPEN OFFICE.ORG CALC

Необходимость определения коэффициентов нелинейной зависимости по имеющимся эмпирическим данным довольно часто возникает при решении естественнонаучных, технических и экономических задач.

В табличном процессоре Microsoft Excel 2007 решение нелинейных задач оптимизации удобно осуществлять с помощью надстройки **Поиск решения**. В табличном процессоре Calc, входящем в состав свободно распространяемого пакета Open Office.Org 3.0, средство **Поиск решения** позволяет решать задачи оптимизации только для линейных моделей. Рассмотрим, как можно найти коэффициенты в случае нелинейной зависимости в электронной таблице Calc по аналитическим формулам, получаемым методом наименьших квадратов. Этот метод может быть использован и при идентификации параметров нелинейных зависимостей в MS Excel и Mathcad.

Прежде всего, с учащимися целесообразно рассмотреть метод наименьших квадратов в общем виде: имеется функциональная зависимость $y = f(x)$, вид зависимости известен. В эту зависимость входят некоторые постоянные a , b , значения которых неизвестны, т. е.

$$y = f(x, a, b). \quad (1)$$

Для нахождения наиболее вероятных значений параметров используют серию из-

вестных значений x и y , т. е. n пар значений $x_{i\text{эмп}}$ и $y_{i\text{эмп}}$. Требуется найти такие значения a и b , которые при подстановке в (1) вместе с $x_{i\text{эмп}}$ дали бы значение $y_{i\text{расч}}$, наиболее близкие к $y_{i\text{эмп}}$.

За меру близости $y_{i\text{расч}}$ и $y_{i\text{эмп}}$ принимают величину

$$S = \sum_{i=1}^n (y_{i\text{эмп}} - y_{i\text{расч}})^2 = \sum_{i=1}^n (y_{i\text{эмп}} - f(x_{i\text{эмп}}, a, b))^2. \quad (2)$$

Тогда задача сводится к поиску таких a и b , при которых S будет наименьшей (наименьшая сумма квадратов). Поэтому идентификация параметров нелинейной зависимости представляет собой частный случай задачи оптимизации.

Известно, что в точке экстремума производные от функции нескольких переменных по этим переменным должны быть равны нулю, т. е. в случае зависимости (1) имеют вид:

$$\begin{cases} \frac{\partial S}{\partial a} = 0, \\ \frac{\partial S}{\partial b} = 0. \end{cases} \quad (3)$$

Заметим, что в общем случае система (3) нелинейна.

Используя соотношения (2) и (3), получаем систему двух уравнений с двумя неизвестными a , b :

$$\begin{cases} \sum_{i=1}^n f[(x_i, a, b) - y_i] \frac{\partial S(x_i, a, b)}{\partial a} = 0, \\ \sum_{i=1}^n [f(x_i, a, b) - y_i] \frac{\partial S(x_i, a, b)}{\partial b} = 0. \end{cases} \quad (4)$$

В частном случае аппроксимации данных с помощью линейной функции

$$y = f(x, a, b) = a + b \cdot x \quad (5)$$

имеем

$$\frac{\partial f}{\partial a} = 1, \quad \frac{\partial f}{\partial b} = x.$$

После подстановки в (4) получаем:

$$\begin{cases} \sum_{i=1}^n [f(x_i, a, b) - y_i] = 0, \\ \sum_{i=1}^n [f(x_i, a, b) - y_i] x_i = 0. \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} a \cdot n + b \cdot \sum_{i=1}^n x_i = \sum_{i=1}^n y_i, \\ a \cdot \sum_{i=1}^n x_i + b \cdot \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i. \end{cases} \quad (6)$$

Если обозначить

$$\begin{aligned} \sum_{i=1}^n x_i &= S_1; & \sum_{i=1}^n y_i &= S_2; \\ \sum_{i=1}^n x_i^2 &= S_3; & \sum_{i=1}^n x_i y_i &= S_4, \end{aligned}$$

то система (6) перейдет в систему

$$\begin{cases} a \cdot n + b \cdot S_1 = S_2, \\ a \cdot S_1 + b \cdot S_3 = S_4. \end{cases} \quad (7)$$

Откуда наиболее вероятные для данного набора x_i и y_i значения a , b :

$$\begin{aligned} a &= \frac{S_3 S_2 - S_1 S_4}{S_3 n - S_1^2}, \\ b &= \frac{S_4 n - S_2 S_1}{S_3 n - S_1^2}. \end{aligned} \quad (8)$$

После изложения идеи метода наименьших квадратов целесообразно рассмотреть его применение, например, для нахождения коэффициентов в законе Мура.

Гордон Мур, один из основателей компании Intel, в 1965 г. сделал весьма знаменательное наблюдение, позже получившее

название *закон Мура*. Представив в виде графика рост производительности микросхем памяти, он обнаружил любопытную закономерность: плотность транзисторов на кремниевой подложке удваивается каждые 18–24 месяца, соответственно в 2 раза растет их производительность, в 2 раза падает их рыночная стоимость. Если такая тенденция продолжится, заключил Мур, то мощность вычислительных устройств экспоненциально возрастет на протяжении относительно короткого промежутка времени. Наблюдение Мура впоследствии блестяще подтвердилось, а обнаруженная им закономерность наблюдается с некоторой коррекцией и в наши дни, являясь основой для многочисленных прогнозов улучшения характеристик компьютеров вообще и процессоров в частности.

Постановка задачи, предлагаемой учащимся, имеет вид. Пользуясь известными данными о количестве транзисторов в процессорах фирмы Intel, методом наименьших квадратов найти значения параметров N_0 , T , входящих в уравнение эмпирического закона Мура.

Математически закон Мура можно представить эмпирической зависимостью:

$$N(t) = N_0 \cdot 2^{\frac{t}{T}}, \quad (9)$$

где N_0 — количество транзисторов на некоторый момент времени (условно считаем его нулевым), $N(t)$ — число транзисторов на кристалле спустя t лет и T — срок (в днях, значение которых уменьшено на 1000), за который число транзисторов возрастает вдвое (см. табл. 1).

Рассмотрим решение данной задачи.

Зависимость между t_i и y_i является нелинейной (точнее, показательной):

$$N_i = N_0 \cdot 2^{\frac{t_i}{T}},$$

но может быть приведена к линейной логарифмированием:

$$\log_2 N_i = \log_2 N_0 + t_i / T. \quad (10)$$

Если обозначить $\log_2 N_i = y_i$; $1/T = b$; $t_i = x_i$; $\log_2 N_0 = a$, то зависимость (10) сведется к линейной зависимости (5). Тогда поиск неизвестных коэффициентов N_0 , T можно будет осуществить по формулам:

$$N_0 = 2^a, \quad T = 1/b. \quad (11)$$

Расчет коэффициентов a и b осуществим по формулам (8).

Фрагмент рабочего листа с исходными данными и получающимися при них результатами в *Calc* имеет вид (см. табл. 2).

Найденное время T составило 716 дней, т. е. удвоение числа транзисторов происходило примерно за 2 года ($716/365 \approx 1,962$).

Для визуальной оценки качества идентификации параметров целесообразно пред-

Таблица 1

Количество транзисторов в процессорах фирмы Intel

	Процессор	Дата выпуска	Количество транзисторов, тыс. шт.
1	4004	15.11.1971	2,3
2	8080	01.04.1974	4,5
3	80186	01.01.1982	29,0
4	80286	01.02.1982	134,0
5	Intel 386 DX	17.10.1985	275,0
6	Intel 486 SX	22.04.1991	900,0
7	Intel Pentium	22.03.1993	3100,0
8	Intel Pentium II	07.05.1997	7500,0
9	Intel Pentium III	26.02.1999	9500,0
10	Intel Celeron	22.03.1999	19000,0
11	Intel Pentium 4	20.11.2000	42000,0
12	Intel Xeon MP	12.03.2002	108000,0
13	Intel Itanium 2	08.07.2002	220000,0
14	Intel Itanium 2	30.06.2003	410000,0
15	Yonah Core Dual	01.11.2006	291000,0
16	Madison	01.02.2006	410000,0
17	Pentium 965 EE	01.04.2006	376000,0
18	Intel Core2 Quad Processor Q9550	01.01.2008	820000,0
19	Intel Core2 Duo Processor E8300	01.04.2008	410000,0
20	Intel Core i7-965 Extreme Edition	01.11.2008	731000,0

Таблица 2

Дата выпуска	Количество транзисторов, тыс.шт., $U_{\text{эмп}}$	Условное время	Время, t	t^*2	t^*y	$\log_2(y)$	Кол-во транз., расч. по урав. регрессии тыс.шт., $U_{\text{расч}}$
15.11.71	2,3000	26252	0,001	0	0	1,2	1,71
01.04.1974	4,5000	27120	0,869	0,76	1,89	2,17	3,97
01.01.1982	29,0000	29952	3,701	13,7	17,98	4,86	61,67
01.02.1982	134,0000	29983	3,732	13,93	26,37	7,07	63,55
17.10.1985	275,0000	31337	5,086	25,87	41,21	8,1	235,76
22.04.1991	900,0000	33350	7,099	50,4	69,67	9,81	1655,6
22.03.1993	3100,0000	34050	7,799	60,82	90,45	11,6	3260,71
07.05.1997	7500,0000	35557	9,306	86,6	119,79	12,87	14028,72
26.02.1999	9500,0000	36217	9,966	99,32	131,69	13,21	26579,89
22.03.1999	19000,0000	36241	9,990	99,8	141,99	14,21	27204,79
20.11.2000	42000	36850	10,599	112,34	162,78	15,36	49060,82
12.03.2002	108000	37327	11,076	122,68	185,2	16,72	77860,5
08.07.2002	220000	37445	11,194	125,31	198,66	17,75	87284,5
30.06.2003	410000	37802	11,551	133,43	215,37	18,65	123327,1
01.11.2006	291000	39022	12,771	163,1	231,8	18,15	401863,33
01.02.2006	410000	38749	12,498	156,2	233,03	18,65	308517,41
01.04.2006	376000	38808	12,557	157,68	232,56	18,52	326655,25
01.01.2008	820000	39448	13,197	174,16	259,26	19,65	607036,51
01.04.2008	410000	39539	13,288	176,57	247,76	18,65	662950,42
01.11.2008	731000	39753	13,502	182,3	263,01	19,48	815586,03
Эмпирические данные			179,78	1954,95	2870,48	266,67	Расчетные данные
			S1	S3	S4	S2	
a		0,78					
b		1,4					
N0		1,7130489619					
T		0,716					

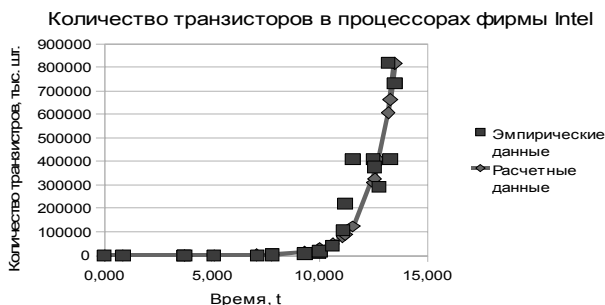


Рис. 1. Количество транзисторов в процессорах фирмы Intel по эмпирическим и расчетным данным

Таблица 3

Зависимость константы скорости разложения этана от температуры

t, C	550	560	570	580	590	600	610	620	630
$k_{\text{экс}} \cdot 10^{-5}, c^{-1}$	2,5	4,7	8,2	12,3	23,1	35,0	57,6	92,4	141,5

ложить учащимся построить график, показанный на рис. 1.

В качестве задания для самостоятельного решения учащимся можно предложить нахождение параметров в уравнении Аррениуса из области химической кинетики. Постановка задачи в этом случае имеет следующий вид. Используя экспериментальные значения константы скорости термического разложения этана (табл. 3), мето-

дом наименьших квадратов найти значения параметров k_0 и E , входящих в уравнение Аррениуса (11).

$$k = k_0 \cdot e^{-\frac{E}{RT}}, \quad (11)$$

где E — энергия активации, Дж/моль; T — температура, К; $R = 8,135$ — универсальная газовая постоянная, Дж/(моль · К).

В. А. Кудинов,

канд. пед. наук, доцент кафедры программного обеспечения и администрирования информационных систем Курского государственного университета

МЕТОДИЧЕСКИЕ ОСНОВЫ ИСПОЛЬЗОВАНИЯ КОРПОРАТИВНЫХ ПОРТАЛОВ УПРАВЛЕНИЯ ЗНАНИЯМИ ДЛЯ ОРГАНИЗАЦИИ АДАПТИВНОГО ОБУЧЕНИЯ

Эффективность использования корпоративного портала управления знаниями (КПУЗ) в учебном процессе определяется эффективностью (производительностью) учебной компьютерной технологии, которая, как подсистема КПУЗ, представляет собой, с одной стороны, совокупность знаний о способах организации образовательного процесса с использованием КПУЗ, а с другой стороны — сам процесс обучения, при котором происходит изменение качественных характеристик, интеллектуальной и эмоционально-волевой сфер личности обучаемого.

Перечислим основные качества, которыми обладают КПУЗ, с методической точки зрения.

1. КПУЗ позволяет осуществить переход от обучения, управляемого только ком-

пьютером, к обучению, управляемому как компьютером, так и обучаемым. Обучаемый становится активным партнером, способным не только отвечать на вопросы КПУЗ, но и задавать ему свои вопросы, на которые КПУЗ либо отвечает сразу, либо подводит пользователя к правильному ответу. Таким образом, диалог обучаемого с КПУЗ всё более приближается к живому диалогу учителя с учеником. Благодаря этой особенности КПУЗ появляется возможность реализовывать уникальные, наиболее подходящие каждому конкретному пользователю обучающие курсы, тем самым всё больше приближаясь к индивидуализации обучения.

2. КПУЗ способен самостоятельно решать те задачи (отвечать на те вопросы), решению которых он должен научить обучаемого.

Благодаря этому появляется возможность:

- более качественного консультирования;
- более глубокого анализа ответов обучаемого при тестировании и существенного упрощения подготовки тестовых задач.

3. В КПУЗ появляется возможность более полного анализа усвоения материала и психофизического состояния обучаемого и, как следствие, адаптации к индивидуальным особенностям обучаемого.

4. В КПУЗ более эффективно осуществляется процесс управления обучением.

5. КПУЗ имеет более высокий уровень интерфейса, приближенный к естественному языковому уровню, что значительно облегчает для пользователя процесс диалога с компьютером.

Таким образом, КПУЗ представляет собой сложную иерархическую систему, состоящую из совокупности взаимодействующих между собой подсистем, каждая из которых решает некоторый определенный класс задач с целью обеспечения оптимального режима обучения с помощью компьютера.

При организации обучения с использованием КПУЗ используется подход, реализующий принципы адаптивного программируемого обучения, позволяющий осуществлять в Intranet/Internet как доставку учебно-методической информации (УМИ), так и интенсивное удаленное интерактивное, в том числе адаптивное, обучение, повысить эффективность и надежность таких систем в целом. При этом процесс обучения (работы обучаемого с объектами УМИ) рассматривается как дискретный процесс, характеризующийся некоторыми устойчивыми состояниями системы. На каждом шаге работы с объектам УМИ обучаемый получает от КПУЗ обучающее воздействие — некоторый объем учебного материала, представленный в виде совокупности текстовой (гипертекстовой), статической графической, анимированной графической- и видеoinформации, а также аудиоданных. При этом если материал воспринят обучаемым, то осуществляется переход в новое устойчивое состояние. Если результат обучающего воздействия оказался отрицательным, то выполняется переход к изучению дополнительного учебного материала или возврат к учебному материалу, представленному другими объектами УМИ. Обработка УМИ, хранение которой осуществляется в базе знаний КПУЗ в виде большого числа фрагментов, имеющих законченное логико-смысловое содержание, с одной стороны, и минимально возможный объем хранения — с другой, выполняется на основе объектной модели [1]. Как и в объектно-ориентированном программировании, где сложные программные объекты

состоят из более мелких и более простых, так и в предлагаемой нами технологии хранения и организации УМИ учебная информация построена по принципу инкапсуляции.

Для размещения в базе данных вся УМИ представляется в виде объектов. В КПУЗ с точки зрения эффективности хранения и последующей обработки предложено выделить объекты различных уровней.

Объектом первого уровня является минимальная единица учебно-методической информации, обусловленная возможностью физического хранения в базе знаний КПУЗ. Основным (информативным) полям объектов первого уровня соответствуют текстовый фрагмент, изображение (рисунок), таблица, звуковой фрагмент, видеофрагмент, программа и т. д. Дополнительными информационными полями объектов являются поля, содержащие информацию о принадлежности к той или иной области знаний, периоду обучения, ключевые слова для поиска информации, которые описывают принадлежность объекта, как части технологии хранения и обработки УМИ.

Объект второго уровня приобретает уже собственные, свойственные ему методы: сборку учебно-методической информации, сборку тестовых заданий, сборку содержания, сборку списка используемой литературы и др.

Объекты последующих уровней строятся из объектов более низких уровней, наследуя их свойства и методы.

Одной из важных отличительных особенностей предлагаемого подхода является то, что любой минимальный фрагмент УМИ содержит в себе не только учебную информацию, но и блок контроля его усвоения, представляемый серией контрольных вопросов. Данный блок, с одной стороны, позволяет оценить качество усвоения данного учебного материала, с другой стороны, обеспечивает возможность предоставления этого фрагмента УМИ обучаемому, если на этапе предварительного тестирования удастся выяснить его уровень знаний по данной тематике.

Для эффективной реализации учебного процесса необходимо разработать различные учебные объекты (УО), что позволит выдавать обучаемому наиболее подходящий набор УО, используя значения параметров модели обучаемого [2].

С точки зрения организации и содержания учебного процесса УО — это базовая учебная единица, объединяющая различные виды и формы обучения и ориентированная на изменение конкретных способностей обучаемого от «незнания» к «знанию».

Разработка УО призвана:

- устранить дублирование, временные и логические разрывы между различ-

ными дисциплинами, видами и формами обучения, усилить связи между отдельными предметами;

- повысить качество обучения (преподавания и восприятия учебного материала обучаемыми);
- улучшить эффективность самостоятельной работы обучающихся.

УО характеризуется семантической самостоятельностью и самодостаточностью и представляется в наглядной форме (текстовой, графической, фото, видео, аудио).

Семантическая самостоятельность подразумевает четкие контуры предмета изучения. Самодостаточность предполагает, что УО содержит только необходимые и достаточные сведения, позволяющие полностью раскрыть содержание изучаемого предмета.

Цель любого УО — вполне конкретное обогащение системы знаний, навыков, умений и/или представлений обучаемого. Таким образом, для освоения понятий курса необходимо разработать УО, как содержащие теоретический учебный материал, так и включающие упражнения для его практического усвоения. В то же время при разработке УО следует предусмотреть возможность его повторного (многократного) использования в рамках различных курсов. С этой точки зрения наиболее значимым является последний уровень иерархии УО — отдельное понятие темы.

Изложение нового понятия обычно включает четыре этапа: дается определение понятия, показываются основные примеры его использования, объясняется его строение и связь с другими понятиями темы, излагаются правила использования данного понятия. Это необходимо учитывать при разработке УО.

Учебные объекты разделяются на две основные группы: информационные и задачные. Информационный УО предназначен для изучения учебного материала и может включать различные виды информации, изложенной с разной степенью детальности. Задачный УО предназначен для практического усвоения учебного материала путем выполнения некоторой задачи.

Для наиболее эффективной организации учебного процесса целесообразно использование модели обучаемого, хранящей в себе всю необходимую информацию [3], что облегчает выбор УО для конкретного обучаемого на определенном шаге его работы с системой.

Общая информация (специальность, учебная программа) влияет на тип УО (пример, строение, определение, правило), т. е. перед началом обучения преподаватель указывает, на каком уровне должен быть изучен предмет для конкретной специальности:

- ознакомление — обучаемому предлагаются общие сведения по изучаемой теме;

- воспроизведение — означает, что обучаемый способен анализировать различные действия и, возможно, их результат после изучения основных концепций данного предмета;
- навыки — показывает, что обучаемый может применить полученные знания, выполняя задания некоторой группы;
- трансформация — предполагает способность обучаемого принимать решения при выполнении незнакомых задач, используя полученные знания.

Детальность пояснений и комментариев, а также примеры напрямую зависят от общего уровня подготовки и текущей работы обучаемого. Для того чтобы определить, что отображать — пример или пояснение, учитываются психологические характеристики, а именно ориентированность (на себя, на задачу, на сотрудничество).

Адаптация реализуется при помощи сценария диалога с учетом модели обучаемого и модели учебного материала. Могут быть использованы три способа создания сценария:

- 1) «полностью преподаватель» — преподаватель создает один или несколько сценариев, которые хранятся в базе знаний;
- 2) «частично система» — преподаватель включает необходимые, по его мнению, УО, а система дополняет сценарий в зависимости от работы обучаемого и его характеристик;
- 3) «полностью система» — агенты системы сами определяют, какой и когда УО отобразить, основываясь на значениях параметров модели обучаемого и модели учебного материала.

Модель учебного материала отображается в виде графа, вершинами которого являются УО. Таким образом, при любом способе создания сценария диалога процесс обучения формируется динамически, т. е. для каждого обучаемого строится свой путь через граф модели учебного материала, в целом, и через структуру отдельного УО в частности.

Литература

1. Кабальнов Ю. С., Тархов С. В., Миновцов Ш. М. Способы хранения и генерации учебных курсов в информационно-обучающей среде, функционирующей на базе технологии WWW // Материалы научно-практической конференции «Информационные недра Кузбасса», посвященной 60-летию Кемеровской области. Кемерово, 2003.
2. Learning objects: What? // Center for International Education. http://www.uwm.edu/Dept/CIE/AOP/LO_what.html
3. Zaitseva L., Boule C. Learning systems in professional training. Workshop «Industry meets research» within the conference Interactive Computer Aided Learning ICL 2005 Villach, Austria 28 — 30 September 2005.

Е. В. Юрина,

*ст. преподаватель кафедры прикладной информатики
Вятского государственного гуманитарного университета*

КОГНИТИВНО-ПРАГМАТИЧЕСКИЙ ПОДХОД В ОБУЧЕНИИ ИНФОРМАТИКОВ-ЭКОНОМИСТОВ РАЗРАБОТКЕ ПРИКЛАДНЫХ РЕШЕНИЙ В СИСТЕМЕ «1С:ПРЕДПРИЯТИЕ»

Главной целью современного инновационного высшего образования является овладение такими базовыми компетенциями, которые позволят приобретать знания самостоятельно. Особенно актуально это для обучения информационно-коммуникационным технологиям, так как постоянные обновления и модификации программных продуктов не позволяют обучать действиям с конкретными пунктами меню, кнопками, и прочими элементами интерфейса. Именно поэтому необходимо искать и внедрять новые эффективные подходы к изучению информационно-коммуникационных технологий в вузе.

В теории и практике преподавания информационно-коммуникационных технологий сложился положительный опыт разработки и применения эффективных подходов. В частности можно отметить следующие: лично-ориентированный подход, целью которого является развитие индивидуальности ученика; деятельностный подход, подразумевающий активное усвоение материала при обучении, и многие другие.

Остановимся на рассмотрении когнитивно-прагматического подхода, позволяющего при помощи специально разработанной системы заданий, во-первых, развивать «персональный познавательный стиль» студента; во-вторых, побуждать студента к самостоятельным умозаключениям по поводу достижения поставленных целей и задач познавательной деятельности на занятии, по поводу выполнения конкретного задания или системы заданий; в-третьих, создавать новый продукт в результате деятельности студента, будь то разработанная и внедренная на предприятии автоматизированная информационная система, программа, документ или система документов, проект, презентация, творческая работа на тему и т. д., т. е. то, что сделано, сотворено самим студентом. Кроме того, необходимым условием при реализации когнитивно-прагматического подхода должна быть связь познавательной деятельности студента с его будущей профессиональной деятельностью.

При разработке системы заданий необходимо учитывать принципы и правила когнитивно-прагматического подхода: разработать систему входного и выходного мо-

нитинга студентов; содержание обучения структурировать с выделением декларативной и процедурной информации, определить последовательность их изучения, в декларативную информацию включить научные основы информационных технологий, как связующее звено между теоретической информатикой и прикладными технологиями, а в процедурную информацию включить весь технологический процесс решения задачи от ее постановки, через формализацию и моделирование до получения конкретного результата; каждый практикум должен являться интеллектуально насыщенным, с присутствием постоянного самоконтроля интеллектуальной деятельности студента, а также постоянного вызова студента к самостоятельным умозаключениям по поводу достижения поставленных целей и задач познавательной деятельности на занятии, по поводу выполнения конкретного задания или системы заданий; студент должен являться активным субъектом познания на занятии, осознавать процесс обучения, создавать новое; задания необходимо связывать с его будущей профессиональной деятельностью, что позволит достичь проявления личной заинтересованности каждого студента.

Пример раздаточного материала, основанного на когнитивно-прагматическом подходе, для проведения у информатиков-экономистов первого практического занятия по разработке прикладных решений в системе «1С:Предприятие» рассмотрим ниже.

Общая проблемная ситуация (постановка задачи).

Вы работаете разработчиком прикладных решений на основе платформы «1С:Предприятие 8». Вам поступил заказ на автоматизацию деятельности фирмы ООО «Мастер» по оказанию услуг ремонта бытовой техники. Вам предстоит разработать автоматизированную информационную систему (АИС) для работы этой фирмы.

Формализация и моделирование.

В конфигурации потребуется создать следующие объекты: справочники, документа, регистры накопления, отчеты; оформить меню, добавить панели инструментов, создать пользователей. Приступаем к созданию модели — новой АИС, которая должна от-

ражать существенные стороны деятельности фирмы по оказанию услуг ремонта бытовой техники.

Модуль 1. Тема «Создание справочников».

Основополагающий вопрос: как спроектировать и создать конфигурацию в системе «1С:Предприятие 8», которая автоматизировала бы деятельность компаний?

Вопрос по теме модуля: как и какие справочники можно создать для АИС фирмы, которая оказывает услуги по ремонту бытовой техники?

Блоки модуля 1.

Блок 1. Начальные настройки. Отладка конфигурации.

Задание 1.

- Для выполнения заданий создайте базу с пустой конфигурацией, а не базу, созданную из шаблона. Запустите программу в режиме конфигуратора.
- Объясните, как вы создали пустую базу и запустили базу в режиме конфигуратора.

Задание 2.

- С помощью какого пункта меню можно запустить отладку программы? Проверьте.

Блок 2. Создание справочников.

Задание 3 (подробно см. таблицу).

- Придумайте наименования справочников, которые, по вашему мнению, должны присутствовать в конфигурации для фирмы, оказывающей услуги по ремонту бытовой техники. Объясните назначение и состав каждого из предложенных вами справочников.

Задание 4.

- Откройте конфигурацию. Объясните, как вы это сделали.
- Исследуйте, как создать справочник. Запишите ваше предположение. Проверьте правильность вашего предположения путем создания наиболее простого по составу из предложенных вами справочников.

Задание 5.

- Какую длину кода и длину наименования можно задавать для справочника?
- Может ли тип кода данных быть строкой? В чем отличие строкового кода от числового?
- Создайте справочник Клиенты (он будет хранить наименования клиентов фирмы).
- Каким образом можно протестировать созданные справочники?

- Протестируйте, добавив в режиме «1С:Предприятие» в справочник «Клиенты троих»: Иванов Михаил Юрьевич, Роман, Спиридонова Галина.

Задание 6.

- Какой из предложенных вами справочников может содержать табличную часть? Какая информация будет там храниться?

- На какой вкладке при создании справочника можно добавить табличную часть?

- Создайте справочник Сотрудники для хранения ФИО сотрудника и информации о его прошлой трудовой деятельности. Трудовую деятельность оформите в виде табличной части, включающей организацию (как вы думаете какого типа и какой длины будет это поле? _____), начало работы (можно ли использовать пробел в записи имени? _____ какого типа и состава будет это поле? _____), окончание работы, должность. Какое использование в свойствах табличной части вы установили: для элемента, для группы или для группы и элемента?

- На какой вкладке можно установить редактирование справочника двумя способами? Настройте.

Задание 7.

- Создайте справочник Номенклатура (для хранения информации об услугах, которые оказывает фирма), с наличием иерархической структуры: две основные группы — Услуги, Материалы, которые также будут разбиты на подгруппы.
- Объясните, на какой вкладке вы установили параметры иерархического справочника и как вы это сделали.
- Какую длину наименования вы построили? Почему?

Задание 8.

- Создайте справочник Склады (для хранения информации о складах). Содержит один предопределенный элемент — склад «Основной», на который поступают все материалы.
- Какой элемент называется предопределенным? _____ Если затрудняетесь ответить, воспользуйтесь помощью программы.
- Объясните, на какой вкладке и как вы установили предопределенность элемента.

Таблица

Оценка заданий в соответствии с когнитивно-прагматическим подходом

Задание	Критерии оценки заданий		
	Развитие студента	Результат самостоятельной деятельности студента	Связь с будущей профессиональной деятельностью студента
Общее задание на практикум: <i>разработайте АИС для работы фирмы, оказывающей услуги по ремонту бытовой техники</i>	Развиваются <i>стили</i> (по М. А. Холодной) <i>кодирования информации</i> : предметно-практический, визуальный, словесно-речевой, сенсорно-эмоциональный; <i>стили переработки информации (когнитивные)</i> : полнезависимость, абстрактность, аналитичность и синтетичность, широкая категоризация, гибкий познавательный контроль, сканирующий контроль, рефлексивный, заострение, когнитивная сложность; <i>стили постановки и решения проблем (интеллектуальные)</i> : адаптивный (исполнительский), эвристический, исследовательский, инновационный; <i>стили познавательного отношения к миру (эпистемологические)</i> : эмпирический, конструктивно-технический, рационалистический рефлексивно-медитативный [3]	1. Разработанная в «1С:Предприятие 8» АИС. Созданы следующие объекты: справочники, документы, регистры накопления, отчеты; оформлено меню, добавлены панели инструментов, введены пользователи, задана их аутентификация. 2. Ответы на вопросы (в письменной форме). 3. Рефлексия своей деятельности (в форме устных ответов на вопросы преподавателя)	Такие же АИС информатики-экономисты будут разрабатывать и в выпускной квалификационной работе, и на производственной практике, и в будущей профессиональной деятельности. Студент «анализирует, прогнозирует, моделирует и создает информационные процессы и технологии в рамках профессионально-ориентированных информационных систем» [1] Задействованные объекты профессиональной деятельности информатика: информационные процессы, функциональные процессы, база данных, ресурсы — в организации, характерной для предметной области «Экономика»; профессионально-ориентированная информационная система. Задействованные виды профессиональной деятельности информатика-экономиста — организационно-управленческая, проектно-технологическая, экспериментально-исследовательская, консалтинговая, аналитическая, эксплуатационная [1]
Задание 3. <i>Придумайте наименования справочников, которые, по вашему мнению, должны присутствовать в конфигурации для фирмы, которая оказывает услуги по ремонту бытовой техники. Объясните назначение и состав каждого из предложенных вами справочников</i>	Развиваются <i>стили кодирования информации</i> : предметно-практический, визуальный, словесно-речевой; <i>стили переработки информации (когнитивные)</i> : полнезависимость, аналитичность и синтетичность, широкая категоризация, гибкий познавательный контроль, сканирующий контроль, рефлексивный; <i>стили постановки и решения проблем (интеллектуальные)</i> : эвристический, исследовательский, инновационный; <i>стили познавательного отношения к миру (эпистемологические)</i> : конструктивно-технический, рационалистический рефлексивно-медитативный	Самостоятельные суждения: наименования справочников создаваемой АИС с описанием их назначения и состава (в письменной форме)	Студент анализирует, прогнозирует, моделирует новую АИС. Задействованные объекты профессиональной деятельности информатика: информационные процессы (сбор, хранение), функциональные процессы (хранение сведений о множестве однородных данных), ресурсы — работники, контрагенты, номенклатура и прочее; профессиональноориентированная информационная система. Задействованные виды профессиональной деятельности информатика-экономиста — организационно-управленческая, проектно-технологическая, экспериментально-исследовательская, аналитическая

*Резюме модуля 1.**Ключевые моменты:*

- для успешного функционирования разрабатываемой АИС необходимы справочники для хранения списка однородных объектов предметной области;

- исследования показали, что справочники могут иметь иерархическую структуру разной степени сложности, разную длину и тип кода, длину наименования; можно создавать справочники с табличной частью, предопределенные справочники.

Вопросы для самоконтроля:

1. Справочники какой структуры можно создавать в конфигураторе «1С:Предприятие 8»?

2. Как создать пустую базу?

3. Как создать справочник иерархической структуры с табличной частью?

4. Как протестировать конфигурацию?

При выполнении заданий студенты в письменной форме отвечают на поставленные вопросы, имеет место обсуждение сложных вопросов в парах, в группе, с преподавателем. Происходит постоянный самоконтроль и контроль со стороны преподавателя в плане проверки ответов на вопросы и этапов работы по созданию конфигурации в программе.

Для проверки соответствия предложенного занятия когнитивно-прагматическому

подходу необходимо проанализировать задания с позиции следующих критериев: развития студента, результата его самостоятельной деятельности, связи с будущей профессиональной деятельностью. В таблице представлены примеры оценки заданий с точки зрения реализации в них когнитивно-прагматического подхода.

Таким образом, применяя когнитивно-прагматический подход при обучении информационно-коммуникационным технологиям, в частности при обучении разработке прикладных решений в системе «1С:Предприятие», можно и нужно достигать получения конкретных результатов деятельности каждого студента на занятии, приближения учебной деятельности к будущей профессиональной.

М. И. Шутикова,

доктор пед. наук, профессор, Череповецкий государственный университет,

И. А. Чеснокова,

Череповецкий государственный университет

О МОДЕЛЯХ УЧЕБНОГО ПРОЦЕССА В СИСТЕМЕ ЗАЧЕТНЫХ ЕДИНИЦ

Развитие общества в современных условиях характеризуется проникновением информационных технологий во все сферы человеческой деятельности. Мировые процессы интеграции стран как в областях экономического, политического сотрудничества, так и в образовательной сфере требуют единых стандартов, средств, методов профессиональной подготовки специалистов и, прежде всего, формирования информационно-культурной личности, интегрированной в современное сообщество.

В сентябре 2003 г. на Берлинской конференции министров образования стран — участниц Болонского процесса Российская Федерация в лице министра образования РФ В. М. Филиппова поставила свою подпись под Болонской декларацией, тем самым обязавшись в ближайшем будущем воплотить в жизнь основные принципы Болонского процесса.

Болонский процесс представляет собой попытку создания единого европейского образовательного пространства, конкурентоспособного в условиях цивилизации начала XXI в. по отношению к образовательным пространствам Америки, Азии и иных регионов планеты. Единое пространство подразумевает взаимную прозрачность образовательной практики разных стран и вузов по отношению к друг другу.

Напомним, что *Болонская декларация содержит следующие ключевые положения:*

- Принятие системы четких и сопоставимых степеней, в том числе через внедрение приложения к диплому, для того чтобы выпускники могли эффективно использовать свои квалификации, способности и навыки во всем европейском пространстве высшего образования.
- Введение двухциклического обучения: бакалавриат и магистратура. Первый цикл длится не менее трех лет. Второй должен вести к получению степени магистра или степени доктора.
- Внедрение европейской системы зачетных единиц (системы кредитов). За основу предлагается принять *Европейскую систему переводных зачетных единиц (European Credit Transfer and Accumulation System — ECTS)*, сделав ее накопительной системой, способной работать в рамках концепции «обучение в течение всей жизни». ECTS призвана обеспечить функции накопления и трансфера зачетных единиц. Совместно с взаимно признаваемыми системами обеспечения качества эта мера облегчит доступ студентов на европейский рынок труда.

- Развитие и стимулирование мобильности студентов, преподавателей и административных работников образования (на базе выполнения предыдущих пунктов). Этого можно добиться, устраняя все преграды для свободного передвижения студентов, преподавателей, исследователей и административного персонала. Особое значение придается социальному аспекту мобильности.
- Развитие европейского сотрудничества в обеспечении качества профессионального образования с целью разработки сопоставимых критериев и методологий.

Упомянутая выше Европейская система взаимозачета кредитов появилась на свет еще до начала Болонского процесса, в 1989 г. Она разрабатывалась для того, чтобы помочь руководству вузов понять и юридически грамотно отразить в документах об образовании результаты обучения их студентов за рубежом. Добиться этого удалось с помощью *введения особых зачетных единиц — кредитов*, начисляемых за освоение отдельных частей образовательной программы. Заключая двусторонний договор о мобильности, вузы брали на себя обязательство признавать кредиты, полученные студентом во время обучения в университете-партнере.

Уже к 2000 г. ECTS получила распространение более чем в 30 странах Европы; количество применяющих ее учебных заведений перевалило за тысячу. После подписания Болонской декларации стало ясно, что данная система достаточно хорошо соответствует идеологии проводимой в России образовательной реформы, и ее решено было использовать при составлении инновационных образовательных программ.

ECTS рекомендует оценивать суммарную трудоемкость дисциплин одного учебного года в 60 кредитов. Исходя из этого, в семестр студент должен заработать 30 кредитов. Кредиты ECTS могут быть зачтены только после успешного выполнения всей установленной учебной работы и надлежащей оценки результатов обучения. Успешность студента определяется с помощью местной (национальной) шкалы оценок.

Со стороны советской, а затем и российской высшей школы к ECTS — системе взаимозачета кредитов проявлялся постоянный интерес. В связи с подписанием Россией Болонской декларации возникла проблема практического внедрения ECTS, и сегодня в ряде отечественных вузов проводится эксперимент по переходу на систему зачетных единиц [1].

В настоящее время в российской высшей школе разработано несколько методик расчетов трудоемкости основных образовательных программ высшего профессионального

образования в зачетных единицах. В основном они отталкиваются от принципов, предложенных в Информационном письме Минобрнауки России от 28 ноября 2002 г. № 14-52-988ин/13. В этом письме «при расчетах трудоемкости основных образовательных программ высшего профессионального образования в зачетных единицах» предлагается исходить из следующего:

- 1 зачетная единица соответствует 36 академическим часам общей трудоемкости продолжительностью по 45 минут (или 27 астрономическим часам);
- максимальный объем учебной нагрузки студента в неделю составляет 54 академических часа, т. е. 1,5 зачетной единицы;
- расчет трудоемкости дисциплины в зачетных единицах производится исходя из деления ее трудоемкости в академических часах на 36 с округлением до целых чисел или до 0,5 по установленным правилам. Зачет по дисциплине и трудоемкость курсовых проектов (работ) входят в общую трудоемкость дисциплины в зачетных единицах;
- одна неделя практики выражается 1,5 зачетной единицы;
- один семестровый экзамен выражается 1 зачетной единицей (3 дня подготовки и 1 день на экзамен);
- трудоемкость итоговой аттестации рассчитывается исходя из количества отведенных на нее недель: 1 неделя соответствует 1,5 зачетной единицы.

Вплоть до середины 1990-х гг. одной из принципиальных организационных особенностей отечественной системы высшего образования был специфический подход к определению трудоемкости образовательных программ: в российской высшей школе она исчислялась в аудиторных часах, тогда как в западных системах образования — в рабочих часах и включала все виды работ, выполнение которых необходимо при освоении той или иной образовательной программы, в том числе все так называемые контактные часы с преподавателем — лекции, семинарские (практические), лабораторные занятия, консультации, часы индивидуальных занятий под руководством преподавателя, а также все формы самостоятельной работы студента, будь то подготовка к аудиторным занятиям или написание реферата, эссе, выполнение курсовой или выпускной квалификационной работы, прохождение практики и стажировки, сдача экзаменов и зачетов и подготовка к ним [3].

Переход российской высшей школы к зачетным единицам происходит постепенно: вначале осуществляется переход от аудиторных часов к рабочим часам или часам общей трудоемкости при сохранении ака-

демического часа как единицы измерения трудоемкости образовательной программы, а затем вводятся зачетные единицы.

Сдерживающим фактором в переходе к использованию зачетных единиц является отсутствие подробных и ясных нормативных документов, а также разработанных на их основе и одобренных министерством рекомендаций и практических пособий. В результате вузы сами для себя разрабатывают определенные правила, по которым они готовятся работать и уже работают с зачетными единицами [2].

Первые шаги по внедрению в образовательную практику системы зачетных единиц доказали, что сложившаяся в России система подготовки специалистов (бакалавров, магистров) не может органично вписаться в формирующееся европейское образовательное пространство, не претерпев структурных изменений.

Нам видится, что в вузах необходимо разрабатывать *информационную систему для расчета зачетных единиц студентов и преподавателей на основе модульной технологии и системы зачетных единиц в соответствии с Федеральным государственным образовательным стандартом высшего профессионального образования.*

При разработке информационной системы (ИС) для расчета объемов учебной нагрузки студентов в зачетных единицах, опираясь на положения об организации учебного процесса в системе зачетных единиц вуза, необходимо учитывать, что ИС должна реализовывать следующее:

- по имеющейся базе данных студентов, преподавательского состава и списка предлагаемых дисциплин в соответствии со специальностью, курсом и базовым учебным планом, после выбора каждым студентом своего индивидуального курса обучения рассчитывать для него зачетные единицы;
- использовать данные о студентах, их выборе дисциплин, рассчитывать нагрузку преподавательского состава, как по федеральному компоненту (базовой части), так и в зависимости от вариативной части по выбору студентов (каждому преподавателю формируем объем нагрузки в зачетных единицах).

По действующим нормативам максимальная аудиторная нагрузка преподавателя — 900 часов. Исходя из деления академической нагрузки в часах на 36, получаем нормативную аудиторную нагрузку профессорско-преподавательского состава, равную 25 кредитам (зачетным единицам). Далее можно принять, что аудиторная годовая нагрузка распределяется в соответствии с должностями следующим образом:

- до 25 кредитов — для ППС в должности преподавателя и старшего преподавателя;

- до 20 кредитов — для ППС в должности доцента;
- до 15 кредитов — для ППС в должности профессора.

Электронная информационная система учебного процесса должна выполнять следующие функции:

- формирование учебного плана по специальности, включающего федеральный компонент (базовую часть), региональный компонент и дисциплины по выбору (вариативную часть) (с расчетом зачетных единиц по дисциплинам плана);
- возможность добавления новой дисциплины при изменении учебного плана;
- редактирование/удаление дисциплин при внесении изменений в учебный план (по часам и по зачетным единицам);
- формирование нагрузки по дисциплинам и в зачетных единицах по преподавателям (учет базовой и вариативной частей учебного плана);
- формирование индивидуальной учебной траектории студента, включая выбор студентом специальности в информационной системе института, курса и группы. При этом ему выделяется дисциплины федерального компонента с учетом итоговых кредитных сумм с разбивкой по дисциплинам. Студенту предоставляется возможность самостоятельно сформировать собственный учебный план на следующий учебный период, добавив дисциплины из вариативной части, включающей региональный учебный компонент и дисциплины по выбору. При добавлении каждой новой дисциплины рассчитываются сумма кредита по дисциплине и итоговые расчеты по кредитам.

При работе с базой данных предполагаются следующие возможности:

- добавление нового студента при его зачислении;
- удаление студента при его отчислении;
- формирование индивидуального учебного плана студента на основе выбора студентом вариативной части учебного плана;
- формирование электронной зачетной книжки (ЭЗК) студента и взаимодействие с ЭЗК как студента, так и преподавателя. ЭЗК связана с электронной базой учебного плана специальности, индивидуальным учебным планом студента, модульной структурой изучения дисциплин, преподавателями, рейтинговым учетом изучения дисциплин и переводом в зачетные единицы.

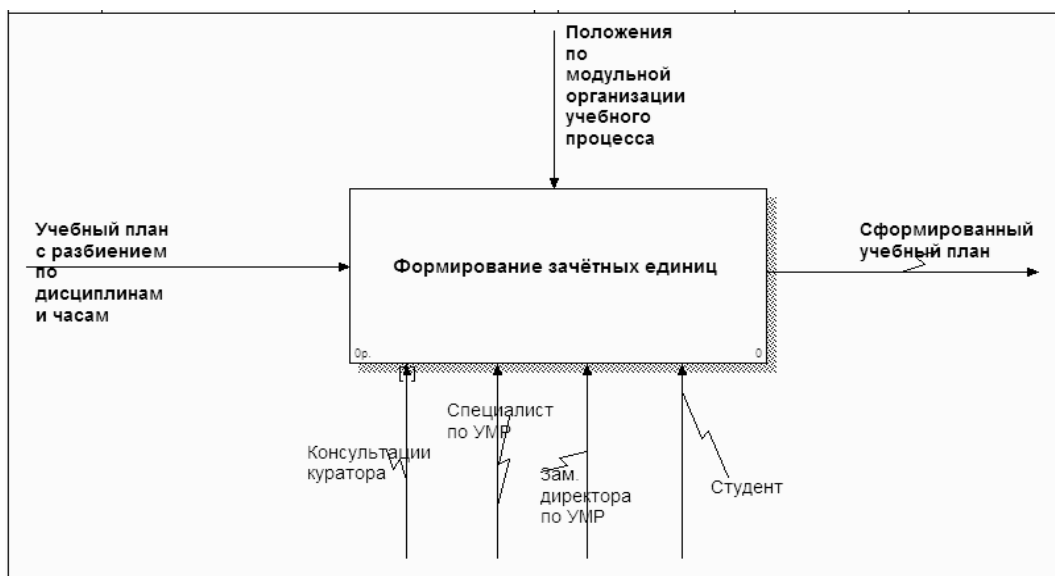


Рис. 1. Структурно-логическая модель учебной среды и алгоритм движения потоков данных

Процесс формирования зачетных единиц по учебному плану можно представить структурно-логической моделью учебной среды (рис. 1).

Для отражения взаимосвязей в модели формирования учебного плана на основе зачетных единиц с учетом индивидуального выбора дисциплин студентом (вариативная часть учебного плана) и опираясь на положения об организации учебного процесса в системе зачетных единиц вуза, предложена информационная модель формирования зачетных единиц студентов (рис. 2).

Информационная система кредитно-модульной организации учебного процесса, описанная в виде логической модели:

- активизирует разработку и внедрение образовательных стандартов вузов на основе ГОС ВПО РФ с учетом региональных особенностей, международных стандартов и требований общественных профессиональных организаций;
- стимулирует развитие различных технологий и форм обучения в вузе (очной, заочной, дистанционной, экстерната);

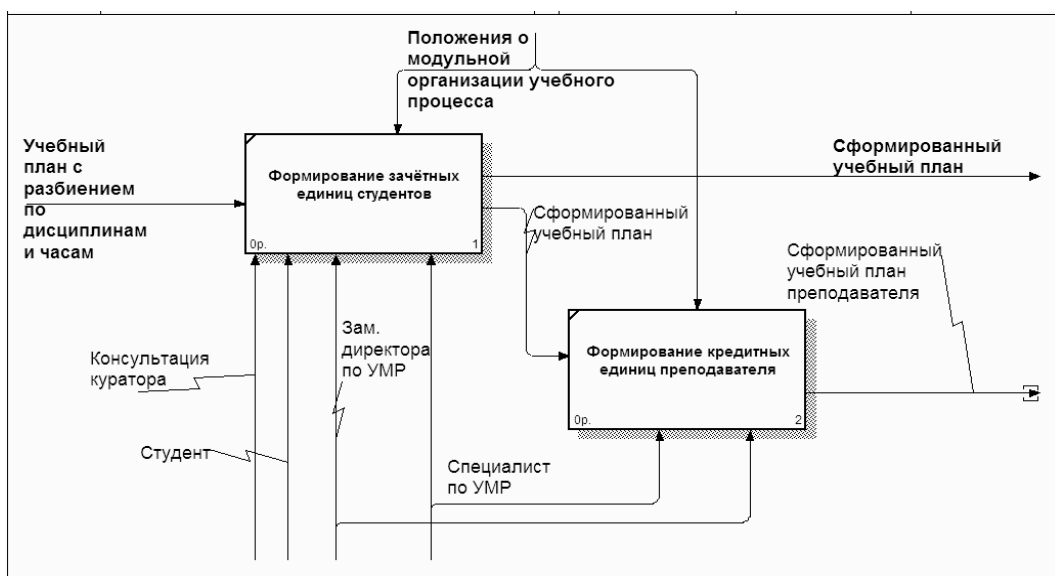


Рис. 2. Логическая модель взаимосвязи «студент — преподаватель — учебный план»

- либерализует учебный процесс и ориентирует его на студента, поощряя самостоятельную работу учащегося (деятельностный подход);
- способствует оптимальному планированию содержания и методического обеспечения курсов с учетом конечных результатов;
- стимулирует совершенствование процессов мониторинга и оценки качества освоения образовательных программ с использованием современных методов диагностики;
- способствует созданию в вузах современных систем менеджмента качества образовательных услуг и подготовки специалистов;

- содействует интеграции российской высшей школы в мировую систему высшего образования.

Литература

1. *Дегтярев Г., Насыров И., Крюков В.* Система зачетных единиц: предложения по внедрению в технических вузах // Высшее образование в России. 2007. № 6.
2. *Сазонов Б.* Академические часы, зачетные единицы и модели учебной нагрузки // Высшее образование в России. 2008. № 11.
3. *Сенашенко В., Кузнецов В., Кузнецова В.* Введение зачетных единиц как организационно-педагогическая проблема // Высшее образование в России. 2008. № 8.

Е. П. Шевчук,

преподаватель кафедры педагогики

Брестского государственного университета им. А. С. Пушкина, Республика Беларусь

ИНДИВИДУАЛЬНЫЙ ПОДХОД К ФОРМИРОВАНИЮ ПРОФЕССИОНАЛЬНОГО САМОСОЗНАНИЯ СТАРШЕКЛАССНИКОВ В СОВРЕМЕННОМ ИНФОРМАЦИОННОМ ОБЩЕСТВЕ

Концепция информационного общества была выдвинута в конце 1970-х гг. как теория постиндустриального варианта общественного развития. Основоположниками этой концепции считаются Д. Белл, Е. Масуда, О. Тоффлер. Фундаментальное значение для понимания информационного общества сыграли книги Ф. Махлоупа: «Знание, его создание и экономическое значение» (Knowledge: Its Creation, Distribution and Economic Significance), «Знание и производство знаний» (Knowledge and Knowledge Production), которые вышли соответственно в 1980 и 1982 гг. В них подчеркивалась важность информации и информационных технологий для развития глобальной экономики и всей человеческой жизни.

В условиях информационного общества выбор профессии, определяющий жизненный путь личности, становится особенно значимым и трудным. Старшекласснику, находящемуся в сложной ситуации выбора, необходима профессиональная психолого-педагогическая поддержка в осуществлении процесса познания, самоузнавания личностных качеств и соотнесения их с требованиями, предъявляемыми будущей сферой деятельности. По мере того как человек приобретает жизненный опыт, расширяет круг своего общения, осваивает новые способы поведения и решения жизненных

проблем, перед ним открываются новые стороны собственного «Я», он постигает жизненные смыслы, обогащающие его внутренний мир и составляющие базис его нравственной жизни. На основе представлений о себе личностью не только строится поведение, но и интерпретируется собственный опыт. Образ «Я» школьника можно считать внутренним фильтром, который определяет его мировосприятие, характер его отношений с миром и людьми, помогает в становлении способности осознавать себя целостной личностью, в развитии самопознания. Особенно важное значение этот процесс приобретает в информационном обществе, которое характеризуется динамикой информационных процессов и сложностью окружающих человека информационных систем.

Стимулирование самопознания учащихся в этих условиях возможно только в определенной среде, на основе нравственных критериев, утверждающих ценность работы над собой и обеспечивающих проявление лучших качеств личности.

Такая среда создается в процессе рефлексивно-сотворческой деятельности, основанной на взаимном принятии, уважении и доверительных отношениях в системах «учитель — ученик», «ученик — ученики»; содержательном межличностном общении

в коллективе старшеклассников, развивающем доминанту постоянного нравственного самосовершенствования.

В качестве педагогических механизмов стимулирования самопознания старшеклассников с целью развития профессионального самосознания нами были выделены:

- организация ценностно-смысловой, рефлексивно-сотворческой деятельности;
- организация психолого-педагогических тренингов и рефлексивно-деловых игр, развивающих навыки самоанализа, проектирования путей личностного роста.

Для решения профориентационных задач необходимо активизировать самопознание школьниками своих индивидуальных особенностей; учить соотносить представления о самом себе с мнением о требуемом уровне развития профессионально важных качеств в избранной сфере деятельности; развивать у учащихся стремление к саморазвитию. Всех этих аспектов мы старались придерживаться в работе *элективного курса «Профессиональный выбор»*.

Данный элективный курс нацелен на решение следующих задач:

- ознакомить старшеклассников с ориентировочной основой ценностно-смысловой, рефлексивно-сотворческой, проектировочной деятельности;
- предоставить возможность овладеть приемами освоения сложного процесса самопознания, духовного роста, самосовершенствования;
- стимулировать дифференцированное познание учащимися своего образа «Я» в различных аспектах;
- развивать позитивное самовосприятие и самоотношение;
- актуализировать потребность в постоянном личностном развитии и стремлении к самоактуализации.

Особое внимание нами было уделено нравственной составляющей профессионального самосознания старшеклассников, так как проблема развития нравственных качеств личности в настоящее время приобрела особую значимость. В процессе работы мы обращались к профориентационному потенциалу предметов гуманитарного цикла, обладающих большими возможностями в решении данной проблемы.

Основополагающим в определении нами подходов к работе явилось следующее положение, выдвинутое Д. Сьюпером: на разных стадиях развития можно управлять, способствуя формированию у индивида интересов и способностей, поддерживая его в стремлении «попробовать» реальной жизни и в развитии его Я-концепции; профессиональное развитие состоит в развитии и реализации Я-концепции [8].

Данная теория соответствует одной из главных целей предложенного нами элективного курса — содействовать формированию профессионального самосознания личности как ориентации на непрерывное, в течение всей жизни, стремление к максимальной нравственной и профессиональной самореализации, базирующейся на непреходящих жизненных ценностях. Вследствие этого нами были выделены следующие параметры, положенные в основу работы данного курса:

- информированность о мире профессий и умение соотнести полученную информацию со своими личностными особенностями; умение планировать свою профессиональную жизнь;
- стремление к максимальной жизненной и профессиональной самореализации;
- постоянная рефлексия нравственных ценностей и соотнесение их с возможностью нахождения полноценной реализации в трудовой деятельности.

В содержание первого параметра нами в настоящее время включены знания о путях приобретения профессии; требуемом уровне образования для различных профессий; требованиях, предъявляемых к современному профессионалу как работодателем, так и обществом; совершенствовании профессиональных навыков; построении карьеры. Речь идет не только о выборе профессии, но и о прогнозировании дальнейших жизненных этапов. Только человек, владеющий информацией о мире профессий в целом — его сферах, законах развития, а также специфическими знаниями о том, какие области мира профессий соответствуют интересам, способностям, знаниям личности, какие шаги необходимо сделать, чтобы осуществить свой выбор, может принимать обоснованные реальные решения.

Для эффeктивного самоопределения в рамках предложенного нами подхода целесообразно создать *информационную среду*, моделирующую различные аспекты процесса профессионального самоопределения. В этой среде каждый ученик должен чувствовать себя комфортно и развивать необходимые для будущей профессиональной деятельности качества, учитывая при этом особенности современного информационного общества. При этом должен соблюдаться «фундаментальный факт учения», понимаемого как процесс, осуществляемый самим индивидом при освоении новых знаний или новых ситуаций, сформулированный С. Пейпертом: любая вещь дается легко, если вам удастся ассимилировать ее в совокупность собственных моделей. Когда же это не получается, то что угодно может оказаться мучительно трудным... Чему может научиться индивид и как он будет учиться, зависит от того, какими моделями он овладел.

Всё это означает, что в процессе профессионального самоопределения необходимо создать такую информационную среду, которая:

- *дружественна* к ученику, т. е. снимает, по возможности, страхи и опасения по отношению к совершению ошибок. Учащийся должен быть уверен, что ошибка всегда может быть исправлена им самим или с помощью учителя. Ошибка может стать предметом обсуждения учителя с целью определения ее причины, но не основанием для наказания;
- *разумно требовательна*, т. е. учащиеся знают, что есть время для проявления свободы выражения и творчества (чаще всего на проблемных семинарах и при выполнении творческих работ) и есть время, когда следует действовать как «формальный исполнитель»;
- *достаточна для принятия обоснованного решения*, т. е. учащиеся приучаются к тому, что в любой момент самостоятельной работы у них есть вся необходимая информация, чтобы определить, как поступить дальше;
- *ориентирована на личность* учащегося, на активное освоение человеком способов познавательной деятельности;
- *помогает реализовать* познавательные потребности личности учащегося в зависимости от его познавательных способностей, обеспечивает возможность его самораскрытия;
- *располагает ученика к сотрудничеству* как с учителем, так и с товарищами. В частности, такая среда дает опыт

принятия коллективных решений, умения избегать конфликтов, что является важным качеством современного специалиста в любой области деятельности.

Суммируя вышесказанное, можно сказать, что информационная среда должна быть соответствующей основной мотивации деятельности индивида — стремлению к успеху и избежанию неудач.

Литература

1. *Абульханова-Славская К. А.* Стратегия жизни. М.: Мысль, 1991.
2. *Климов Е. А.* Развивающийся человек в мире профессий. Обнинск: Принтер, 1993.
3. *Кухарчук А. М., Седова Е. Л., Лях В. В.* Психодиагностика в профессиональном самоопределении учащихся. Минск: Бел. наука, 2000.
4. *Поляков С. Д.* Технологии воспитания. М.: ГИЦ ВЛАДОС, 2002.
5. *Пряжников Н. С.* Профессиональное и личностное самоопределение. (Методы активизации). Мозырь: Белый ветер, 1998.
6. *Пряжников Н. С.* Игры и методики для профессионального самоопределения старшеклассников. М.: Первое сентября, 2004.
7. *Романова Е. И.* 99 популярных профессий. Психологический анализ и профессиональные программы. СПб.: Питер, 2003.
8. *Хьелл Л., Зиглер Л.* Теории личности. (Основные положения, исследования и применение). СПб.: Питер Ком, 1998.
9. *Lea H. D., Leibowitz Z. B.* Adult Career Development: Concepts, Issues & Practicies. Alexandria: (Eds.), 1992.

Х. С. Мусаева,

ст. преподаватель кафедры языкознания университета «Юждаг», г. Дербент, Республика Дагестан

НЕКОТОРЫЕ ОСОБЕННОСТИ ОБУЧЕНИЯ РУССКОМУ ПРАВОПИСАНИЮ В НАЦИОНАЛЬНОЙ ШКОЛЕ В УСЛОВИЯХ ИНФОРМАЦИОННОГО ОБЩЕСТВА

Владение языком, обеспечивающее свободное общение в сфере общественной деятельности, науки, техники и культуры, предполагает умение не только говорить, выражать свои мысли, но и грамотно писать, а также эффективно использовать информационные технологии. Современное человеческое общество живет в период, характеризующийся небывалым увеличением информационных потоков. Это относится

как к экономике, так и образовательной сфере. На современном этапе развития общества грамотное письмо и использование информационных технологий являются удобным средством общения и занимают огромное место в жизни людей.

В условиях национальной школы выработка у учащихся прочных навыков правописания приобретает социальную значимость, поскольку соблюдение ими общепри-

нятых норм письменной речи способствует развитию и совершенствованию билингвизма. И здесь незаменима компьютерная технология, поскольку она дает возможность оптимизировать и рационализировать образовательную функцию за счет применения новых средств сбора, передачи и преобразования информации.

Актуальность вопросов информатизации всех сфер общественно-экономической жизни очевидна, поскольку сегодня возрастает потребность в разработке и применении эффективных и адекватных реальной действительности компьютерных программ и технологий. Реформа методов обучения повлекла за собой не только перестройку организации процесса автоматизации образовательной деятельности, но и распространение новых форм реализации этой деятельности.

Решающее значение для успешного формирования навыков грамотного письма могут иметь обучающие компьютерные программы. Вместе с тем успешное формиро-

вание навыков грамотного письма возможно при осознании основных теоретических положений орфографии.

Орфография, или правописание, как особая область языка, устанавливающая единые нормы написания слов и их форм, представляет собой весьма сложную систему. Сфера действия орфографии достаточно широка. Она регламентирует:

- буквенное обозначение звукового состава слова;
- написание прописных и строчных букв, слитные, дефисные и раздельные написания;
- перенос части слова с одной строки на другую;
- графические сокращения.

Важнейшим условием эффективности обучения русскому правописанию в азербайджанской школе является тщательный сопоставительный анализ орфографических систем русского и родного языков, который может быть осуществлен и с помощью информационных технологий.

А. Г. Гейн,

доктор пед. наук, профессор кафедры алгебры и дискретной математики Уральского государственного университета им. А. М. Горького,

Е. А. Кокшарова,

преподаватель кафедры математики и методики обучения математике Нижнетагильской государственной социально-педагогической академии

ИСПОЛЬЗОВАНИЕ ПЕДАГОГИЧЕСКОЙ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ОЦЕНКИ КАЧЕСТВА ОБУЧАЮЩИХ ТЕСТОВ

Современный уровень развития общества предъявляет повышенные требования к результатам работы специалистов в различных областях. Выполнение этих требований во многих случаях оказывается возможным лишь тогда, когда на вооружение берутся адекватные задачам информационные технологии. Среди них всё большую роль играют средства, разрабатываемые в тех или иных направлениях искусственного интеллекта, в частности применение экспертных систем для оценки качества принимаемых решений.

В сегодняшней системе образования обучающее воздействие на ученика состоит преимущественно в методически оправданном применении определенного дидактического материала. Формы такого применения могут быть довольно разнообразными. В настоящее время активно развивается такое направление, которое принято называть обучающим тестированием. Его мето-

дологическое отличие состоит в том, что не столько подвергается оценке конечный результат выполнения тестового задания, сколько отслеживается тот путь, который был избран учащимся для получения ответа. Тем самым снимается один из основных недостатков традиционного тестирования, по результатам которого нельзя быть уверенным в том, что правильный ответ был получен правильным путем, но и при получении неправильного ответа практически невозможно выявить, в чем состояла ошибка ученика, носит ли она принципиальный характер или была следствием недостаточной аккуратности. Обучающее тестирование позволяет сочетать в процедуре контроля качества обучения основные преимущества тестовой технологии (стандартизированность и объективность оценки) с возможностью анализировать уровень освоения учащимся тех умственных действий, которыми он должен обладать и которые он обязан уметь

использовать в достижении поставленных перед ним учебных целей (в рамках конкретного школьного предмета). Обучающее тестирование, по сути, является инновационной педагогической технологией [1].

Важным отличием тестовых технологий от иных технологий, применяемых в обучении, состоит в том, что можно оценить качество каждого отдельного теста, а не только всей совокупности используемых дидактических материалов, примененных в данной методике обучения. Для обучающих тестов параметры качества могут быть выявлены априорно, и это позволяет конструировать обучающие тесты с заранее заданным уровнем качества [1]. Однако составление и просчет параметрических характеристик обучающего теста и тем более определение направлений повышения его качества отнимают у преподавателя достаточно большое количество времени и сил, что подчас заставляет его отказаться от составления обучающего теста. Применение информационных технологий в форме педагогической экспертной системы позволяет в значительной степени разрешить эту проблему.

Под педагогической экспертной системой (ПЭС) мы понимаем интеллектуальную систему, в которую включены знания специалистов о некоторой конкретной педагогической области и которая в пределах данной области способна принимать верные, с точки зрения эксперта, решения. Важно, чтобы такая экспертная система могла явным образом аргументировать принимаемые ею экспертные решения.

Данный программный продукт применим в области образования, так как разработан специально для педагогов, имеющих представление об обучающем тестировании

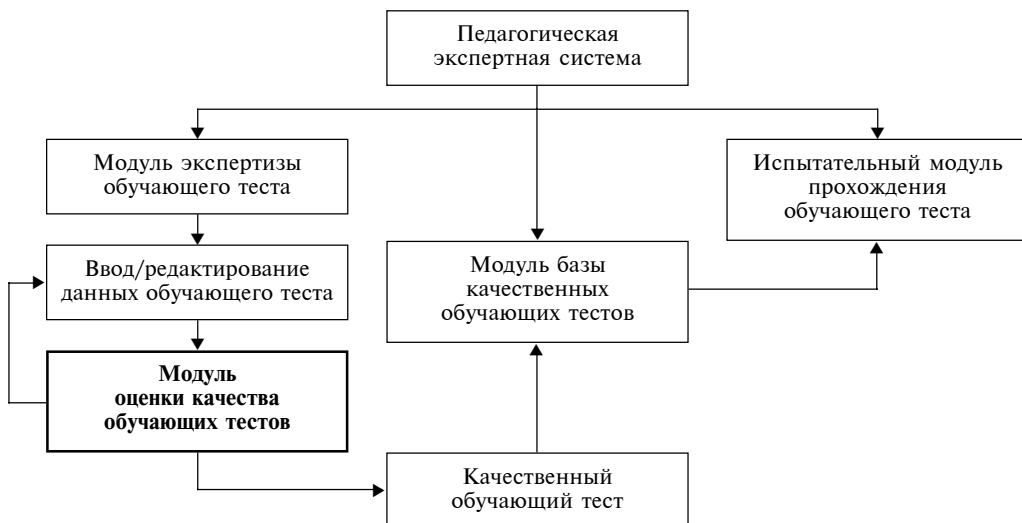
и использующих тесты в своей педагогической деятельности.

Программный продукт обладает простым и понятным интерфейсом, доступен в использовании как преподавателям, имеющим навыки работы с персональным компьютером, так и новичкам, удовлетворяет психолого-педагогическим и эргономическим требованиям, предъявляемым к программным педагогическим системам.

На основе проведенного нами анализа экспертных систем, используемых в сфере образования, была определена структура педагогической экспертной системы как совокупность следующих компонентов и модулей (см. схему).

В главном модуле оценки качества обучающих тестов осуществляется сопоставление значений параметров теста со шкалой критериев, отвечающих за валидность и надежность теста. При этом данные параметры подразделяются на две группы: первая группа — это параметры, отвечающие непосредственно за качество теста, вторая — это параметры, отвечающие за стратегию обучения. К первой группе относятся такие параметры, как уровень альтернативности теста минимальный, уровень альтернативности теста максимальный, степень неравномерности теста; ко второй — уровень сложности теста максимальный, уровень сложности теста минимальный.

На основании второй группы параметров можно судить о том, для какого уровня подготовки учащихся может применяться тот или иной тест и насколько отличается по уровню трудности один обучающий тест от другого. В базе тестов они располагаются в порядке возрастания минимального уровня сложности, а так как уровень сложности теста отвечает за адаптацию теста к уровню подготовленности учащегося, то по



Структура педагогической экспертной системы

численному значению данного параметра можно судить о пригодности теста для использования учащимися данного уровня подготовленности. В случае успешного прохождения теста начального уровня сложности можно переходить к следующему уровню и т. д. Таким образом, педагог, ориентируясь по данным показателям, может выстраивать индивидуальную стратегию обучения для каждого учащегося.

В свою очередь каждый тест, входящий в базу данных системы, состоит из заданий, которые можно разделить на два типа. Задания первого типа направлены на развитие умений думать, анализировать задачу, применять знания, т. е. на развитие аналитико-синтетических операций. Задания второго типа направлены на выявление и устранение пробелов в знаниях. Можно сказать, что задания второго типа реализуют стандартную контролируемую функцию. Тем самым без заданий первого типа тест перестает быть обучающим, он становится традиционным тестом контроля знаний, умений, навыков. Однако и без заданий второго типа обучающий тест также, по видимому, существовать не может. Ведь любой тест в качестве основы должен содержать решение некоторой задачи, имеющей определенный ответ, и, следовательно, от учащегося требуется выполнение действий, представляющих собой приме-

нение знаний в конкретной ситуации. Поэтому выявление характеристики, не только учитывающей структуру учебного процесса, но и отвечающей в обучающих тестах за направленность их на обучение, есть необходимое условие для определения качества обучающего теста как средства для обучения.

В качестве параметра, численно описывающего данную характеристику теста — направленность на обучение — естественно взять отношение числа вопросов первого типа к общему числу вопросов в тесте. Границы изменения этого параметра, дающие заданный уровень качества теста, определялись экспериментальным путем и были установлены от 0,3 до 0,6. Необходимо также определить критерий, по которому задание теста будет квалифицироваться как задание первого или второго типа [1].

Таким образом, благодаря разработанной ПЭС учителю достаточно удобно контролировать процесс обучения каждого учащегося и выстраивать индивидуальную стратегию обучения.

Литература

1. *Сенюгова Н. А.* Технология конструирования тестов учебной деятельности как средства оценивания результатов обучения: Дис. ... д-ра пед. наук. Киров: ВГГУ, 2006.

М. В. Епанчинцева,

учитель информатики средней общеобразовательной школы № 4, г. Нефтеюганск

ФОРМИРОВАНИЕ ПСИХИЧЕСКИХ ПРОЦЕССОВ ЧЕРЕЗ КОМПЛЕКС РАЗВИВАЮЩИХ ЗАДАЧ В ПРОПЕДЕВТИЧЕСКОМ КУРСЕ ИНФОРМАТИКИ

В теоретических и практических исследованиях, посвященных проблеме обучения, выделяют несколько типов задач развивающего характера:

- задачи на самостоятельное «открытие» нового;
- задачи на самостоятельный выбор известного способа решения и установление необходимых связей;
- проблемные задачи, решение которых предполагает хотя и управляемый учителем, но самостоятельный поиск еще неизвестных учащимся закономерностей, способов действия, правил.

Между системой обучения и ходом умственного развития учащихся существует тесная взаимосвязь, которая подчиняется определенным закономерностям. Поиск данных закономерностей в настоящее время од-

ной из центральных проблем педагогической психологии. Умение решать задачи познавательного характера является важным критерием сформированности мыслительных способностей младших школьников. Информатика как метанаука позволяет рассматривать задачи самых различных областей, поэтому их подбор должен способствовать общему развитию ребенка. Следовательно, задачи, рассматриваемые на занятиях по информатике должны быть развивающими.

Комплекс развивающих задач пропедевтического курса информатики должен основываться на знаниях учащихся по другим предметам и охватывать следующие виды деятельности:

- 1) операции мыслительной деятельности:
 - сравнение множеств по их мощности;
 - выделение подмножеств из множества по определенным признакам;

- исключение из множества элементов с не совпадающими с остальными элементами свойствами;
 - формирование множеств знаний об объектах, их поведенческой сущности;
 - использование различных моделей для представления множества знаний об объектах и др.;
- 2) учебную деятельность для формирования стиля алгоритмического мышления:
- исполнение алгоритма;
 - запись алгоритма в виде блок-схем, табличную запись алгоритма;
 - запись алгоритма в командах исполнителя;
 - запись алгоритма на языке программирования;
 - конструирование алгоритмов для получения результата;
 - анализ простейших алгоритмов;
 - рекурсивное представление алгоритмов;
- 3) деятельность по формированию практических навыков работы с компьютером:
- сборку объектов в множества;
 - рисование объектов, заполнение, дизайн;
 - работу с текстами, нотами, числовой информацией, интеллектуальными исполнителями и др.

Формирование у учащихся умения пользоваться приемом сравнения осуществляется в несколько этапов:

- выделение признаков или свойства одного объекта;
- выявление сходства между признаками трех, четырех и более объектов;
- установление сходства и различия между признаками двух объектов.

Формирование у детей логического приема сравнения необходимо начинать со II класса. В качестве объектов для сравнения нужно использовать предметы или рисунки с изображением предметов, хорошо знакомых детям, в которых они могли выделить те или иные признаки, опираясь на имеющиеся у них представления.

В основе приема *классификации* также лежит умение выделять признаки предметов и устанавливать между ними сходство и различие. При формировании приема сравнения дети решают задания на классификацию хорошо знакомых им предметов, например: «Убери лишний предмет...». Постепенно по мере изучения различных понятий по информатике задания на классификацию стали включать устройства компьютера, их назначение и т. д.

Уделяя развитию *анализа* особое внимание, необходимо учить детей не только выделять элементы того или иного объекта, его различные признаки или соединять элементы в единое целое, но и включать объект в новые связи, видеть его новые функции.

Формированию этого умения способствует:

- рассмотрение данного объекта с точки зрения различных понятий;
- постановка различных заданий к данному объекту.

Разрабатывая упражнения на развитие приема *обобщения*, необходимо помнить о том, что в начальном обучении наиболее часто применяется эмпирическое обобщение, при котором обобщенные знания являются результатом индуктивных умозаключений. В связи с этим необходимо решать следующие задачи:

1) рассмотреть как можно больше частных объектов, в которых повторяется та закономерность, которую ученики должны подметить;

2) варьировать виды частных объектов, то есть использовать предметные ситуации, схемы, таблицы, отражая в каждом виде объекта одну и ту же закономерность;

3) помогать детям словесно формулировать свои наблюдения с помощью наводящих вопросов, уточнять и корректировать те формулировки, которые они предлагают.

Формировать у младших школьников умение обобщать наблюдаемые факты индуктивным способом нужно через задания, при выполнении которых дети могут сделать неверные обобщения: «Сравните объекты, найдите общее и сделайте вывод».

При выполнении заданий на развитие *теоретического обобщения* дети должны осуществлять анализ данных, о каком-либо одном объекте или ситуации с целью выявления существенных внутренних связей. Эти связи фиксируются абстрактно и становятся той основой, на которой в дальнейшем выполняются конкретные действия.

Непременным условием полноценного интеллектуального развития является формирование у учащихся способности *доказывать те суждения*, которые они высказывают. Для сознательного выполнения дедуктивных умозаключений учащимися нужно провести подготовительную работу, направленную на усвоение вывода, закономерности, свойства в общем виде.

Систематическая работа по формированию психических процессов у учащихся способствует тому, что дети (хотя и не все одновременно и в одинаковой степени) начинают осознавать собственные умственные операции в процессе анализа, синтеза, сравнения, установления взаимосвязи между объектами, пользоваться ими при решении других подобных задач, а также переносить их в другие области знаний.

Постановка учебной задачи, ее совместное с учащимися решение, организация оценки найденного способа действия — это основные составляющие метода, который

соответствует целям и содержанию развивающего обучения. Средства и приемы постановки учебной задачи, организации ее решения и оценки будут отвечать своему назначению только в том случае, если они в максимальной степени учитывают реальный ход развертывающейся поисковой деятельности учащихся.

Использование на уроках информатики рабочих тетрадей для учеников с печатной основой позволяет организовать совместную и индивидуальную работу в классе и дома, способствует закреплению знаний по предмету, совершенствует моторно-двигательные навыки учащихся, формирует культуру выполнения ими письменных заданий.

М. М. Богомазова,

преподаватель информатики Тольяттинского военного технического института

КОНЦЕПЦИИ МЕТОДИЧЕСКИХ ПОДХОДОВ К ФОРМИРОВАНИЮ ИНФОРМАЦИОННОЙ ПОДГОТОВКИ КУРСАНТОВ ВОЕННОГО ВУЗА

В зависимости от подходов к пониманию идей, сущности, методов обучения, психологических механизмов формирования знаний, навыков и умений в педагогической науке сложился ряд концепций обучения.

При этом различные концепции обучения можно классифицировать по следующим основаниям:

1. По подходам к определению целей и содержания обучения выделяются следующие концепции:

- ассоциативная теория обучения;
- концепция дидактического энциклопедизма;
- концепция дидактического формализма;
- концепция дидактического утилитаризма;
- концепция функционального материализма;
- парадигмальная концепция;
- кибернетическая концепция обучения;
- концепция лично-ориентированного обучения и персонализации подготовки.

2. Выделяют концепции, основанные на различных методологических подходах к обучению:

- материалистическую концепцию;
- бихевиористскую концепцию;
- прагматическую концепцию.

3. Концепции обучения подразделяются в зависимости от понимания сущности процесса обучения:

- традиционные концепции;
- педоцентрическая концепция;
- концепция сотрудничества.

4. Дидактические концепции, основанные на подходах к структурированию учебного материала:

- программированное обучение;
- концепция модульного обучения.

5. Концепции, обусловленные различными подходами к методике обучения:

- концепция поэтапного формирования умственных действий;
- концепция проблемного обучения.

Из всего вышеперечисленного можно отметить, что в процессе подготовки военнослужащих наибольшее применение нашли следующие концепции:

- ассоциативно-рефлекторная;
- поэтапного формирования умственных действий;
- проблемного и проблемно-деятельностного обучения;
- программированного обучения [2].

Ассоциативно-рефлекторная концепция обучения — основная и наиболее часто реализуемая в подготовке военнослужащих. В ее основе лежит, последовательность обучения военнослужащих, основанная на логике процесса познания. Данная концепция может быть представлена в следующем виде:

- восприятие учебной информации (знания—знакомства);
- осмысление воспринятой информации (знания—репродукции);
- запоминание и сохранение учебного материала (знания—умения);
- применение знаний на практике (знания—трансформации);
- творческая профессиональная деятельность (мастерство) [4].

Эффективное усвоение знаний, формирование навыков и умений, развитие интеллектуальных качеств зависят не только от познавательной активности обучаемых воинов, но и от накопления ими конкретных приемов и способов профессиональной деятельности. В этом плане наибольший эффект дает обучение военнослужащих на основе концепции поэтапного формирования умственных действий. Прин-

ципиальный смысл методов, основанных на данной концепции, заключается в том, что процесс усвоения знаний и формирования соответствующих навыков и умений протекает в ходе овладения определенной деятельностью (выполняя действия — получаю нужные знания о них). Поэтому основным в обучении будет не предварительное заучивание знаний для последующего их применения, а овладение определенными действиями (умственными, сенсорными, двигательными), с помощью которых закрепляются в памяти и знания, необходимые для выполнения самих этих действий. Вводные лекции, рассказы и т. д. будут играть мотивирующую и установочную роль и в общем объеме занимать мало места. Основу в таком обучении образует практическая реализация [3].

Концепция проблемно-деятельностного обучения реализует два основополагающих принципа развития психики человека: принцип проблемности и принцип деятельности в обучении. Сущность проблемно-деятельностной концепции обучения заключается в том, что в процессе учебных занятий создаются специальные условия, в которых военнослужащий, опираясь на приобретенные знания, самостоятельно обнаруживает и осмысливает профессиональную (учебную) проблему, мысленно и практически действует в целях поиска и обоснования наиболее оптимальных вариантов ее решения [1].

Подготовка военнослужащих на основе проблемно-деятельностной концепции обучения предполагает усиление проблемности при проведении учебных занятий, специфическую организацию их мыслительной и практической познавательной деятельности. Проблемно-деятельностное обучение воинов позволяет не только приобретать новые знания, вырабатывать новые навыки и умения, но и развивать интеллектуальные способности, накапливать опыт творческого решения разнообразных задач.

Достаточно успешно реализуется в условиях Вооруженных сил концепция программированного обучения, под которым понимается управляемое усвоение программированного учебного материала с помощью обучающих средств (ЭВМ, программированного учебника, кинотренажера и др.).

Программированный учебный материал представляет собой совокупность сравнительно небольших порций учебной информации, подаваемых обучаемому в определенной логической последовательности. После каждой порции информации дается контрольное задание в виде вопроса, задачи, упражнения, которое нужно выполнить. В случае правильного выполнения контрольного задания обучающий получает новую порцию учебной информации. Функцию контроля выполняет обучающее устройство. В соответствии с данной концепцией по-

строены автоматизированные учебные курсы по освоению компьютерной техники. В настоящее время данная концепция всё активнее применяется при изучении различных учебных предметов в военных вузах, поскольку в образовании внедряются новые информационные технологии [1].

Из всего вышесказанного и исходя из ряда исследований, проводимых психологами и педагогами, в том числе и военными, можно сделать вывод, что при изучении теоретических вопросов более эффективны методики, основанные на ассоциативно-рефлекторной концепции обучения, а при формировании умений и навыков целесообразно использовать методики, базирующиеся на концепции поэтапного формирования умственных действий.

Если рассматривать данные концепции в ходе формирования информационной подготовки, то лучше всего подходят концепции проблемно-деятельностного и программированного обучения. Они хорошо лучше подходят для обучения информатике. Информатика по своей сути является в большей степени практической наукой, поэтому для усовершенствования навыков работы, определенных знаний и умений требуется работать с большим количеством практических задач. Теория тоже играет свою роль, но основные теоретические знания обучающие уже получили в школе. Если рассматривать курсантов как обучающихся, то здесь присутствуют еще и другие факторы по которым можно сказать, что это концепции будут актуальны. Курсант, в отличие от студента гражданского вуза, должен освоить две специальности в тех же временных рамках. Если к тому же учесть специфику обучения в военном институте, когда распорядок рабочего дня у курсанта строго регламентирован и ограничено время самоподготовки, то встает вопрос о модернизации и интенсификации учебного процесса, обеспечивающей прочное усвоение знаний по базовым дисциплинам. Ничто так не укрепляет знания, как практические задачи. Немаловажную роль в процессе приобретения знаний играет мотивация изучения, так как наибольший эффект в обучении достигается в случае твердого убеждения учащегося в необходимости получаемых знаний для последующей работы. Получить эту мотивацию мы можем путем работы на занятиях с практическими заданиями. Многие курсанты, в том числе выпускники, не испытывают потребности в расширении и углублении знаний информатики.

Таким образом, педагогической наукой разработано достаточно много разнообразных педагогических концепций, позволяющих на научной основе, эффективно и качественно обучать военнослужащих. Выбор соответствующей концепции обучения зависит от характера подготовки специа-

листов для войск, содержания проводимых учебных занятий, профессиональной компетенции и методического мастерства офицера.

Литература

1. Военная педагогика / Под ред. О. Ю. Ефремова, СПб.: Питер, 2008.

2. *Коротков Э. Н.* Современные концепции обучения и их применение в подготовке военных кадров. М.: ВПА, 1976.

3. Основы военной психологии и педагогики / Под ред. А. В. Барабанщикова. М.: Просвещение, 1988.

4. *Фокин Ю. Г.* Теория и технология обучения. М.: Академия, 2006.

Требования к статьям, представляемым в рубрику «Теория и практика информатизации образования»

В рубрике «Теория и практика информатизации образования» публикуются статьи по результатам диссертационных исследований в области информатизации образования и методики преподавания информатики.

Все присланные статьи рецензируются. **Публикация статей возможна только при наличии положительного отзыва.** Поскольку рецензирование и предпечатная подготовка материалов занимают два-три месяца, статьи следует присылать в редакцию заблаговременно.

Обращаем особое внимание на то, что целевая аудитория журнала «Информатика и образование» — школьные учителя информатики, завучи по информатизации, преподаватели педвузов, осуществляющие подготовку студентов по специальности «Информатика», поэтому при публикации приоритет отдается материалам, которые могут помочь в практической работе **учителям информатики**, а также статьям, освещающим вопросы подготовки студентов — будущих **учителей информатики**. К сожалению, ни объем журнала, ни его ориентация не позволяют освещать **все** вопросы, связанные с информатизацией образования.

Присланные статьи должны удовлетворять следующим требованиям:

1. Тексты выполнены в редакторе MS Word.
2. Формат листа — А4.
3. Все поля — по 2 см.
4. Шрифт — Times New Roman, кегль — 12 пт, расстояние между строками — полтора интервала.
5. Графические материалы вставлены в текст.
6. Список литературы упорядочен в алфавитном порядке.
7. Объем материала в указанном формате (включая рисунки и список литературы) — 5 полос.

Обязательно должны быть указаны сведения об авторе (авторах):

1. Фамилия, имя, отчество полностью.
2. Научное звание.
3. Должность и место работы.
4. Адрес автора (почтовый адрес с индексом).
5. Телефоны: домашний и рабочий — обязательно с кодом города, мобильный — при наличии.
6. Электронный адрес.

Уважаемые коллеги!

**Обращаем ваше внимание, что соблюдение указанных выше требований
значительно влияет на результативность работы редакции
с вашими материалами.**



КОНКУРС ИНФО 2010

Научно-методический журнал
«Информатика и образование»
объявляет о проведении в 2010 году
конкурса по двум номинациям:

«Рисуем на компьютере в начальной школе»

«Изучение сетевых технологий в VII–XI классах»

Руководит конкурсом **Организационный комитет** (далее – Оргкомитет), состоящий из представителей Российской академии образования, ведущих методистов, членов редакционной коллегии и редакции журнала.

Условия конкурса

1. Участником конкурса может стать любой человек, работающий в системе образования.
2. Возраст участников не ограничен.
3. Участником конкурса может быть индивидуальный заявитель или группа авторов.
4. Участниками конкурса могут быть как граждане России, так и граждане других стран, приславшие свои учебно-методические разработки на русском языке.
5. Форма участия в конкурсе – заочная.

Номинации конкурса

1. Рисуем на компьютере в начальной школе.
2. Изучение сетевых технологий в VII–XI классах.

Сроки проведения конкурса

Конкурсные работы принимаются с 1 января 2010 года по 1 мая 2010 года.

Итоги конкурса будут опубликованы на сайте журнала «Информатика и образование» www.infojournal.ru.

Лучшие работы будут опубликованы в журнале «Информатика и образование».

С критериями оценки конкурсных работ, формой заявки и требованиями к оформлению конкурсной работы вы можете ознакомиться на сайте журнала по адресу: www.infojournal.ru

Работы высылаются в адрес Оргкомитета до 1 мая 2010 года.

Представленные на конкурс работы не возвращаются, рецензии не высылаются.

Контактная информация

Телефон: (495) 210-56-89

Факс: (495) 497-67-96

E-mail: readinfo@infojournal.ru



XIV РОССИЙСКИЙ ОБРАЗОВАТЕЛЬНЫЙ ФОРУМ

Всё об образовании!

14

Всё для образования!

20 – 23 апреля 2010 г.

г. Москва, КВЦ «Сокольники»

2010 – ГОД УЧИТЕЛЯ

ТЕМАТИЧЕСКИЕ РАЗДЕЛЫ ФОРУМА

СОВРЕМЕННОЕ ОБРАЗОВАНИЕ – учебные заведения всех уровней общего, профессионального, дополнительного образования и переподготовки кадров

РЕГИОНЫ РОССИИ – региональные образовательные системы. Роль регионов в реализации национальной образовательной стратегии – инициативы «Наша новая школа»

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ – современные информационно-программные образовательные технологии, образовательные интернет-проекты, дистанционное образование

УЧЕБНАЯ ЛИТЕРАТУРА – учебная, методическая, развивающая и познавательная литература, обеспечивающая государственные образовательные стандарты нового поколения

ВСЁ ДЛЯ ОБРАЗОВАНИЯ – комплексное проектирование, строительство и оснащение образовательных учреждений, новые модели мобильных образовательных комплексов

БЕЗОПАСНОСТЬ И ЗДОРОВЬЕ УЧАЩИХСЯ – здоровьесберегающие программы, детско-юношеские спортивные школы, оздоровительные центры, туристические компании

ВПЕРВЫЕ НА РОФ!

РЕГИОН-ГОСТЬ – Новая категория участия

4-й ВСЕРОССИЙСКИЙ ПРОФЕССИОНАЛЬНЫЙ КОНКУРС «ИННОВАТИКА В ОБРАЗОВАНИИ»



Гран-при конкурса «Жемчужина российского образования»

Научно-методическое сопровождение конкурса: Академия повышения квалификации и профессиональной переподготовки работников образования (АПКиППРО) – федеральный оператор Приоритетного национального проекта «Образование»

Профессиональная поддержка: Московский институт открытого образования (МИОО)

ВСЕРОССИЙСКИЙ КОНКУРС ПРОИЗВОДИТЕЛЕЙ УЧЕБНОЙ ТЕХНИКИ И ОБОРУДОВАНИЯ

Организаторы: Межгосударственная ассоциация разработчиков и производителей учебной техники (МАРПУТ), ЗАО «МВК»

ВСЕРОССИЙСКИЙ КОНКУРС «ЛУЧШИЙ ДИЗАЙН СТЕНДА»

Организатор: ЗАО «МВК»

ВСЕРОССИЙСКИЙ КОНКУРС «САМЫЙ ГОСТЕПРИИМНЫЙ УЧАСТНИК ФОРУМА»

Организатор: ЗАО «МВК»



www.schoolexpo.ru

Организатор:

ЗАО «Международная
Выставочная Компания»

При содействии КВЦ «Сокольники»

Дирекция выставки:

Тел./факс: (495) 925-34-83

E-mail: npi@mvk.ru, ebv@mvk.ru

www.schoolexpo.ru

Традиционно форум проходит при поддержке:

Министерства образования и науки РФ,
Федерального агентства по образованию РФ,
Комитета по образованию
Государственной Думы РФ,
Комитета по образованию Совета Федерации
Федерального Собрания РФ,
Торгово-промышленной палаты РФ

Генеральные партнеры:

Межгосударственная ассоциация разработчиков и производителей учебной техники (МАРПУТ),
Профсоюз работников народного образования и науки РФ

на правах рекламы

Генеральные информационные спонсоры:



РЕГИОНАЛЬНЫЕ ПРЕДСТАВИТЕЛЬСТВА ЗАО «МВК»: МВК УРАЛ: (343) 371-24-76, МВК ВОЛГА: (843) 291-75-89

Научно-методический журнал «ИНФОРМАТИКА И ОБРАЗОВАНИЕ»

Учредители — Российская академия образования,
издательство «Образование и Информатика»



12 выпусков в год

ОСНОВНЫЕ РУБРИКИ ЖУРНАЛА:

- ◆ Общие вопросы
- ◆ Информатизация школы
- ◆ Методика
- ◆ Задачи
- ◆ ИКТ в образовании
- ◆ Педагогический опыт
- ◆ ИКТ в предметной области
- ◆ Зарубежный опыт
- ◆ Информатика в начальной школе

ПОДПИСНЫЕ ИНДЕКСЫ

в каталоге «Роспечать»:

70423 — для индивидуальных подписчиков;

73176 — для предприятий и организаций;

в объединенном каталоге «Пресса России» — 26097

Методический журнал «ИНФОРМАТИКА В ШКОЛЕ»



8 выпусков в год

ПОДПИСНЫЕ ИНДЕКСЫ

в каталоге «Роспечать»:

81407 — для индивидуальных подписчиков;

81408 — для предприятий и организаций;

в объединенном каталоге «Пресса России» — 45751

КОНТАКТНАЯ ИНФОРМАЦИЯ:

ТЕЛЕФОН : (495) 210-56-89 ФАКС (495) 497-67-96

САЙТ: WWW.INFOJOURNAL.RU