

Три “кита” школьной информатики

Как обычно, начиная перед новым учебным годом новую жизнь, привожу в порядок свою библиотечку. Специфика нашего предмета такова, что немало хороших книг быстро устаревают. Когда у меня дома заканчиваются самые дальние полки, я отношу такие книги в школу, но наступает момент, когда и там места не остается. Приходится с частью книг расставаться. Этот момент всегда самый мучительный (как правило, проще бывает найти или сделать лишнюю полку).

Но есть книги, на которых стоит невидимый штамп — “хранить вечно”. Признаюсь честно, я не помню, когда в последний раз открывал “кирпичи” Кнута, но выбросить их, конечно, рука не повернется никогда. Такие книги собраны у меня на отдельных полках, и просто перебирать их — особое удовольствие. Но не всегда.

С мая 2004 г. мне больно брать в руки замечательную, легендарную книжку “Программирование для математиков”. На этом учебнике, написанном А.Г. Кушниренко совместно с его бывшим студентом Г.В. Лебедевым, выросло не одно поколение студентов мехмата МГУ. Много ключевых идей этой книги нашли свое отражение в первых школьных учебниках информатики, изданных миллионными тиражами (говорят, общий тираж основного учебника Кушниренко составил более 7,5 млн. экз.).

Геннадий Викторович Лебедев трагически погиб 12 мая 2004 года. Последний раз я видел его задолго до этого, в 2000 г., на встрече в редакции “Информатики”. А познакомились мы в 1999 г., на совещании в зале коллегий старого здания министерства на Чистых прудах. Это был период жесткой (и, оглядываясь назад, вполне продуктивной) борьбы “пользователей” и “программистов”. Выступление Лебедева было коротким (он вообще изъяснялся кратко, математически точно, но очень ярко — приводя наглядные примеры и понятные сравнения). Для тех, кто был знаком с материалами его архангельских лекций, Лебедев не сказал ничего нового, на остальных же его слова произвели сильное впечатление. (Эти “12 лекций” публиковала наша газета, потом книжка вышла небольшим тиражом в “Лаборатории базовых знаний”; по-моему, в издательстве ее можно купить и сегодня.)

“Школьная информатика стоит на трех “китах”. Первый “ки” — курс должен развивать алгоритмическое мышление школьника как особую форму мышления, специфическую для информатики как науки и очень важную в любой сфере деятельности человека. Второй “ки” — курс должен быть “настоящим”, он может упрощать и сокращать фундаментальные научные положения, приближая их к пониманию школьников, но нельзя включать в курс то, чего нет в науке, нельзя учить чему-то только потому, что учить этому легко и приятно. Третий “ки” — курс должен быть “полным”, должен давать адекватное представление о современной информационной реальности. Сокращая и упрощая, нельзя отказываться от некоторых опорных элементов, даже если их изучение может оказаться сложным”.

Эти положения, сформулированные почти 15 лет назад, для меня и сегодня являются ключевыми методическими принципами.

У меня есть долг. Долг более личный, чем профессиональный. Мы пока так и не смогли сделать специальный номер памяти Лебедева. Планов было много, но как-то все не складывалось. Слишком бережная эта работа. Но мы обязательно сделаем такой номер.

С.Л. Островский,
главный редактор

СОДЕРЖАНИЕ НОМЕРА

ИНФОРМАЦИЯ

Набор слушателей на курсы повышения квалификации 2
Читайте в ближайших номерах “Информатики” .. 20
Фестиваль исследовательских и творческих работ учащихся “Портфолио” 31

КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

В.Ф. Бурмакина, И.Н. Фалина. Как готовиться к тестированию по проверке ИКТ-компетенции школьников. Лекция 2 ... 3–9

ТЕЛЕКОММУНИКАЦИИ

А.А. Дуванов. Азы информатики. Выходим в Интернет 10–20

СЕМИНАР

Е.А. Еремин. Как компьютер читает файл 21–26

МНЕНИЯ

В.А. Матюхин. Преподавание программирования с использованием системы автоматической проверки решений 27–29

АЗЫ ПСИХОЛОГИИ — УЧИТЕЛЮ ИНФОРМАТИКИ

В.Арсланьян. Что и как мы оцениваем? 30–31

“НАЧАЛКА” № 12

Газета-клуб для всех, кто учит информатике маленьких детей

Н.Д. Шумилина. Алгоритмическая гимнастика: человечки “бумажные” и заводные 32–36

“В МИР ИНФОРМАТИКИ” № 77

Газета для пытливых учеников и их талантливых учителей

Семинар
Вычислительная машина с автоматическим управлением 37–39

“Ломаем” голову
“Тарабарская грамота” ... 39

Задачник
Ответы, решения, разъяснения к заданиям из № 7–9/2006 40–44
Может ли быть такое? 44

Microsoft Excel углубленно
Показываем текущее время в Microsoft Excel 44–45

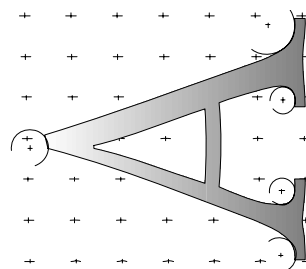
Внимание! Конкурс
Итоги конкурсов № 47–48 для учащихся 45–47
Конкурс № 50 для учащихся 47

КОНКУРС

“Как это делаю я”. Методический конкурс для учителей информатики .. 48

№ 18 (523)

16–30 сентября 2006



Методическая газета для учителей информатики

ИНФОРМАТИК



ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ «ПЕРВОЕ СЕНТЯБРЯ»
 ГАЗЕТА «ИНФОРМАТИКА»
 ОТДЕЛЕНИЕ ПЕДАГОГИЧЕСКОГО ОБРАЗОВАНИЯ ФГП МГУ ИМ. М.В. ЛОМОНОСОВА

ОБЪЯВЛЯЮТ НАБОР СЛУШАТЕЛЕЙ НА КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ НА 2006/2007 УЧЕБНЫЙ ГОД, ПЕРВЫЙ ПОТОК

Курсы проводятся в режиме дистанционного обучения (взаимодействие со слушателями производится посредством обычной или, при наличии у слушателя возможности, электронной почты). Продолжительность обучения — 7 месяцев, нормативный срок освоения учебного материала — 72 часа. Лекционный материал (8 лекций) и контрольные (2 работы) будут публиковаться на страницах газеты «Информатика» (для курса 07-007) или отправляться по почте (для остальных курсов). Итоговую работу слушатели будут выполнять в своих учебных заведениях.

После успешного окончания курсов слушатели получают удостоверение установленного образца о прохождении курсов повышения квалификации от Педагогического университета «Первое сентября» и Отделения педагогического образования ФГП МГУ им. М.В. Ломоносова.

Стоимость обучения составляет 990 рублей за один курс.

В 2006/2007 учебном году мы предлагаем четыре курса по вашей специальности:

Код	Курс
07-001	И.Г. Семакин. Информационные системы в базовом и профильном курсах информатики
07-002	Е.В. Андреева. Методика обучения основам программирования на уроках информатики
07-006	А.А. Дуванов. Основы web-дизайна и школьного «сайтостроительства»
07-007	И.Н. Фалина, В.Ф. Бурмакина. Как готовиться к тестированию по проверке ИКТ-компетенции школьников

Мы также предлагаем один общепедагогический курс, предназначенный для всех работников образования:

21-001	С.С. Степанов. Теория и практика педагогического общения
--------	--

Для зачисления на курсы необходимо прислать в Педагогический университет «Первое сентября» заявку. Пожалуйста, используйте только приведенный ниже бланк или его ксерокопию. Регистрация слушателей производится с 1 апреля по 30 сентября 2006 г. После регистрации вам будет выслан комплект документов с правилами обучения и счетом для оплаты. Вы оплатите счет лишь в том случае, если вас устроят предлагаемые условия (факт подачи заявки ни к чему не обязывает).

ЗАЯВКА Прошу выслать мне комплект документов для зачисления на курсы повышения квалификации. 07-18

ФАМИЛИЯ

ИМЯ

ОТЧЕСТВО

ИНДЕКС

АДРЕС

Телефон (с кодом города): (_____) _____

Электронный адрес (если есть): _____

Место работы: _____

Должность: _____ Стаж работы по специальности: _____

ВНИМАНИЕ! К обучению на курсах повышения квалификации допускаются сотрудники образовательных учреждений, работающие по соответствующей специальности.

Я хочу пройти обучение по курсам (укажите коды выбранных вами курсов):

— — — — —

Если вы обучались в 2005/2006 году на наших курсах, укажите, пожалуйста, ваш идентификатор:

**Заявки следует направлять по адресу: ул. Киевская, д. 24, г. Москва, 121165,
 Педагогический университет «Первое сентября». Справки по тел.: (499) 249-47-82**

ГАЗЕТА "ИНФОРМАТИКА" И ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ "ПЕРВОЕ СЕНТЯБРЯ"
(лицензия А225682, № 020503 от 19.07.2006)
ПРЕДСТАВЛЯЮТ НОВЫЙ ДИСТАНЦИОННЫЙ КУРС
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ ДЛЯ УЧИТЕЛЕЙ ИНФОРМАТИКИ

В.Ф. Бурмакина, И.Н. Фалина

Как готовиться к тестированию по проверке ИКТ-компетенции школьников

Учебный план

№ газеты "Информатика"	Учебные материалы
17/2006	Лекция 1. Цели и задачи проекта по оценке ИКТ-компетентности девятиклассников. Цели и метод тестирования. Описание когнитивных компетентностей, оцениваемых тестом. Структура теста.
18/2006	Лекция 2. Описание познавательных деятельности, составляющих компетентность "определение информации". Примеры заданий на выработку этого умения.
19/2006	Лекция 3. Описание познавательных деятельности, составляющих компетентность "доступ к информации". Как выбрать информационный ресурс, соответствующий заданным критериям? Какие стратегии можно и нужно использовать для поиска информации в многочисленных, часто противоречивых, источниках?
20/2006	Лекция 4. Описание познавательных деятельности, составляющих компетентность "управление информацией". Могут ли ваши ученики разработать самостоятельно или выбрать из предложенных такое представление исходной информации, которое будет наиболее понятно конкретной аудитории? Понимают ли ваши ученики, что с конфиденциальной информацией надо обращаться в соответствии с определенными нормами? Контрольная работа № 1.
21/2006	Лекция 5. Описание познавательных деятельности, составляющих компетентность "оценка информации". Школьнику предлагается несколько информационных источников (например, статьи, сайты и т.п.), из которых он должен выбрать один, наиболее полно удовлетворяющий заданной потребности.
22/2006	Лекция 6. Описание познавательных деятельности, составляющих компетентность "интеграция информации". Если школьник умеет анализировать различные источники по одной и той же теме и на основе этой деятельности создавать новую информацию в сжатом и точном изложении, то он обладает компетентностью "интеграция информации".
23/2006	Лекция 7. Описание познавательных деятельности, составляющих компетентность "создание информации". Школьник обладает компетентностью "создание информации", если он умеет сбалансированно осветить проблему на основе имеющейся, в том числе и противоречивой, информации. Контрольная работа № 2.
24/2006	Лекция 8. Описание познавательных деятельности, составляющих компетентность "сообщение информации". Могут ли ваши ученики адаптировать представленную информацию для конкретной аудитории, умеют ли грамотно цитировать источники, воздерживаться от провокационных высказываний при передаче информации конкретной аудитории?
Итоговая работа. Итоговая работа должна быть отправлена в Педуниверситет "Первое сентября" до 28 февраля 2007 г.	

Лекция 2.

Описание познавательных деятельности, составляющих компетентность "определение информации".

Примеры заданий на выработку этого умения

Авторы курса "Как готовиться к тестированию по проверке ИКТ-компетентности школьников" — учителя информатики, собственно, поэтому-то курс читается на страницах газеты "Информатика". Однако формирование ИКТ-компетентности происходит на всех уроках в школе, происходит во внеурочное время при общении со сверстниками, родителями. Более того, именно на уроках литературы, истории, русского языка активно формируются навыки работы с текстом как художественным, так и учебным. Формирование у школьников ИКТ-компетентности на уроках негуманитарного цикла требует от учителей использования специальных методов и приемов:

1) в первую очередь учитель должен быть настроен на формирование этой компетентности (грубо говоря, помнить о ней всегда);

2) потребуются изменение дидактических целей типовых заданий, которые вы обычно даете своим учащимся (целей

будет как минимум две: изучение конкретного учебного материала и формирование ИКТ-компетентности);

3) на уроках следует выделять время для самостоятельной работы с текстом с дальнейшим групповым осуждением;

4) формированию ИКТ-компетентности помогает использование активных методов обучения [1] (групповая или командная работа, деловые и ролевые игры и т.д.).

Терминология, используемая в данном курсе

Ниже приведены толкования терминов (взяты в основном из БСЭ), используемых в нашем курсе.

Деятельность — специфическая для человека форма активности, направленная на целесообразное преобразование окружающего мира. Цель деятельности определяется потребностями личности или общества. Исходя из вида потребности, различают материальную и духов-

ную деятельность. Основными элементами деятельности являются:

- 1) осознание деятельности в виде мотива;
- 2) оценка наличной ситуации;
- 3) формулирование цели;
- 4) выработка решения как цепи последовательных преобразований наличной ситуации в целевую ситуацию;
- 5) выбор средств деятельности;
- 6) реализация решения.

Когнитивная, или познавательная, деятельность — процесс постижения действительности и приобретения знаний. Познавательная деятельность протекает не в форме пассивного созерцания некоторых внешне данных объектов, а в виде ряда организованных в систему идеальных действий, операций, формирующих определенные “идеальные объекты”, которые и служат средствами для познавательного освоения, отражения объективного мира.

Умение — действие, для выполнения которого необходим сознательный самоконтроль; способность выполнять некоторое действие по определенным правилам.

Навык — доведенное до автоматизма умение решать тот или иной вид задачи. Всякий новый способ действия, протекая первоначально как некоторое самостоятельное, развернутое и сознательное действие, затем в результате многократных повторений может осуществляться уже в качестве автоматически выполняемого компонента действия, т.е. навыка в собственном смысле слова. В отличие от привычки навык, как правило, не связан с устойчивой тенденцией к актуализации в определенных условиях.

Таким образом, для формирования ИКТ-компетентности необходимо активизировать познавательную деятельность учащихся с определенными целями. Слова “познавательный” и “когнитивный” в данном контексте являются синонимами. Деятельность состоит из действий, этот термин в определенных условиях будем заменять терминами “умение” или “навык”.

Таксономия учебных целей Б.Блума и таксономия учебных задач Д.Толлингеровой

Определение ИКТ-компетентности, приведенное в лекции 1, было сформулировано в документе “Отчет “ИКТ-компетентность в мировой практике. Показатель ИКТ-компетентности учащихся и работников образования как индикатор результативности Проекта ИСО”, подготовленный экспертно-аналитическим центром НФПК”. Содержание этого определения обсуждалось в Москве в Психологическом институте Российской академии образования в рамках “круглого стола” “ИКТ-компетентность: парадоксы выделения фундаментальных составляющих параметров и их измерение”, состоявшегося 15 марта 2006 года.

Составляющие ИКТ-компетентность когнитивные действия расположены в порядке возрастания познавательной сложности. Основой такой иерархии является, в частности, таксономия учебных целей Биньямина Блума.

Каждое познавательное умение проверяется с помощью определенных специально подобранных задач. Б.Блум выделяет 6 категорий целей обучения: *знание, понимание, применение, анализ, синтез, оценка*.

Знание является первой из категорий целей обучения, так как запоминание является только частью более сложных процессов соотнесения, оценки, реорганизации, на которых строятся более высокие категории учебных целей (2–6).

Понимание отражается, по мнению Б.Блума, в трех типах поведения:

1. Трансляция (переход от конкретного к абстрактному, использование других терминов, символов, перевод на другой язык, пересказ своими словами);

2. Интерпретация (реорганизация идей в сознании человека, выделение значимых идей, их внутренних связей, обобщение, объяснение, краткое изложение);

3. Перенос знаний (оценка и предсказание, основанное на понимании направлений, тенденций, правил).

Если овладение знаниями на уровне понимания обеспечивает использование обучаемым абстрактных понятий, правил, принципов по данной теме, то усвоение учебного материала на уровне **применения** предполагает, что он будет применять их правильно в похожей ситуации. Различия между пониманием и применением как категориями усвоения учебного материала были зафиксированы в ходе исследования Б.Блума, в котором группе студентов были предложены два теста — один на понимание фактов и принципов, другой — на их применение. Значимость различий между тестом знания и применения подтвердилась результатом эксперимента (корреляция 0,31 и 0,54).

По мнению Б.Блума, категория **анализ** направлена на выделение отдельных частей материала, определение их взаимосвязей и принципов организации, определение предположений, выводов, концепций, которых автор текста придерживается, хотя не высказывает явно.

Категория **синтез** обозначает умение комбинировать элементы так, чтобы получить целое, обладающее новизной; собирать материал из разных источников вместе так, чтобы полученная модель или структура была более понятной, чем исходный материал. *Понимание, применение знаний и анализ* материала являются необходимым условием решения задачи *синтеза*, но задачи этих категорий не требуют оригинальности, уникальности решения. Б.Блум выделяет следующие подкатегории синтеза:

1. Обучаемый пытается в своем выступлении, докладе, сочинении передать некоторые идеи и свой опыт другим с целью информировать, описать, выразиться, убедить.

2. Разработка плана предполагаемых действий.

3. Формулирование и проверка гипотезы, рассмотрение возможных вариантов решения, определение наиболее рационального, нахождение решения множества частных задач в общем виде.

Оценка как категория таксономии обозначает умение оценивать значение того или иного материала (утверждения, художественного произведения, исследовательских данных), основывается на четких критериях, на адекватном понимании и анализе явлений, что принципиально

отличает ее от субъективных “мнений”. Оценка, являясь высшей категорией таксономии учебных целей и используя все другие категории, не обязательно становится последним этапом процесса решения задачи, она может быть прелюдией к приобретению новых знаний, новому пониманию или применению, анализу или синтезу.

Любые измерители достижений учащихся реализуются на практике в виде учебных задач, поэтому пристального внимания заслуживает построенная на основе таксономии учебных целей Б.Блума *таксономия учебных задач Д.Толлингеровой*, в которой все задачи про ранжированы по возрастанию когнитивной сложности и операциональной ценности.

В своей работе “Психология проектирования умственного развития детей” [3] Дана Толлингерова предложила таксономию учебных задач, разделенных по операциям, необходимым для их выполнения. Учебные задачи в ней разделены на 5 категорий, содержащих 27 типов учебных задач:

1. *Задачи, требующие мнемического воспроизведения данных:*

- 1.1. Задачи по узнаванию;
- 1.2. Задачи по воспроизведению отдельных фактов, чисел, понятий;
- 1.3. Задачи по воспроизведению дефиниций, норм, правил;
- 1.4. Задачи по воспроизведению больших текстовых блоков, стихов, таблиц и т.п.

2. *Задачи, требующие простых мыслительных операций:*

- 2.1. Задачи по выявлению фактов (измерение, взвешивание, простые исчисления и т.п.);
- 2.2. Задачи по перечислению и описанию фактов;
- 2.3. Задачи по перечислению и описанию процессов и способов действий;
- 2.4. Задачи по разбору и структуре (анализ и синтез);
- 2.5. Задачи по сопоставлению и различению (сравнение и разделение);
- 2.6. Задачи по распределению (категоризация и классификация);
- 2.7. Задачи по выявлению взаимоотношений между фактами (причина, следствие, цель, влияние, функция, полезность, способ и т.п.);
- 2.8. Задачи по абстракции, конкретизации и обобщению;
- 2.9. Решение несложных примеров (с неизвестными величинами и т.п.).

3. *Задачи, требующие сложных мыслительных операций с данными:*

- 3.1. Задачи по переносу (трансляция, трансформация);
- 3.2. Задачи по изложению (интерпретация, разъяснение смысла, значения, обоснование);
- 3.3. Задачи по индукции;
- 3.4. Задачи по дедукции;
- 3.5. Задачи по доказыванию (аргументации) и проверке (верификации);
- 3.6. Задачи по оценке.

4. *Задачи, требующие сообщения данных:*

- 4.1. Задачи по разработке обзоров, конспектов, содержания и т.д.;
- 4.2. Задачи по разработке отчетов, трактатов, докладов;
- 4.3. Самостоятельные письменные работы, чертежи, проекты.

5. *Задачи, требующие творческого мышления:*

- 5.1. Задачи по практическому приложению;
- 5.2. Решение проблемных задач и ситуаций;
- 5.3. Постановка вопросов и формулировка задач и заданий;
- 5.4. Задачи по обнаружению на основании собственных наблюдений (на сенсорной основе);
- 5.5. Задачи по обнаружению на основании собственных наблюдений (на рациональной основе).

Дидактическая ценность системы учебных задач, по мнению Д.Толлингеровой, связана с выполнением поставленной педагогической цели: если целью учителя было проверить знания учащегося, то достаточно, чтобы тест содержал задачи первой категории, если же цель — проверить, как учащийся использует сложные мыслительные операции, то задачи 1–2-й категорий не позволят гарантировать достижения поставленной цели.

Описание когнитивных действий, составляющих компетентность “определение” или “идентификация” информации

Прежде чем мы начнем работать с первой познавательной составляющей ИКТ-компетентности, заметим, что очень часто для успешного выполнения учебной задачи требуется владение несколькими когнитивными действиями, и отнесение учебной задачи к какой-либо категории подразумевает, что данное познавательное действие имеет преобладающее значение при выполнении задания.

В тест по оценке ИКТ-компетентности, который был предложен девятиклассникам в апреле–мае 2006 года, было включено три задания на оценку когнитивного действия “определение информации”. К сожалению (а скорее, недоумению), результаты оценки сформированности этого умения оказались самыми низкими (см. диаграмму из лекции 1). Объяснить этот низкий результат пока сложно, ведь именно формированием этого навыка занимаются с первого класса на уроках гуманитарного цикла, в средней школе — абсолютно на всех уроках. Вспомните, в начальной школе учащимся предлагается дать название художественной картине, репродукцию которой учитель повесил в классе; учащимся предлагается озаглавить фрагмент текста; найти в тексте описания того, как был одет персонаж, и т.д. и т.п. Без этого умения мы не сможем выполнять другие когнитивные действия, входящие в ИКТ-компетентность. Суть действия “определение информации” — сформулировать вопрос: “Что же мне надо найти в данном источнике информации”.

Конкретный пример из теста на это когнитивное умение приведен в конце лекции. Однако мы советуем нарушить последовательное чтение и познакомиться с

примером из теста. Затем вы еще раз вернетесь к его анализу.

Замечание 1. Обратите внимание, что любое тестовое задание дается в виде описания жизненной ситуации (сценарий задания). Это делается специально, для того чтобы симитировать реальную среду, в которой учащемуся приходится решать аналогичные задачи в жизни.

Замечание 2. Знакомясь с заданиями, вы уже обратили внимание на большой объем текста, который учащийся должен прочесть и переработать при выполнении задания. По представленным нам данным Министерством образования и науки РФ средний девятиклассник функционально читает текст со скоростью 200 слов в минуту. Разработчики тестов даже несколько снизили требования к скорости функционального чтения: 180 слов в минуту.

Замечание 3. Разработчикам тестовых заданий было поставлено условие: выполнение задания не должно требовать знаний по конкретной школьной дисциплине, т.е. содержания заданий должны быть построены на общекультурных вопросах, “житейских” ситуациях и т.д.

Обратимся к таблице-структуре ИКТ-компетентности, приведенной в лекции 1.

Определение (идентификация) информации	Умение точно интерпретировать вопрос (формирование конкретного информационного запроса, в том числе мысленного)
	Умение детализировать вопрос (умение составлять уточняющие запросы в случае необходимости)
	Нахождение в тексте информации, заданной в явном или в неявном виде
	Идентификация терминов, понятий
	Обоснование сделанного запроса

Рассмотрим примеры конкретных заданий, которые направлены на формирование и оценку каждой составляющей этого действия.

1. Умение точно интерпретировать вопрос (формирование конкретного информационного запроса, в том числе мысленного). Умение детализировать вопрос

Приведем следующее задание: “Ваш старший брат недавно женился. Они с женой снимают квартиру. У них скоро должен родиться ребенок. Брат жалуется, что в его квартире много тараканов. Вы хотите помочь брату вывести тараканов. Вам посоветовали прочитать статью “После атомной войны выживут только тараканы и крысы”. До начала чтения статьи сформулируйте вопросы, по которым вы будете искать информацию для решения своей проблемы. Прочитайте статью и выполните задание, приведенное после нее”.

ПОСЛЕ АТОМНОЙ ВОЙНЫ ВЫЖИВУТ ТОЛЬКО ТАРАКАНЫ И КРЫСЫ

Интервью с энтомологом Натальей Новиковой, сотрудником “Зоопарка живых насекомых” и Зоологического музея РАН.

— Тараканов известно больше 3600 видов (от 4 мм до 10 см), обитают они в тропиках, — рассказывает Наталья Васильевна. — Но есть и на Севере, в Приморском крае, редкие виды. В наших квартирах — всего два вида: прусак (рыжий) и черный, которые привыкли к человеку. В природе они не живут, только в домах. Черные раньше пришли к людям, на Руси считалось, что тараканы

в доме — к удаче, к богатству: если переезжаешь на новое место, надо брать с собой. А рыжих в XVIII веке солдаты привезли из Германии — их и называли “прусаками”. Но немцы считают, что этих тараканов русские к ним занесли — еще раньше, когда приезжали в Германию работать на стеклянных заводах.

Тропические тараканы у нас не выживут без особого ухода — им нужна тепло и влажность, а то замерзнут или продует их. А “наши” приспособились к перепадам температур.

— Так вот почему трудно с тараканами бороться — они быстро приспосабливаются?

— Конечно! Ученые считают, что после атомной войны выживут только тараканы и крысы. Они очень древние — жили еще на молодой Земле, когда был другой радиационный фон. И до сих пор здравствуют... В эволюции побеждает тот, кто завоевывает пространство. Если бы не человек, насекомые (тараканы в том числе) могли бы завоевать мир — и царили бы на планете. До сих пор очень много неоткрытых, неизвестных науке видов насекомых.

— Правда, что тараканы не могут без воды?

— Без пищи они могут прожить больше месяца, погрызут бумажку, обои, деревяшку, а без воды не смогут. Но влагу всегда найдут — на трубах скапливается. Или сбегают попить у соседей — и вновь к вам вернуться, если ваши запахи им нравятся. Наши тараканы всеядны, а тропические — сладкоежки. Для зеленых летающих тараканов из Южной Америки мы мед разводим, варенье даем.

В наших домах у тараканов свой мир, они гуляют из квартиры в квартиру. У меня подруга — чистюля: в новой квартире сняли обои до штукатурки, промазали гелями, продезинфицировали, но тараканы все равно живут у них!..

— А кто враги тараканов?

— У подруги азиатская овчарка — так она ест тараканов, а иногда и пчел. А моя кошка не ест. Муж шутит: хоть бы какая польза от нее была!.. Пишут, что мадагаскарский таракан съедает нашего прусака. Но я не пробовала: из дому тараканов сюда принести нельзя, там же они все травленные — могут погибнуть музейные экспонаты. Поэтому здесь у нас только липкие ловушки, никакой химии!..

Настоящие враги тараканов — пауки. Вот этих пепельных тараканов с Канарских островов мы разводим на корм паукам.

— Не все тараканов уничтожают — некоторые наблюдают, другие разводят их. Скажем, чтобы тараканы бега устраивать...

— Известно, что тропические тараканы бегают со скоростью до 4 км/ч, наши прусаки — до 1 км/ч. Черные — медленнее, но забираются в любые щели.

Р.С. Я позвонил по объявлению “Профессиональное уничтожение тараканов, насекомых и грызунов” (на столбе сорвал). Отвечает старушка, называет сумму — 750 рублей за всю квартиру. Обещает полное уничтожение — и гарантия 2 года! “А если появятся?” — “Так это, милый, зависит от того, как вы вести себя будете...” — “Чем ваши средства отличаются от магазинных?” — “Да у нас они покрепче будут!” — “А для детей не вредно?” — “Ни человеку, ни животным вреда не будет. Вот у меня у самой собачка — и ничего, живет... Эти средства только на тараканов действуют!”

Видно, спасение утопающих — дело рук самих утопающих. С помощью “народных” средств — фирменный дезодорант, гель, ловушки — тараканов мы вывели. Не до конца, отдельные экземпляры изредка выползают.

“АиФ Петербург”

№ 46 (535) от 12 ноября 2003 г.

Напишите советы брату по борьбе с тараканами.

Конец задания

Комментарии к заданию

Учащийся должен читать текст с определенной целью (это называется функциональным чтением). Цель чтения формируется через вопросы. В данном случае вопросы могут быть следующими: Чем травить тараканов (кстати, ответа на этот вопрос в статье нет)? Что едят тараканы? Что они пьют? Где любят жить? И т.д. Заметим, что мысленные вопросы формулируются и во время чтения текста в соответствии с изначальной целевой установкой.

Далее учащимся предлагается написать советы брату по борьбе с тараканами на основе статьи (основой этой работы будут являться те мысленные вопросы, которые учащиеся сформулировали до чтения или во время чтения информации). Верными ответами будут:

- в квартире не должно быть мест, где сыро и жарко;
- в квартире не должно быть источников для питья тараканам (немытая посуда, вода около кухонной раковины и т.д.);
- в квартире не должно быть неубранных остатков пищи;
- в квартире не должны находиться предметы с сильным запахом, притягивающим тараканов;
- следует расставить липучие ловушки;
- перекрыть места миграции тараканов из одной квартиры в другую.

Суть задания: *точно интерпретировать вопрос (проблему)*. Проблема формулируется так: “Надо помочь брату”. От учащегося требуется сформулировать (переформулировать, интерпретировать в соответствии с реальной ситуацией) уточняющий вопрос-цель: “Я буду читать эту статью с целью найти советы по борьбе с тараканами”. Далее этот общий вопрос конкретизировать (детализировать).

Если учащийся сумел переформулировать вопрос, то он скорее всего сможет из текста *выделить информацию, заданную в явном или неявном виде*.

Таким образом, существенной характеристикой когнитивного действия “определение информации” является умение (навык) формулировать вопрос-цель: “С какой целью я читаю эту информацию. Что мне в ней надо найти”.

2. Нахождение в тексте информации, заданной в явном или в неявном виде. Идентификация терминов, понятий

Приведем пример заданий на формирование данных действий из предметной области “Информатика”.

В теме “Кодирование информации. Системы счисления” довольно много понятий и определений, с которыми школьники должны уверенно работать. При предъявлении материала мы (учителя) вводим определения, рисуем на доске схему классификации систем счисления и т.д. Более того, именно по такому принципу у нас строятся школьные учебники: определения выделены жирным шрифтом (курсивом, взяты в рамочку); все таблицы и схемы нарисованы. Без сомнения, все эти приемы способствуют запоминанию материала, его усвоению; упрощают работу с учебной литературой. Но, с другой стороны, “академическая” советская организация материала в школьном учебнике не способствует формированию навыков работы с информацией. У любой медали есть две стороны, и изменяющееся общество подталкивает разработчиков учебников к иной подаче материала.

Вводное занятие по теме “Системы счисления” можно построить следующим образом: рассказать о системах счисления обычный материал (об унарной системе счисления, римской, славянской, которая была на Руси до Петра I, двоичной и десятичной, т.е. то, что вы обычно

рассказываете; но не давать определений, тем более под диктовку). После этого предложить задание такого типа.

Задание

Ребята, прочитайте текст о системах счисления. Выпишите все термины изучаемой темы, которые необходимы вам для работы с различными системами счисления. О каких общеизвестных системах счисления рассказывается в этом тексте? Если можете, приведите примеры записи чисел в этих системах счисления.

Сейчас в большинстве стран мира, несмотря на то что в них говорят на разных языках, считают одинаково, “по-арабски”. Но так было не всегда. Еще каких-то пятьсот лет назад ничего подобного и в помине не было даже в просвещенной Европе, не говоря уже об Африке или Америке. Но тем не менее числа люди все равно как-то записывали. У каждого народа была своя собственная или позаимствованная у соседа система записи чисел. Одни использовали буквы, другие — значки, третьи — закорючки. У кого-то получалось удобнее, у кого-то не очень.

Ведь не так-то просто даже имея цифры (значки, которыми записываются числа), записать какое-нибудь число. Для этого нужна система счисления (способ записи чисел с помощью цифр). (Сразу хочу предупредить, что системы счисления бывают непозиционными и позиционными, аддитивными и мультипликативными. Но не стоит пугаться таких грозных слов, на деле все очень просто.)

Самая простая система счисления была еще у древних людей. Какое число нужно записать, столько сделают засечек на палке, или в кучку камешков положат. Но это удобно, пока числа небольшие. Вы только представьте себе число 1000, записанное с помощью кучки камешков, а 1 000 000? Неудобно?

Стали люди придумывать, как по-другому записывать большие числа. Для начала решили, что каждые 10 палочек заменять загогулиной, и счет пошел легче! Так появилась аддитивная система счисления.

Но люди никогда не стоят на месте, они постоянно что-нибудь изобретают. Не захотелось людям вырисовывать по десятку палочек да загогулинок, и решили каждое круглое число обозначить по-особому. Но для этого потребовалось большое количество цифр-символов, и, чтобы не изобретать велосипед, решили использовать алфавит (свой родной). Так и появилась на свет алфавитная аддитивная система счисления. Такая система очень долго использовалась по всей Европе и во многих государствах за ее пределами.

Но далеко не все народы делали свои записи с помощью алфавита или слоговых знаков. В Китае иероглифы не позволили появиться такой системе счисления, и тогда ученые изобрели немного другую систему, названную “мультипликативно-аддитивная система счисления”. Эта система имела одно очень важное свойство: в ней одна и та же цифра в зависимости от расположения в записи числа могла иметь разные значения. Именно такой системой счисления мы с вами сейчас и пользуемся.

Решение и комментарии к заданию

1. В тексте используются следующие термины: система счисления, цифры и алфавит системы счисления, палочная (унарная) система счисления, аддитивная система счисления, аддитивно-мультипликативная система счисления.

Кроме этих терминов, используется термин “круглые числа”; если вы не рассказывали об этом учащимся, то и спрашивать не следует.

Все термины в тексте указаны явно, некоторые приведены с определениями.

2. Рассказывается об унарной (палочной) системе счисления; рассказывается о десятичной системе счисления; об алфавитных системах счисления (например, славянской, которая существовала на Руси до Петра I).

3. От ученика требуется продемонстрировать умение находить информацию, заданную в неявном виде. Например, в абзаце о “круглых” числах говорится и о римской системе счисления. Ученик в состоянии понять, о какой системе счисления здесь говорится, даже если он никогда не слышал о “круглых” числах. Но для этого он должен всегда останавливаться при чтении и задавать себе конкретные вопросы.

В последнем абзаце говорится о привычной нам десятичной системе счисления, хотя некоторые школьники могут написать и о двоичной системе.

4. Обязательно разберите решение этого задания и объясните, как выполнять задания по нахождению нужной информации (сначала обязательно сформулировать вопросы типа “Что ищем”, выделить ключевые слова (термины), можно применять метод сопоставления уже известных знаний с только что прочитанной информацией на предмет идентификации и т.д.).

5. Это же задание можно дать и в более сложном варианте. “Дайте определения следующим понятиям: цифра системы счисления, система счисления, алфавит системы счисления, непозиционная система счисления, позиционная система счисления; аддитивная система счисления, аддитивно-мультипликативная система счисления”.

На основе статьи о тараканах можно дать задание с другой целью поиска информации, но опять же на формирование умения “определение информации”. Условие задания может быть следующим: “На основе газетной статьи подготовьте краткое сообщение по биологии о тараканах”. Очевидно, что при чтении учащийся должен задавать себе другие вопросы (сколько видов, как быстро бегают и т.д.).

Пример тестового задания “Разработка плакатов для школьных предметов”

Время на выполнение: 5 минут

На экране находится описание задания (полный сценарий)

Вы с друзьями решили заработать деньги на туристическую поездку во время каникул. Вам предложили работу в фирме, занимающейся изготовлением плакатов для школьных предметов. Вы согласились. Ваш школьный предмет — информатика. Руководитель проекта подготовил файл, на основе которого ваш друг создал таблицу “История развития вычислительной техники”. Вам поручена функция проверяющего, то есть вы должны перепроверить работу друга и исправить при необходимости допущенные ошибки. За каждую ошибку с вас и вашего друга снимут определенный процент денег. Исходный файл и файл с таблицей находятся в папке Мои документы на диске С.

При смене экрана в левой части появляется **Краткий сценарий**, остальная часть экрана представляет собой два окна, в каждом из которых можно открыть файл.

Краткий сценарий

Перед вами два окна, в которых отображается содержимое папки Мои документы. Откройте в одном окне файл с таблицей, а в другом — файл с информацией об истории вычислительной техники. Сравните содержи-

мое файлов и отметьте галочкой в таблице те строки, которые не должны быть включены в эту таблицу. Для этого нажмите мышкой на неправильную строку. Чтобы убрать галочку, нажмите снова. Отметив все неправильные строки таблицы, переходите к следующему заданию.

В папке Мои документы находятся следующие файлы:
Африка.doc
Бэббидж и другие.doc
Ньютон.doc
Пифагор и Колмогоров.doc
Таблица.doc

Содержимое файла Бэббидж и другие.doc

В 1833 г. английский математик Чарльз Бэббидж (1792–1871) детально разработал проект аналитической машины. Бэббиджу помогала математик Ада Лавлейс (1815–1852) — первый программист-леди: она создала для машины несколько программ, которые хранились на специальных перфорированных картах.

В 1673 г. Готфрид Вильгельм Лейбниц сконструировал арифмометр, позволяющий механически выполнять четыре арифметических действия.

Американский инженер Герман Холлерит (1860–1929) в 1887 году создал электрическую счетную машину — табулятор. Изобретение Холлерита намного ускорило обработку статистических данных.

В 1937 г. Алан Тьюринг предложил универсальную схему вычислений. Его результаты были сформулированы в терминах гипотетической “машины” с удивительно простой структурой, которая обладала всеми необходимыми признаками универсальной вычислительной машины.

В 1642 г. французский математик Блез Паскаль (1623–1662) сконструировал счетное устройство, чтобы облегчить труд своего отца — налогового инспектора. Данное устройство состояло из системы зубчатых колес, вращающих наборные диски с цифрами.

В 1946 г. Джон П. Экерт (род. 1919) и Джон В. Могли (1907–1980) разработали один из первых компьютеров для армии США — ENIAC. По сравнению с современными ЭВМ он был очень громоздок — занимал целый зал и при этом выполнял гораздо меньше операций. ENIAC работал в 1000 раз быстрее, чем “Марк-1”, однако для задания программы приходилось в течение нескольких часов или даже дней подсоединять нужным образом провода.

В 1945 г. к работе Могли и Экерта был привлечен знаменитый математик Джон фон Нейман, который подготовил доклад об этой машине. Доклад был разослан многим ученым и получил широкую известность, поскольку в нем фон Нейман ясно и просто сформулировал общие принципы функционирования универсальных вычислительных устройств, т.е. компьютеров.

В 1943 г. американец Говард Эйкен с помощью работ Бэббиджа на основе электромеханических реле смог построить вычислительную машину под названием “Марк-1”. Ранее мысли, подобные идеям Бэббиджа, были высказаны немецким инженером Конрадом Цузе, который в 1941 г. построил машину Z1.

Содержимое файла Таблица .doc

История развития вычислительной техники

Фамилия	Год создания	Название ЭВМ
Блез Паскаль	1642	Счетная машина
Готфрид В. Лейбниц	1673	Арифмометр
Чарльз Бэббидж	1833	Аналитическая машина
Ада Лавлейс	1815–1852	Аналитическая машина
Герман Холлерит	1887	Табулятор
Говард Эйкен	1943	Марк-1
Конрад Цузе	1941	Z1
Джон П. Экерт, Джон В. Могли	1946	ENIAC
Джон фон Нейман	1945	ENIAC

Оценка сложности выполняемых действий

При выполнении теста специализированная тестирующая программа может отслеживать все действия учащегося, которые он выполняет на компьютере (открытие/закрытие файла, удаление/вставка строк и т.д.). Для каждого участника тестирования ведется протокол его действий. При тестировании, проходившем в апреле—мае, данное задание оценивалось следующим образом:

Наблюдаемое действие	Сложность действия	Оценка действия в зависимости от выполненных работ
Определяет названия нужных файлов	Средняя	<i>Низкая:</i> открывает последовательно все файлы. <i>Средняя:</i> открыл лишний один файл (убедился, что не тот, и закрыл). <i>Высокая:</i> открыл сразу два нужных файла
Определяет нужную информацию на основе прочтения файла Бэббидж и другие.doc и лишнюю информацию удаляет	Высокая	<i>Низкая:</i> оставляет всю информацию. <i>Средняя:</i> удаляет только А.Лавлейс. <i>Высокая:</i> удаляет А.Лавлейс и фон Неймана

Замечания к оценке действий учащегося. Как вы уже знаете, за проведением первого этапа тестирования наблюдали члены группы разработчиков программного продукта и разработчики тестовых заданий. В результате наблюдения за участниками тестирования во время выполнения теста и по результатам опроса на выходе после тестирования мы поняли, что дети в силу своего природного любопытства открывают все файлы, а не

только те, которые им нужны для выполнения задания. Так, например, если они открыли файл *Ньютон.doc* и текст им показался интересным (а он действительно интересный!), то они его читают независимо от того, сколько времени осталось на выполнение теста.

При разработке методов оценки выполнения заданий мы предполагали, что учащиеся будут работать оптимальным образом. Видимо, мы не учли возрастную специфику. Результаты оценивания в настоящий момент анализируются.

Вопросы и задания

1. Объясните, что вы понимаете под “функциональным чтением”.
2. Придумайте задание на формирование когнитивного действия “определение информации” с ориентацией содержания на п. 2.7 по таксономии Толлингеровой.
3. Подумайте, какой термин вам более понятен (удобен, близок): “определение информации” или “идентификация информации”? А какой термин лучше употреблять при работе с учащимися? Обоснуйте свой вывод.

Литература

1. *Фалина И.Н., Мохова М.Н.* Использование активных методов обучения на уроках информатики. / Информатика № 9/2006.
2. *Гальперин П.Я.* Развитие исследований по формированию умственных действий. // Психологическая наука в СССР в 2 т., т. 1. М., 1959.
3. *Толлингерова Д., Голоушкова Д., Канторкова Г.* Психология проектирования умственного развития детей. М., Прага: Роспедагентство, 1994.
4. *Bloom B.S.* Taxonomy of Educational Objectives. Handbook I: Cognitive Domain. NY: McKay, 1956.
5. *Зелман М.* Report on ICT Literacy and Standards, 2004.

Сценарий работы учащегося

Описание действия	Что ученик видит	Необходимые действия на экране	Время
Читает общее задание (89 слов)	Текст задания		25 сек.
Определяет, какие файлы ему надо открыть	Видит содержимое папки Мои документы	Выполняет операцию открытия файлов	5 сек.
Открывает два файла: Бэббидж и другие.doc Таблица.doc	Два одновременно открытых файла в не перекрывающихся окнах		10 сек.
Изучает структуру таблицы в файле Таблица.doc (53 слова), проанализировав заголовки, выделил критерии, по которым информация из файла Бэббидж и другие.doc перенесена в таблицу. Мысленно формулирует вопрос: “Как мне определить лишнюю информацию”?	Два открытых файла	Формирует вопрос, уточняющий задание. Детализирует вопрос. Уточняющий вопрос: “Кто не является непосредственным разработчиком вычислительной техники?”	60 сек.
Читает файл Бэббидж и другие.doc (285 слов) и определяет (выделяет мысленно), какую информацию ему надо удалить из таблицы и редактирует файл Таблица.doc	Два открытых файла	Должен удалить из файла Таблица.doc строку с именами Джона фон Неймана и Ады Лавлейс	200 сек.

Азы информатики.

Выходим в Интернет

А.А. ДУВАНОВ,

г. Переславль-Залесский,

kurs@robotland.pereslavl.ru

Продолжение. Начало см. в № 17/2006

Компьютерные сети. Академия-1.

Общая информация по сетям

В этом разделе приводится общая информация по сетям (более подробные, по сравнению с Читальным залом, сведения о компьютерных сетях).

Классификация компьютерных сетей

Два компьютера можно соединить друг с другом через стандартные компьютерные порты, без каких-либо сетевых адаптеров. Для передачи информации по такому соединению требуются только специальные программы, которые обычно входят в состав операционных систем.

Такое соединение двух компьютеров называют *псевдосетью*.

Реальные сети можно классифицировать по разным признакам:

- Территориальная распространенность (локальные, глобальные сети).
- Принадлежность (семейные сети, домовые сети, сети организаций, предприятий, ведомств, региональные сети, государственные сети, международные сети).
- Скорость передачи информации (низкоскоростные — до 10 Мбит/с, среднескоростные — до 100 Мбит/с, высокоскоростные — свыше 100 Мбит/с).
- Канал передачи информации (проводные: коаксиальный кабель, витая пара, оптоволокно, телефонная линия, бытовая электросеть; беспроводные: передача в диапазоне радиоволн или инфракрасном диапазоне).
- Топология (общая шина, звезда, дерево, кольцо, ячеистая сеть).

Рассмотрим подробнее эти классификации.

Территориальная принадлежность

- Локальные вычислительные сети (ЛВС, LAN — *Local Area Network*).
- Глобальные вычислительные сети (ГВС, WAN — *Wide Area Network*).

Локальная сеть объединяет компьютеры (и другие устройства, например, принтеры, факсы, накопители информации) на небольшой территории.

Небольшая территория позволяет прокладывать дорогие кабельные каналы связи, обладающие высокой скоростью передачи информации: кабели “витая пара” (скорости передачи по разным стандартам: Ethernet — 10 Мбит/с, Fast Ethernet — 100 Мбит/с, Gigabit Ethernet — 1000 Мбит/с), оптоволоконные кабели (скорость передачи в стандарте 10G Ethernet достигает значения в 10 Гбит/с).

Эти скорости значительно превышает величину, необходимую для передачи звуковой информации без компрессии (около 1,5 мегабит в секунду) и полноэкранного видео в формате MPEG2 (около 4,5 мегабит в секунду). И даже скорость передачи данных современных винчестеров, которая не превышает 800 мегабит в секунду.

Таким образом, через локальную сеть можно комфортно (без задержек) работать с данными, расположенными на другом компьютере, например, просматривать по сети видеоролики. Такой способ работы называется режимом онлайн (online, на линии), в отличие от режима офлайн (offline, с отключенной линией), при котором данные сначала копируются по сети на компьютер, а затем сеть отключается, и данные используются автономно.

Локальная сеть может быть расположена внутри одной комнаты или покрывать расстояние нескольких десятков километров.

Глобальная сеть охватывает значительные географические территории и связывает между собой компьютеры и сети компьютеров, расположенные в разных городах и странах.

Прокладка дорогих скоростных каналов связи не всегда экономически оправдана, особенно на больших расстояниях. Для связи между компьютерами часто используют телефонные линии (56 килобит в секунду) и спутниковую радиосвязь (до 5 мегабит в секунду).

Обычной является ситуация, когда локальная сеть входит в состав глобальной. В этом смысле граница между локальными и глобальными сетями довольно условна.

К какой сети принадлежит компьютер школьной сети, которая подключена к Интернету? С одной стороны — это компьютер локальной сети, а с другой — глобальной!

Принадлежность

Семейные сети, домовые сети, сети организаций, предприятий, ведомств, региональные сети, государственные сети, международные сети.

Скорость передачи информации

- Низкоскоростные (до 10 Мбит/с).
- Среднескоростные (до 100 Мбит/с).
- Высокоскоростные (свыше 100 Мбит/с).

В качестве быстродействия сети указывают скорость передачи данных по каналам связи и измеряют ее в килобитах в секунду* (1 Кбит/с = 1000 бит/с) и более крупных единицах: мегабитах в секунду (1 Мбит/с = 1000 Кбит/с), гигабитах в секунду (1 Гбит/с = 1000 Мбит/с) и даже в терабитах в секунду (1 Тбит/с = 1000 Гбит/с).

Один символ обычно кодируется 1 байтом. Так как в одном байте восемь бит, то чтобы определить, сколько символов (байт) в секунду способна пропустить сеть, нужно указанную величину быстродействия поделить на 8.

Реальная скорость передачи данных по сети всегда ниже скорости канала связи и зависит как от протокола сети (правил передачи данных), так и от интенсивности работы пользователей в текущий момент.

Так в сетях, работающих по протоколу Ethernet, сообщения передаются небольшими порциями (пакетами) по общей для всех рабочих станций разделяемой среде. Передачу пакета можно начать лишь тогда, когда среда свободна, значит, чем больше желающих начать работу, тем больше времени уходит на ожидание паузы в сети у каждого передатчика.

Пакет, кроме собственно фрагмента данных, содержит служебную информацию: начальную преамбулу (для синхронизации передатчика и приемника), адрес отправителя и адрес получателя, длину пакета, контрольную сумму (для проверки целостности данных).

Кроме того, приемник посылает передатчику квитанцию о благополучном приеме, а испорченные пакеты приходится передавать заново.

В силу этих причин реальная скорость передачи данных (например, компьютерного файла) существенно ниже скорости работы канала связи.

* В сетевых технологиях скорость передачи информации принято измерять в десятичном виде: приставки К, М, Г, Т обозначают умножение одного бита/с соответственно на 10^3 , 10^6 , 10^9 , 10^{12} (а не на 2^{10} , 2^{20} , 2^{30} , 2^{40}).

Как это ни покажется удивительным, но и “обычные” килобайты бывают не только двоичные (1 килобайт = 1024 байт), но и десятичные (1 килобайт = 1000 байт).

Например, производители жестких дисков часто указывают объем своих изделий в десятичных мегабайтах и гигабайтах. Так, если на жестком диске написано 100 Гб, то реально он может вместить только 93,13 двоичных гигабайт, а 6,87 Гб относятся к рекламному трюку!

Канал передачи информации

- Проводные (передача по коаксиальному кабелю, витой паре, оптоволоконному кабелю, телефонным проводам, проводам бытовых электросетей).

Телефонная сеть — популярный канал связи для подсоединения к серверу глобальной сети. Скорость передачи данных зависит от типа модема, качества телефонной линии от телефонной розетки пользователя до узла АТС (Автоматической Телефонной Станции) и от типа самой АТС. Обычно скорость передачи находится в пределах от 14 Кбит/с до 56 Кбит/с.

Коаксиальный кабель (рис. 2.1) устроен так же, как телевизионный кабель: в центре — медная жила, затем изоляция, затем металлическая оплетка, наконец — внешний слой изоляции. Коаксиальный кабель обеспечивает скорость передачи данных в 10 Мбит/сек. (стандарт Ethernet).

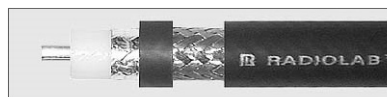


Рис. 2.1. Коаксиальный кабель

Витая пара (рис. 2.2) представляет собой от 2 до 4 пар проводов в изоляции, свитых между собой для уменьшения помех и помещенных в общую изоляционную оболочку. Витая пара обеспечивает скорость передачи данных до 1000 Мбит/сек. (стандарт Gigabit Ethernet).

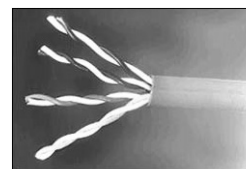


Рис. 2.2. Кабель витая пара

Оптоволоконный кабель (рис. 2.3) похож на коаксиальный кабель, только вместо центрального медного провода здесь используется тонкое (диаметром порядка 1–10 мкм) стекловолокно, а вместо внутренней изоляции — стеклянная или пластиковая оболочка, не позволяющая свету выходить за пределы стекловолокна.



Рис. 2.3. Оптоволоконный кабель

Информация по оптоволоконному кабелю передается не электрическим сигналом, а световым. Главный его элемент — прозрачное стекловолокно, по

которому свет проходит на огромные расстояния (до десятков километров) с незначительным ослаблением. Скорость передачи данных по оптоволоконному кабелю в стандарте 10G Ethernet составляет 10 Гбит/с, но может быть и больше.

Металлическая оплетка кабеля обычно отсутствует, так как экранирование от внешних электромагнитных помех здесь не требуется, однако иногда ее все-таки применяют для механической защиты от окружающей среды.

Обычная бытовая *электрическая сеть* может быть использована для организации компьютерной локальной сети. Для передачи данных применяют специальные устройства (например, устройства стандарта HomePlug, *рис. 2.4*):



Рис. 2.4. Устройство стандарта HomePlug

Основные достоинства такой локальной сети: нет необходимости в проводке специальных сетевых коммуникаций, компьютеры не “привязаны” к сетевым разъемам, их можно разместить в любом месте, где есть розетка электропитания.

Скорость передачи данных для устройств HomePlug составляет от 14 Мбит/с (для HomePlug 1.0) до 200 Мбит/с (для HomePlug AV). Дальность — до 10 км.

- Беспроводные (передача в диапазоне радиоволн или инфракрасном диапазоне).

Спутниковый радиоканал связи обеспечивает передачу данных со скоростью до 5 Мбит/с.

Среди беспроводных сетей в последнее время популярны технологии Wi-Fi и BlueTooth.

Технология *Wi-Fi* (от *Wireless Fidelity*, дословно переводится как *беспроводная точность воспроизведения*). Передача данных по радиоканалу на частоте около 2,4 ГГц. Скорость передачи данных составляет от 10 до 50 Мбит/с. Зона покрытия каждого узла Wi-Fi-сети составляет около 100–150 метров в помещении и до 500 метров (иногда больше) на открытом пространстве.

Чтобы пользователь оказался в сети Wi-Fi, ему достаточно просто попасть в радиус ее действия. Все настройки производятся автоматически. Сегодня существует множество устройств, поддерживающих Wi-Fi. Прежде всего — ноутбуки и карманные компьютеры (КПК).

Технология *BlueTooth* (дословно с английского — “синий зуб”). Обеспечивает низкое по стоимости и энергопотреблению, надежное, защищенное сете-

вое соединение для передачи данных со скоростью до 1 Мбита/с, в радиусе 10 метров (появляются устройства, работающие на расстоянии до 100 метров). Работа ведется на радиочастоте около 2,45 ГГц. Сегодня эта технология популярна для создания локальных сетей в пределах дома, офиса, а также для беспроводной коммуникации различных электронных устройств (например, компьютера с клавиатурой, мышью, принтером, цифровой камерой, мобильным телефоном, MP3-плеером и даже микроволновой печью и холодильником).

Информационная защищенность соединения обеспечивается специальным шифрованием передачи. “Понять” друг друга могут только те устройства, которые настроены на один и тот же шаблон связи, посторонние приборы воспримут переданную информацию как обычный шум.

Топология (способ соединения компьютеров в сеть)

Популярные топологии:

- общая шина (все узлы подсоединены к общему каналу связи);
- звезда (все узлы подсоединены к одному выделенному узлу);
- кольцо (каждый узел соединен с двумя другими; соединения образуют кольцо);
- дерево (иерархическая структура);
- ячеистая сеть (сообщение от одного узла к другому может проходить по нескольким маршрутам).

Локальные компьютерные сети

Локальная сеть — соединение компьютеров, как правило, внутри здания или в пределах небольшой территории в одну группу для совместного использования информации, устройств и услуг.

Локальная сеть соединяет между собой несколько (возможно несколько тысяч) компьютеров при помощи сетевых кабелей, реже другим способом (например, с использованием радиосвязи или бытовой электросети).

Для работы локальной сети каждый ее компьютер должен иметь сетевой адаптер, который, подобно видеоадаптеру, устанавливается в разъем материнской платы компьютера.

Сетевой адаптер играет роль преобразователя информации, поступающей от компьютера, в форму, пригодную для передачи по каналу связи и обратно.

При построении сетей со сложной структурой используют дополнительные коммуникационные устройства: хабы, коммутаторы и маршрутизаторы (см. Академию-2).

Общая шина

На *рис. 2.5* показан пример соединения компьютеров по схеме *общая шина*. При таком соединении

все рабочие станции и сервер подсоединяются к общему кабелю.

Сервер локальной сети



Рис. 2.5. Общая шина

Рабочие станции обмениваются данными друг с другом, сервер оказывает дополнительные услуги: предлагает место на своем жестком диске, принтеры, сканеры, факсы, другие подключенные к нему устройства, организует различные сетевые службы (почта, файловые архивы, тематические страницы, доски объявлений, новостные группы, форумы, чаты, конференции...).

По протоколу Ethernet сигнал от одного сетевого узла (рабочей станции или сервера) передается по общему кабелю, и его “слышат” все другие узлы. Передачу узел начинает лишь тогда, когда в сети “тихо” — мешать чужой передаче запрещено.

Сеть с такой организацией называется Ethernet-сетью с *разделяемой средой* (разделяемой средой здесь является общий кабель).

Чтобы один узел не занял сеть надолго, информация передается небольшими порциями (*пакетами*). После передачи пакета узел делает паузу, и ей может воспользоваться другой сетевой участник для начала своей передачи.

Сервер может и отсутствовать — он не управляет работой сети, но полезен, так как предлагает пользователям дополнительные услуги.

Подробнее о том, как работает сеть с разделяемой средой в стандарте Ethernet, можно прочитать в Академии-2.



Рис. 2.6. Звезда

Сети, имеющие топологию *общая шина*, требуют небольшого количества кабеля, но труднее поддаются диагностике и ремонту по сравнению с сетями, построенными по схеме *звезда*.

Звезда

На рис. 2.6 показан пример построения Ethernet-сети по схеме *звезда*. Все рабочие станции сети и сервер подсоединяются к *портам* (разъемам) специального устройства под названием *хаб* (от англ. *hub* — концентратор).

Поступающий на порт хаба пакет транслируется на все остальные его порты, поэтому сеть с хабом тоже является сетью с разделяемой средой, как и сеть с общей шиной (разделяемая среда состоит из сегментов кабеля, соединенных хабом).

Сети с топологией *звезда* надежны, ведь разрыв кабеля на отдельном узле никак не влияет на работу остальной части сети.

Дерево

Дерево — иерархическое соединение узлов, исходящее из общего узла-корня. Между двумя любыми узлами существует только один маршрут.

Пример Ethernet-сети с иерархической структурой показан на рис. 2.7. Корневой хаб объединяет подсети подразделений одного предприятия.

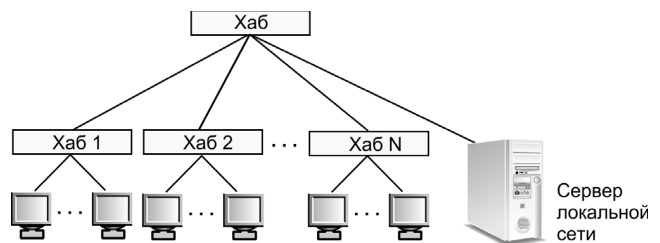


Рис. 2.7. Дерево

Иерархическая сеть, построенная на хабах, по-прежнему остается сетью с одной разделяемой сре-



дой, и принцип ее работы такой же, как у сети с общей шиной: пакет от одного узла транслируется на все остальные узлы этой сети.

Когда среду разделяют много пользователей, дожидаться “тишины” для начала передачи может оказаться сложно. Поэтому для больших сетей вместо хаба используют другое устройство — *коммутатор*.

Коммутатор, как и хаб, соединяет узлы сети своими портами. Но в отличие от хаба устройство наделено “интеллектом” (программным обеспечением): коммутатор передает данные только в тот порт, на котором расположен получатель.

Таким образом, коммутатор делит сеть на отдельные разделяемые среды, повышая скорость работы сети в целом.

На рис. 2.8 показан вариант сети предприятия. В ней корневой хаб заменен коммутатором. Теперь каждое подразделение имеет свою разделяемую среду, независимую от разделяемых сред других подразделений.

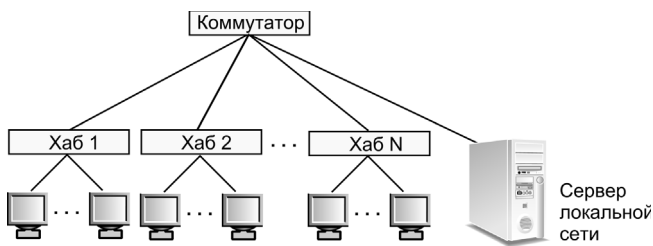


Рис. 2.8. Дерево с коммутатором в корне

Сообщение передается за пределы подразделения лишь тогда, когда это действительно необходимо (серверу или пользователю другого подразделения).

Кольцо

Кольцо — топология, в которой каждый узел сети соединен с двумя другими узлами, образуя кольцо (петлю). Данные передаются от одного узла к другому в одном направлении (по кольцу). Каждый компьютер работает как повторитель, ретранслируя сообщение к следующему компьютеру (рис. 2.9).

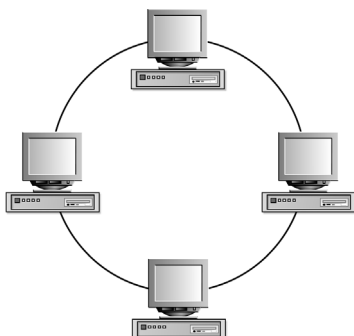


Рис. 2.9. Кольцо

В такой сети одна разделяемая среда, но принцип ее работы отличается от принципа работы разделяемой среды Ethernet.

Изначально по кольцу передается специальное сообщение-*маркер* (другое название: *токен*) — признак свободной среды. Узел может начать передачу лишь тогда, когда получает маркер. Теперь вместо маркера по кольцу следует пакет с данными. Получатель пакета выполняет обратную операцию: заменяет пакет маркером — сеть снова свободна.

По описанному выше алгоритму работают сети, построенные по технологиям Token Ring и FDDI.

В кольцо можно включить и сервер, тогда он будет оказывать пользователям дополнительные услуги.

Ячеистая сеть

Топология, которая более характерна для глобальных сетей. Ее отличительный признак: между парой узлов существует более одного маршрута (рис. 2.10). Для выбора оптимального пути применяются специальные устройства — *маршрутизаторы* (хабы и коммутаторы не работают, когда в сети есть петли).

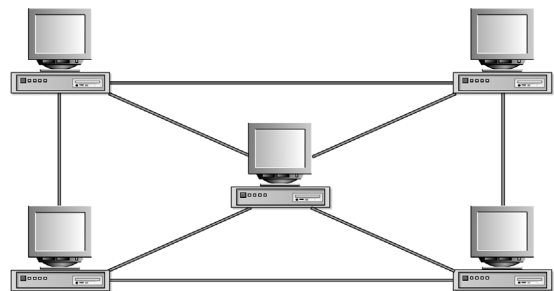


Рис. 2.10. Ячеистая сеть

Ячеистые сети — это сети с *коммутацией пакетов*, то есть такие, в которых пакеты не “разбрасываются” по всем направлениям (как в сетях Ethernet с разделяемой средой), а целенаправленно “проталкиваются” от узла к узлу по направлению к пункту назначения.

За продвижение пакетов в такой сети отвечают маршрутизаторы. Они определяют соседний узел, в который нужно передвинуть пакет для приближения его к пункту назначения.

Маршрутизатор — сетевое устройство (отдельное или на обычном компьютере), которое подобно коммутатору соединяет (коммутирует) узлы сети в том случае, когда это необходимо для передачи пакета. Но в отличие от коммутатора маршрутизатор способен работать в ячеистых сетях и выбирать из разных вариантов наиболее рациональный маршрут для продвижения пакета к пункту назначения.

Глобальные компьютерные сети

Глобальные сети охватывают большие географические пространства.

Примеры глобальных сетей — международные сети Интернет и Фидонет.

Интернет

Интернет — наиболее популярная глобальная компьютерная сеть. В состав ее входят и отдельные компьютеры, но большей частью локальные сети по всему миру.

В Интернете нет единого центра управления. Каждый интернет-сервер берет на себя ответственность за передачу данных от своих клиентов другим интернет-серверам, а также за прием сообщений, предназначенных для своих клиентов.

Кроме того, каждый интернет-сервер обязан принимать и передавать дальше транзитные сообщения, которые передают ему другие интернет-серверы, обеспечивая сквозное прохождение данных по глобальной сети.

Все эти сетевые обязанности сервер выполняет при помощи специального программного обеспечения, которое использует сетевые протоколы передачи данных.

Сетевым протоколом называется согласованный и утвержденный стандарт, содержащий описание форматов данных и правил приема и передачи. Протоколы служат для синхронизации работы сети.

В Интернете нет единого центра управления, и каждый интернет-сервер самостоятельно решает, какому из связанных с ним соседей передать данные, чтобы они дошли до конечного адресата, не заблудившись в сети.

В Интернете нет центра управления, но существует международная неправительственная организация *Internet Society* (ISOC), которая утверждает сетевые стандарты (протоколы) и следит за адресной дисциплиной в сети.

Сообщения в Интернете передаются маленькими порциями (пакетами).

Важным свойством Интернета является многовариантность маршрутов прохождения данных (Интернет имеет ячеистую топологию). Эта вариативность позволяет доставлять сообщения даже тогда, когда отдельные узлы Интернета выходят из строя.

Подробнее об устройстве Интернета будет рассказано в части II “Как работает Интернет”.

Вопросы и ответы Академии-1

Объясните смысл терминов

1. Псевдосеть

Ответ. Соединение компьютеров без каких-либо сетевых адаптеров через стандартные компьютерные порты.

2. Реальная сеть

Ответ. Соединение компьютеров с использованием сетевых адаптеров.

3. Территориальная распространенность сети

Ответ. Различают локальные сети (небольшая территория) и глобальные сети (значительное географическое пространство).

4. Принадлежность сети

Ответ. Признак, характеризующий объединение пользователей сети: семейные сети, домовые сети, сети организаций, предприятий, ведомств, региональные сети, государственные сети, международные сети.

5. Быстродействие сети

Ответ. Быстродействие сети измеряют количеством бит, которое сеть способна передать за одну секунду по своим каналам связи. По этому признаку сети подразделяют на:

- низкоскоростные (до 10 Мбит/с)
- среднескоростные (до 100 Мбит/с)
- высокоскоростные (свыше 100 Мбит/с)

Следует отметить, что реальная скорость работы сети всегда ниже скорости передачи данных по ее каналам связи: сообщения передаются частями (пакетами), которые, кроме передаваемых данных, содержат служебную часть (преамбулу, адрес отправителя, адрес получателя, контрольную сумму). Кроме того, реальная скорость передачи зависит от интенсивности работы пользователей в каждый момент времени: передатчику приходится ожидать паузу (в сетях Ethernet) или специальный сигнал-маркер (в сетях Token Ring и FDDI) для начала передачи.

6. Тип канала передачи

Ответ. Различают проводные и беспроводные каналы связи:

- Проводные каналы: коаксиальный кабель, витая пара, оптоволоконный кабель, телефонные провода, провода бытовой электросети.

- Беспроводные каналы: передача в диапазоне радиоволн или инфракрасном диапазоне.

7. Топология *Общая шина*

Ответ. Все узлы сети подсоединены к общему каналу связи.

8. Топология *Звезда*

Ответ. Все узлы подсоединены к одному выделенному узлу (хабу, коммутатору или маршрутизатору).

9. Топология *Дерево*

Ответ. Иерархическая структура с выделенным узлом в корне.

10. Топология *Кольцо*

Ответ. Каждый узел соединен с двумя другими; соединения образуют кольцо.

11. Топология *Ячеистая сеть*

Ответ. Паутинное соединение. Сообщение от одного узла к другому может проходить по нескольким маршрутам.

12. Сетевой протокол

Ответ. Сетевым протоколом называется согласованный и утвержденный стандарт, содержащий описание форматов данных и правил приема и передачи. Протоколы служат для синхронизации работы сети.

13. Сеть Ethernet

Ответ. Сети Ethernet — набор технологий (и протоколов) построения сети: Ethernet, Fast Ethernet, Gigabit Ethernet, 10G Ethernet.

Первый вариант этих технологий, собственно и называемый Ethernet, описывает правила построения сетей с одной разделяемой средой (при использовании только хабов) или нескольких разделяемых сред (при использовании коммутаторов и маршрутизаторов).

Пакет, переданный одним узлом в разделяемой среде Ethernet, получают все другие узлы, входящие в состав этой сети. Но только тот узел принимает пакет полностью, для которого он предназначен (узлы анализируют адрес получателя, который передается в составе пакета).

14. Сеть Token Ring

Ответ. Сеть Token Ring — это сеть, построенная на кольцевой топологии и работающая по следующему алгоритму.

По кольцу в сети передается специальный сигнал — маркер свободной среды. Когда станция получает маркер, она может вместо него отправить в сеть информационный пакет. Пакет перемещается по кольцу, пока не попадает на станцию назначения. Здесь пакет из сети изымается, а в сеть снова запускается маркер.

Сети Token Ring и Ethernet используют разделяемые среды, но алгоритмы их работы разные.

15. Хаб

Ответ. Хаб — устройство для соединения каналов связи в сети.

Хаб — устройство без программного обеспечения; передает пакет, полученный с одного порта на все остальные.

16. Коммутатор

Ответ. Коммутатор — устройство для соединения каналов связи в сети.

Коммутатор — устройство с программным обеспечением; передает пакет только в порт, на котором находится получатель.

17. Маршрутизатор

Ответ. Маршрутизатор — устройство для соединения каналов связи в сети.

Маршрутизатор — устройство с программным обеспечением; определяет оптимальный маршрут и передает пакет в соседний узел по этому маршруту.

Ответьте на вопросы

1. По каким признакам можно классифицировать компьютерные сети?

Ответ. По приведенным ниже (и другим).

- Территориальная распространенность
- Принадлежность
- Быстродействие
- Тип канала передачи информации
- Топология

2. Что понимается под быстродействием компьютерной сети?

Ответ. Скорость передачи данных по каналам связи.

3. Почему реальная скорость передачи данных по сети всегда ниже быстродействия сети?

Ответ. Сообщения передаются небольшими порциями (пакетами). Пакет, кроме собственно фрагмента данных, содержит служебную информацию: начальную преамбулу (для синхронизации передатчика и приемника), адрес отправителя и адрес получателя, длину пакета, контрольную сумму (для проверки целостности полученного пакета). То есть по сети всегда пересылается больше данных, чем их содержит исходное сообщение.

Чем больше пользователей в сети, тем больше нагрузка на каналы связи: время ожидания для начала передачи пакета возрастает.

В силу этих причин, а также из-за посылки квитанций о приеме и из-за перепосылки испорченных пакетов реальная скорость передачи данных (например, компьютерного файла) существенно ниже скорости работы канала связи (быстродействия сети).

4. Управляет ли сервер работой сети?

Ответ. Нет. Сервер не управляет сетью, он оказывает услуги своим клиентам (обрабатывает их запросы): предлагает место на своем жестком диске, принтеры, сканеры, факсы, другие подключенные к нему устройства, организует различные сетевые службы (почта, файловые архивы, тематические страницы, доски объявлений, новостные группы, форумы, чаты, конференции...).

5. Объясните принцип работы Ethernet-сети с общей шиной.

Ответ. В такой сети все узлы (рабочие станции и сервер) подключены к одному общему кабелю. Сеть работает по следующим правилам:

- Сообщение для передачи разделяется на пакеты.
- Пакет, переданный по кабелю, получают все узлы к нему подключенные.
- Передачу можно начать только тогда, когда в сети нет другой передачи.
- После передачи пакета узел делает паузу, и ей может воспользоваться другой сетевой участник для начала своей передачи.

6. Почему передаваемое по сети сообщение разделяют на пакеты?

Ответ. Сообщение передается небольшими порциями, чтобы дать возможность другим станциям начать свою передачу в паузах между передачей пакетов.

7. В чем преимущества и недостатки сети с общей шиной по сравнению с сетью, имеющей топологию звезда?

Ответ. Сети, имеющие топологию общая шина, требуют небольшого количества кабеля, но труднее поддаются диагностике и ремонту по сравнению с сетями, построенными по схеме звезда.

Обрыв общего кабеля в любом месте приводит к выходу из строя сети в целом.

Сети с топологией звезда надежны, ведь разрыв кабеля на отдельной рабочей станции никак не влияет на работу остальной части сети.

8. В каких сетях используется и как работает хаб?

Ответ. Хаб — сетевое коммуникационное устройство, которое используют для построения Ethernet-сетей по топологиям звезда и дерево.

Хаб имеет несколько портов (разъемов), к которым подключаются сетевые кабели рабочих станций (и сервера).

Поступающий на порт хаба пакет транслируется на все остальные его порты, поэтому сеть с хабами работает так же, как и сеть, построенная на общей шине (это сети с разделяемой средой).

9. Что такое Ethernet-сеть с разделяемой средой?

Ответ. Это сеть, которая работает по следующим правилам:

- Сообщение для передачи разделяется на пакеты.
- Пакет, переданный по сети, получают все узлы к ней подключенные.
- Передачу можно начать только тогда, когда в сети нет другой передачи.
- После передачи пакета узел делает паузу, и ей может воспользоваться другой сетевой участник для начала своей передачи.

Сеть с общей шиной — всегда сеть с разделяемой средой.

Сеть с топологией звезда и дерево является сетью с разделяемой средой (одной), если в ее соединительных узлах используются только хабы (а не коммутаторы и маршрутизаторы).

10. Как работает сеть с технологией Token Ring?

Ответ. Эта сеть имеет кольцевую топологию с одной разделяемой средой. Изначально по кольцу передается специальное сообщение-маркер (другое название: *токен*) — признак свободной среды. Узел может начать передачу лишь тогда, когда получает маркер. Теперь вместо маркера по кольцу следует пакет с данными. Получатель пакета выполняет обратную операцию: заменяет пакет маркером — сеть снова свободна.

11. В разделяемой среде Ethernet пакет, переданный одной станцией, получают все другие станции, подключенные к сети. Верно ли это утверждение для сетей Token Ring?

Ответ. Нет. Передача пакета по кольцу обрывается, когда достигает станции назначения.

Если станция 1 передает пакет для станции 2, то станция 3 получит от станции 2 уже не пакет, а маркер свободной сети (рис. 2.11).

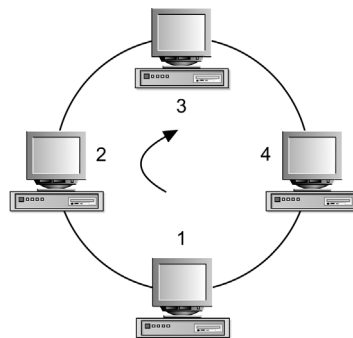


Рис. 2.11. Сеть Token Ring

Если станция 1 передает пакет для станции 3, то этот пакет получают станции 2 и 3, а станция 4 вместо пакета получит маркер.

12. Чем отличается хаб от коммутатора?

Ответ. Коммутатор в отличие от хаба передает пакет только в тот порт, на котором расположен получатель (хаб транслирует пакеты по всем портам).

13. Является ли сеть, в которой используются коммутаторы, сетью с разделяемой средой?

Ответ. Нет. Коммутатор своими портами разбивает сеть на подсети, в каждой из которых своя разделяемая среда (если в них нет других коммутаторов или маршрутизаторов).

14. Объясните принцип работы сети с технологией Token Ring.

Ответ. В такой сети одна разделяемая кольцевая среда, но принцип ее работы отличается от принципа работы разделяемой среды в Ethernet-сетях, построенных по топологиям общая шина, звезда и дерево.

Изначально по кольцу передается специальное сообщение-маркер (другое название: *токен*) — признак свободной среды. Узел может начать передачу лишь тогда, когда получит маркер. Теперь вместо маркера по кольцу следует пакет с данными. Получатель пакета выполняет обратную операцию: заменяет пакет маркером — сеть снова свободна.

15. Объясните принцип работы сети с коммутацией пакетов.

Ответ. Сеть, в которой пакет передается соседнему узлу по направлению к станции назначения, называется сетью с коммутацией пакетов.

Коммутация пакетов характерна для ячеистой сети. В ней узлы могут связывать несколько маршрутов, и требуется специальное устройство, отвечающее за выбор оптимального пути.

Таким устройством является маршрутизатор: он может входить в состав универсального компьютера

или быть отдельным устройством с микропроцессором и памятью, в которую “зашивается” соответствующее программное обеспечение.

Принцип работы сети с коммутацией пакетов: пакет передается маршрутизатором в тот соседний узел, который является первым звеном оптимального пути от маршрутизатора до станции назначения.

Если соседний узел является конечным, путешествие пакета завершается, в противном случае маршрутизатор текущего узла определяет следующий шаг, заново вычисляя оптимальный маршрут от себя до станции назначения.

Коммутация пакетов характерна для ячеистой сети. Но можно представить ячеистую сеть, в которой каждый узел связан со всеми другими узлами отдельными каналами связи. В такой сети сообщения можно передавать независимо друг от друга без коммутации пакетов и без разделения среды. Но такие сети почти не используются: они дороги (много каналов связи), а выход из строя одного канала приводит к необходимости коммутации пакетов для продолжения работы.

Коммутировать пакеты можно не только в ячеистой сети, но и в иерархической (которая является частным случаем ячеистой). В иерархической сети коммутировать пакеты может не только маршрутизатор, но и коммутатор.

16. Может ли хаб (коммутатор, маршрутизатор) работать в сети с ячеистой топологией?

Ответ. Хабсы и коммутаторы работают только в сетях без циклов, то есть в таких, в которых между любыми двумя узлами существует ровно один маршрут. Это сети с топологиями: общая шина, звезда, дерево.

Маршрутизатор ориентирован на работу в сетях с циклами, то есть на работу в таких сетях, в которых между узлами может существовать несколько маршрутов. Маршрутизатор вычисляет оптимальный путь для пакета и проталкивает его в соседний узел по пути следования.

Маршрутизатор может работать в кольцевой сети, в ячеистой сети и, конечно, в иерархических сетях (которые являются частным случаем ячеистых).

17. Какой топологией обладает сеть Интернет?

Ответ. Интернет — это “стихийная” сеть, связанная в которой не строилась по заранее продуманному плану. Конечно, это сеть с ячеистой топологией.

Более подробно. Интернет объединяет отдельные компьютеры, но в большей степени он состоит из других сетей, которые входят в него как подсети.

Подсети Интернета могут иметь самую разнообразную топологию, в частности, немало среди них сетей Ethernet, построенных на разделяемых средах и имеющих в силу этого иерархическую структуру (без циклов).

Подсети, подсоединяясь к Интернету, обязаны на внешнем уровне работать по правилам Интернета, в частности, маршрутизировать свои и транзитные пакеты, учитывая ячеистую топологию связей глобальной сети. Внутри же, в своем “королевстве”, подсети могут работать как угодно и обладать любой топологией.

Поэтому при ответе на вопрос “какой топологией обладает сеть Интернет?” можно сказать так: на магистральном уровне (уровне соединения подсетей Интернета) — это ячеистая сеть с коммутацией пакетов. В подсетях же Интернета топология может быть самой разной, от общей шины до ячеистой топологии (локальные сети Интранет).

Вопросы зачета (с ответами)

В заданиях 11–15 зачетного класса нужно указать топологию сети по приведенному рисунку. Перед решением Зачетного класса рекомендуется прочитать нижеследующее замечание.

Замечание. При решении заданий, в которых требуется определить топологию сети, указывайте минимальную топологию из всех возможных.

Дело в том, что звезда, например, является частным случаем дерева, а дерево — частным случаем ячеистой структуры.

Вложения описанных в учебнике топологий отражает схема, показанная на рис. 2.12.

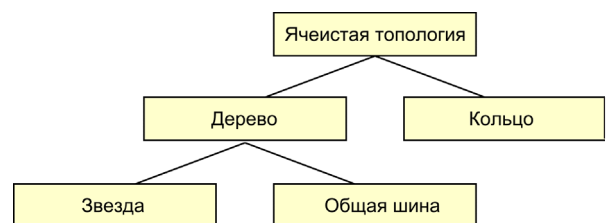


Рис. 2.12. Иерархия топологий

Заметим, кстати, что соединения звезда и общая шина топологически совпадают (рис. 2.13).

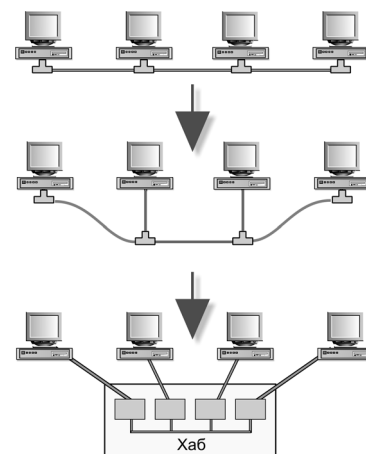


Рис. 2.13. Звезда и общая шина топологически совпадают

Звезда есть дерево с корнем-хабом.

А дерево есть частный случай ячеистой структуры (между любыми двумя узлами в дереве ровно один маршрут).

Кольцо, конечно, деревом не является, но оно тоже частный случай ячеистой структуры.

В силу вложенности описанных выше топологий, правильными ответами на вопрос “какая топология показана на рис. 2.14?” будут: звезда, дерево, ячеистая.

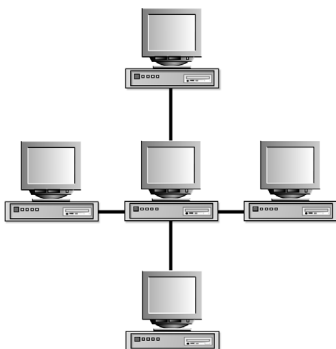


Рис. 2.14. Какая топология?

Но указывать в качестве ответа надо топологию, расположенную на нижнем уровне вложенности, считая при этом топологию звезда и общая шина различными.

Для приведенного выше примера нужно назвать в качестве ответа: “звезда”.

В каждом задании Зачетного класса отметьте *все* правильные высказывания.

1. LAN:

- 1.1. глобальная сеть
- 1.2. мировая сеть
- 1.3. Интернет
- 1.4. компьютерная сеть
- 1.5. локальная сеть

Правильные ответы: 4, 5.

2. WAN:

- 2.1. компьютерная сеть
- 2.2. локальная сеть
- 2.3. глобальная сеть
- 2.4. Интернет

Правильные ответы: 1, 3.

3. Интернет:

- 3.1. сеть с единым управляющим центром
- 3.2. компьютерная сеть
- 3.3. локальная сеть
- 3.4. глобальная сеть
- 3.5. мировая сеть

Правильные ответы: 2, 4, 5.

4. Сервер:

4.1. обеспечивает доступ к своим устройствам (диски, принтеры, факсы...)

4.2. необходимый элемент сети (сеть без него работать не будет)

4.3. управляет работой сети

4.4. обеспечивает сетевой сервис (почты, файловый архив, web-страницы...)

Правильные ответы: 1, 4.

5. Сообщение для передачи в сеть делят на небольшие порции (пакеты) по следующим причинам:

5.1. чтобы один узел не занял надолго канал связи

5.2. передача коротких сообщений обходится дешевле

5.3. чтобы повысить защиту информации от взлома

5.4. каналы связи не способны передавать длинные сообщения

Правильный ответ: 1.

6. Пакет, передаваемый в Ethernet-сеть с разделяемой средой:

6.1. проталкивается от узла к узлу до станции назначения

6.2. разделяется на столько частей, сколько в сети рабочих станций

6.3. перемещается по каналу, который заранее подготавливается (коммутируется)

6.4. получают все узлы этой сети

Правильный ответ: 4.

7. Пакет, передаваемый в сеть с коммутацией пакетов:

7.1. получают все узлы этой сети

7.2. проталкивается от узла к узлу до станции назначения

7.3. разделяется на столько частей, сколько в сети рабочих станций

7.4. перемещается по каналу, который заранее подготавливается (коммутируется)

Правильный ответ: 2.

8. Хаб:

8.1. может работать в ячеистой сети

8.2. проталкивает пакет в соседний узел по пути следования пакета

8.3. устройство для соединения каналов связи в сети

8.4. передает пакет, полученный с одного порта на все остальные

8.5. передает пакет только в порт, на котором находится получатель

8.6. работает под управлением программного обеспечения

8.7. делит сеть своими портами на независимые разделяемые среды

Правильные ответы: 3, 4.

9. Коммутатор:

9.1. делит сеть своими портами на независимые разделяемые среды

9.2. может работать в ячеистой сети

9.3. устройство для соединения каналов связи в сети

9.4. передает пакет, полученный с одного порта на все остальные

9.5. передает пакет только в порт, на котором находится получатель

9.6. работает под управлением программного обеспечения

Правильные ответы: 1, 3, 5, 6.

10. Маршрутизатор:

10.1. передает пакет, полученный с одного порта на все остальные

10.2. передает пакет только в порт, на котором находится получатель

10.3. работает под управлением программного обеспечения

10.4. может работать в ячеистой сети

10.5. проталкивает пакет в соседний узел по пути следования пакета

10.6. устройство для соединения каналов связи в сети

Правильные ответы: 2, 3, 4, 5, 6.

11. Укажите топологию сети:

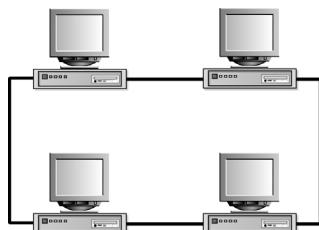


Рис. 2.15

Правильный ответ: кольцо.

12. Укажите топологию сети:

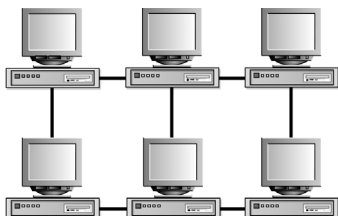


Рис. 2.16

Правильный ответ: ячеистая.

13. Укажите топологию сети:

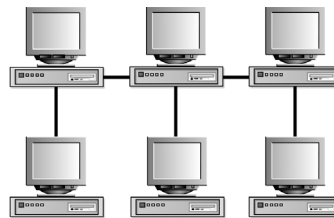


Рис. 2.17

Правильный ответ: дерево.

14. Укажите топологию сети:

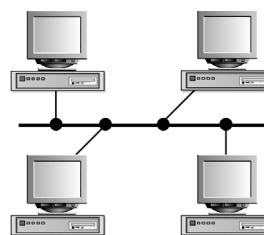


Рис. 2.18

Правильный ответ: общая шина.

15. Укажите топологию сети:

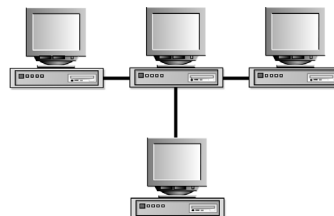


Рис. 2.19

Правильный ответ: звезда.

Читайте в ближайших номерах "Информатики"

- Абсолютный хит сезона (!) — "Примерные ответы на профильные экзаменационные билеты", подготовленные Е.А. Ереминым и А.П. Шестаковым.

- Продолжение курса лекций В.Ф. Бурмакиной и И.Н. Фапиной.

- Новые задания методического конкурса "Как это делаю я".

- Следующие статьи из нового цикла А.А. Дуванова, посвященного компьютерным сетям.

- Задания с подробным разбором (материал полностью готов к творческому воспроизведению) конкурса для младших школьников "ТРИЗформашка", подготовленные пермскими авторами.

☞ Кроме того, в октябре все подписчики "Информатики" получат брошюру из серии Библиотечка "Информатики" "Эффективность алгоритмов" (автор — Я.Н. Зайдельман).

В брошюре на примерах трех задач с простыми условиями рассматриваются очень интересные и нетривиальные вопросы. Как выбор алгоритма решения влияет на время счета даже на самом быстром компьютере? Как сравнить эффективность алгоритмов? Всегда ли существует эффективный алгоритм? Стоит ли вообще задумываться над вопросами эффективности, когда вычислительные мощности современных компьютеров позволяют даже весьма громоздкие задачи решать за приемлемое время "в лоб"?

Некоторые решения сопровождаются подробным анализом и математическими выкладками.

Материал может быть использован даже при изучении алгоритмизации на базовом уровне и совершенно незаменим для профильного курса информатики.

Как компьютер читает файл

Е.А. ЕРЕМИН,
г. Пермь

Окончание. См. № 17/2006

Несколько экспериментов

Согласно старой доброй школьной традиции, после изучения теоретического материала полезно для закрепления знаний попытаться применить их на практике. Предлагается не нарушать традиции и проделать несколько экспериментов, непосредственно иллюстрирующих материалы статьи.

В качестве среды для экспериментов используется распространенная в образовании система программирования Delphi. Автор предполагает наличие некоторых практических навыков по работе с системой (помещение компонентов на форму, изменение их размера, установка значений свойств), а также начальное знакомство с основными операторами Паскаля. Глубоких знаний языка программирования не требуется, несмотря на неочевидность некоторых обсуждаемых в статье приемов.

Эксперимент 1. Чтение файла

Для более детального знакомства с процессом чтения файла посмотрим, как данное действие может программироваться в системе Delphi. Предлагается испробовать три варианта, демонстрирующие различные по степени подробности и сложности способы.

В качестве общей ко всем вариантам подготовительной работы сделаем следующее. Запустим Delphi и стандартным образом создадим на автоматический сгенерированной форме Form1 несколько простых компонентов. Начнем с многострочного поля редактирования Memo1, которое предназначено для отображения прочитанного текстового файла. Для инициализации процесса чтения установим кнопку Button1. Наконец завершим подготовительную часть добавлением на форму диалога открытия файла OpenFileDialog. Заметим, что данный весьма полезный для программирования обмена с диском компонент находится на вкладке **Dialogs** (найдите соответствующую закладку на палитре компонентов). Кроме того, данный компонент не является визуальным, так что помещаемый на форму квадратик с символической пиктограммой после запуска программы отображаться не будет, следовательно, его местоположение на форме не имеет никакого значения.

Далее создадим обработчик кнопки Button1. Для этого, согласно принятым в Delphi “сокращенным” приемам работы, достаточно дважды щелкнуть по ее изображению левой кнопкой мыши. В ответ на наши действия система создаст пустое тело обра-

ботчика главного события для выбранного компонента: в данном случае Button1Click. Набираемый текст обработчика будет разным для каждого из вариантов.

Вариант А. Чтение файла целиком

В данном случае файл читается максимально просто — целиком. Обработчик состоит из одного оператора IF и имеет вид

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  IF OpenFileDialog.Execute
  THEN
    Memo1.Lines.LoadFromFile(OpenDialog1.FileName)
end;
```

(здесь и далее в листингах жирным шрифтом выделен набираемый текст; остальную часть программы система генерирует автоматически).

Программа работает следующим образом. При нажатии кнопки Button1 для компонента OpenFileDialog вызывается метод Execute¹, который рисует на экране стандартный диалог Windows и отслеживает все манипуляции пользователя по выбору файла. Оператор IF требуется потому, что человек, вообще говоря, может отменить диалог, ничего не выбрав, так что загрузка файла не потребуется. Для поддержки такой возможности метод Execute, являясь функцией, возвращает логическое значение true только тогда, когда выбор был сделан. В этом случае и производится чтение файла: для текстовых строк Lines компонента Memo1 вызывается метод LoadFromFile, который обеспечивает считывание с диска необходимого файла. Его имя передается из свойства FileName компонента OpenFileDialog в качестве параметра.

В приведенном примере мы видим, что Delphi обеспечивает чтение файла по заданному имени и, строго говоря, по адресу ОЗУ (последний определяется местом хранения текста внутри компонента Memo1). Доступ к файлу выглядит совершенно “пользовательски”: указали имя и получили полный текст из файла; совсем как в Блокноте или любом другом текстовом редакторе.

Вариант В. Чтение файла по строкам

Теперь рассмотрим классический вариант чтения файла с использованием вызовов стандартных дисковых процедур. С этой целью для того же самого

¹ Глагол “execute” в переводе на русский язык означает “выполнить”.

события `Button1Click` наберем более сложный обработчик.

```
procedure TForm1.Button1Click(Sender: TObject);
VAR f: TextFile;
    s: string;
begin IF OpenFileDialog1.Execute THEN
  BEGIN AssignFile(f, OpenFileDialog1.FileName);
        Reset(f);
        WHILE NOT EOF(f) DO
          BEGIN Readln(f, s);
                Memo1.Lines.Add(s)
          END;
        CloseFile(f);
  END
end;
```

Обработчик устроен аналогично предыдущему варианту, но в данном примере расположенная после `THEN` условно выполняемая часть является более подробной. Она состоит из вызова нескольких стандартных функций и показывает, как раньше программировалось поэлементное (в нашем случае построчное) чтение файла. Разберем текст программы более детально.

Процедура `AssignFile` устанавливает связь между файловой переменной `f` и реальным файлом на диске. В дальнейшем все дисковые операции уже будут выполняться путем обращения к этой переменной без упоминания имени файла.

Затем процедурой `Reset` подготавливается чтение файла. Система при этом проверяет наличие указанного файла на диске и, если его находит, то инициализирует канал обмена и выделяет в ОЗУ все необходимые служебные области; в противном случае на экран выводится сообщение об ошибке и обработчик аварийно завершается.

Далее следует цикл до конца файла (`EOF` — *End Of File*), который с помощью процедуры `Readln` построчно читает текст. Прочитанная строка заносится в переменную `s`, которая благодаря методу `Add` добавляется в `Memo1`.

Процедура `Close` служит для корректного завершения работы с файлом. В случае чтения ее вызов необязателен (мы рискуем лишь напрасно оставить занятой служебную область, выделенную в ОЗУ); в случае записи без данной процедуры файл будет иметь нулевую длину, и система не сможет с ним в дальнейшем работать.

На примере рассмотренного варианта мы видели типичную картину работы с файлом: он открывается (подготавливается к обмену), производится чтение и файл закрывается (происходит корректное завершение обмена). Именно в таком стиле традиционно работают с файлами программисты. Обратим внимание, что и при таком подходе никакие технические особенности не видны: все они скрыты в дисковых процедурах `Reset`, `Read`, `Close` и др.

Вариант С. Блочное чтение файла

Наконец, рассмотрим еще один вариант чтения файла. Он интересен с той точки зрения, что позволяет имитировать (именно имитировать!) чтение содержимого файла по секторам.

В Паскале предусмотрены специальные процедуры для работы с блоками данных, которые рассматривают содержимое файла как последовательность байт, лишенную какого-либо конкретного содержания. Иными словами, их можно одинаковым образом применять не только к текстовым, но и к числовым, графическим и всем прочим файлам. Данный режим чтения хорошо иллюстрирует, как работает с файлами операционная система; в самом деле, при копировании файлов она не делает никакой разницы между текстовыми и музыкальными файлами.

Размер блока может быть задан произвольным количеством байт, начиная от одного, но с позиций нашей статьи наиболее интересно установить его равным 512 байтам — тогда читаемый логический блок будет эквивалентен реальному сектору.

Обработчик для варианта С будет следующим.

```
procedure TForm1.Button1Click(Sender: TObject);
const r = 512; {размер блока 512 байт}
var f: File;
    buf: array[1..r] of Char;
    i, NumRead: Integer;
begin IF OpenFileDialog1.Execute THEN
  BEGIN AssignFile(f, OpenFileDialog1.FileName);
        Reset(f, r);
        REPEAT FOR i := 1 to 512 DO
          buf[i] := CHR(0);
          BlockRead(f, buf, 1, NumRead);
          Memo1.Lines.Add(buf);
          {Memo1.Lines.Add("=====");}
        UNTIL NumRead = 0;
        CloseFile(f);
  END
end;
```

Программа во многом похожа на предыдущую, но имеет целый ряд особенностей. Во-первых, обратите внимание на то, что файловая переменная `f` теперь объявлена как файл, лишенный конкретного типа; его называют еще *нетипизированным* файлом. Во-вторых, в процедуре `Reset` для такого файла задается второй параметр (константа `r = 512`), определяющий размер блока. Наконец, в переменной `buf` для чтения блока организуется буфер необходимого размера.

Далее в цикле `Repeat` производится чтение блоков. Оно начинается с очистки буфера, что оказывается полезным при чтении последнего неполного блока. Собственно чтение осуществляется с помощью специальной процедуры `BlockRead`, при обращении к которой указывается 4 параметра: файловая переменная, буфер для чтения, количество читае-

мых блоков и переменная, в которую система возвратит количество действительно прочитанных блоков. Благодаря последнему параметру удастся распознать окончание цикла: когда файл закончится, в качестве значения результата будет возвращен 0.

Подчеркнем, что блочное чтение является файловой процедурой, т.е. она не может выйти за пределы файла. Поэтому с ее помощью нельзя увидеть те байты последнего сектора, которые хранятся на диске после байтов, реально входящих в файл. Это единственное отличие функционирования данной программы от чтения реальных секторов.

Прочитанные блоки, длина которых совпадает с длиной секторов, попадают в переменную `buf` и затем добавляются в компонент `Memo1`. Строка в фигурных скобках в приведенной программе является комментарием и компилятором игнорируется. Если же убрать скобки, то в программу вставится оператор, разделяющий двойной линией между собой текст разных секторов.

Эксперимент 2. Чтение каталогов и отслеживание путей

Цель данного эксперимента — научиться читать различные каталоги с тем, чтобы проверить теоретические сведения, изложенные на этапе 2, в частности, отследить логику, изображенную на *рис. 6* (см. № 17).

Для решения поставленной задачи создадим на форме следующие компоненты (см. *рис. 9*). В качестве органа ввода, в который будем набирать интересующий нас путь, добавим компонент `Edit1`. Для отображения информации о каталоге поместим на форму таблицу `StringGrid1` (см. вкладку компонентов `Additional`), где количество столбцов (свойство `ColCount`) установим равным трем. Наконец, как и в предыдущих экспериментах, создадим пусковую кнопку `Button1`. Придадим окну нашего приложения эстетически приятный вид, например, такой, как на *рис. 9*.

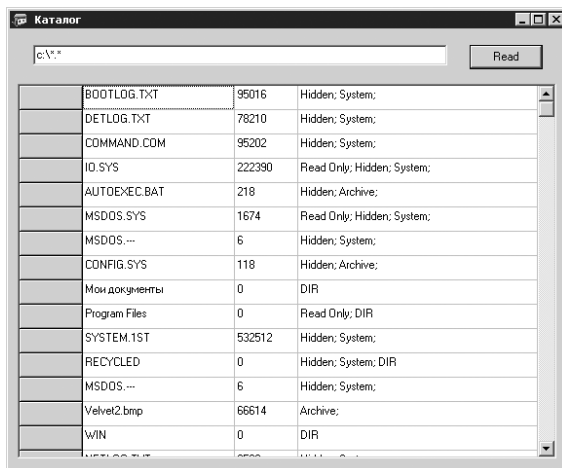


Рис. 9. Вид окна программы для эксперимента 2

Теперь создадим обработчик нажатия кнопки согласно процедуре, описанной ранее в эксперименте 1, и наберем текст обработчика, добавив к нему процедуру под названием `analiz` для отображения информации об одной записи каталога.

```

procedure analiz(sr: TSearchRec);
var s: string;
begin
with Form1.StringGrid1 do
  begin Cells[1,RowCount-1] := sr.Name;
    Cells[2,RowCount-1] := IntToStr(sr.Size);
    s := '';
    if (sr.Attr and faReadOnly) <> 0 then
  s := s + 'Read Only; ';
    if (sr.Attr and faHidden) <> 0 then
  s := s + 'Hidden; ';
    if (sr.Attr and faSysFile) <> 0 then
  s := s + 'System; ';
    if (sr.Attr and faVolumeID) <> 0 then
  s := s + 'Label; ';
    if (sr.Attr and faDirectory) <> 0 then
  s := s + 'DIR ';
    if (sr.Attr and faArchive) <> 0 then
  s := s + 'Archive; ';
    Cells[3,RowCount-1] := s;
  end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var sr: TSearchRec;
  FileAttrs: Integer;
begin
with StringGrid1 do
  begin RowCount := 1;
    Cells[1,0] := '';
    Cells[2,0] := '';
    Cells[3,0] := '';
  end;
  FileAttrs := faAnyFile;
  if FindFirst(Edit1.Text, FileAttrs, sr) = 0 then
  begin
    with StringGrid1 do
    begin analiz(sr);
      while FindNext(sr) = 0 do
      begin RowCount := RowCount + 1;
        analiz(sr);
      end;
      FindClose(sr);
    end;
  end;
end;
end;

```

Проследим за работой программы.

Сначала обработчик очищает таблицу, предварительно сделав ее однострочной (заранее неизвестно, сколько именно строк потребуется). Переменной `FileAttr`, отвечающей за атрибуты интересующих нас файлов, присваивается значение `faAnyFile`, указывая тем самым, что нас интересует любой файл. Затем начинается анализ каталога.

Для поиска в каталоге по шаблону в ОС, начиная с СР/М, предусмотрены две функции: `FindFirst` — поиск первого вхождения и `FindNext` — поиск следующего вхождения. Сначала применяется первая, а затем

циклически вызывается вторая. Признаком того, что поиск не увенчался успехом, является возврат функциями ненулевого значения в качестве результата.

При обращении к функции `FindFirst` задаются следующие параметры: шаблон для поиска (у нас он берется из поля ввода `Edit1`), атрибуты искомого файла (у нас любой) и переменная особого типа `TSearchRecord`, в которую будет возвращена структурированная определенным образом информация об очередном найденном объекте, удовлетворяющем шаблону. При повторном обращении, производимом с помощью функции `FindNext`, задание первых двух параметров становится ненужным.

Итак, когда мы набрали шаблон, например, `C:*.*`, т.е. любой файл с диска `C:`, и нажали кнопку, обработчик сперва обращается к функции `FindFirst` и, если поиск был успешным, о чем свидетельствует нулевое возвращаемое значение, начинает отображение каталога.

Для вывода информации о полученной строке каталога в таблицу используется процедура `analiz`, которую мы разберем чуть позднее.

После ее вызова следует циклическое обращение к функции `FindNext`. Если он успешен, то увеличением свойства `RowCount` к таблице добавляется еще одна строка и снова вызывается процедура `analiz`. Когда поиск станет безрезультатным, следует вызвать процедуру `FindClose` для завершения поиска и освобождения использовавшегося ОЗУ.

Процедура `analiz` достаточно проста. Она обращается к полученной из каталога записи и, преобразовав содержимое ее компонентов в необходимую текстовую форму, заносит в таблицу. Текстовое поле `sr.Name` с именем при этом просто копируется, а `sr.Size`, отвечающее за размер, предварительно преобразуется из числовой формы в строковую.

Несколько сложнее обстоит дело с числовым полем `sr.Attr`, каждый бит которого отвечает за отдельный признак (атрибут) файла. Поэтому биты последовательно выделяются с помощью операции `and` с соответствующей константой файлового атрибута (ниже приводится их таблица, взятая из файла помощи Delphi). Если соответствующий бит не равен 0, то файл данный статус имеет и к строковой переменной `s` добавляется соответствующий текст.

Constant	Value	Description
<code>faReadOnly</code>	<code>\$00000001</code>	Read-only files
<code>faHidden</code>	<code>\$00000002</code>	Hidden files
<code>faSysFile</code>	<code>\$00000004</code>	System files
<code>faVolumeID</code>	<code>\$00000008</code>	Volume ID files
<code>faDirectory</code>	<code>\$00000010</code>	Directory files
<code>faArchive</code>	<code>\$00000020</code>	Archive files
<code>faAnyFile</code>	<code>\$0000003F</code>	Any file

Таблица в особом переводе не нуждается, ибо все слова в ней компьютерно грамотному человеку известны. Видно, что все шестнадцатеричные (об основании говорит символ '\$') константы в ней, за исключением последней, являются степенями двойки, т.е. их двоичное значение содержит только один установленный бит; именно он и будет выделен операцией "И". Что касается последней записи, то она, напротив, содержит единицы во всех используемых для представления атрибутов младших разрядах, так что применение такой битовой маски не повлияет ни на один из битов.

Строго говоря, обо всех этих бинарных деталях можно было и не рассказывать, но подобные методы работы с битовыми полями в Паскале распространены настолько широко, что просто жаль пройти мимо...

При анализе получаемых результатов советуем обратить внимание на следующие детали. Все каталоги (они помечены в таблице надписью `DIR`) формально имеют нулевую длину. Так сделано, чтобы их нельзя было прочесть с помощью стандартных дисковых функций. Добавим, что большинство системных файлов являются скрытыми (`Hidden`), т.е. при стандартных настройках не отображаются. Некоторые (например, `MSDOS.SYS` и каталог `Program Files`) при этом защищены от удаления (`Read Only`).

Изучив имеющиеся в корневом каталоге файлы, попробуйте пройти цепочку каталогов, изображенную на *рис. 6*, для некоторого выбранного вами файла. Надеюсь, данный эксперимент позволит вам на практике закрепить полученные при чтении статьи теоретические знания.

Эксперимент 3. Чтение логических секторов средствами Windows

В последнее время в разговорной речи часто приходится слышать фразу "это отдельная песня". Так вот, чтение сектора дискеты средствами Windows — это даже не отдельная песня, а, наверное, что-то вроде оперной арии. Проблема заключается в следующем. В любом IBM-совместимом компьютере в ROM BIOS имеется² готовая программа чтения секторов дискеты, иначе как обеспечить загрузку с данного устройства? ОС MS-DOS, оперируя логическими секторами, производит пересчет их номеров и затем обращается к BIOS (см. описание этапов 5 и 6). Любой программе в этой ОС также разрешено напрямую читать сектора, используя BIOS. Что касается Windows, то в более новой ОС приложениям DOS ради преимущества разрешено работать с диском напрямую, но для Windows-приложений "из сообра-

² До того, как в компьютер загрузится любой Windows!

жений безопасности” это запрещается. Ситуация во многом напоминает попытку наложить штраф на гражданина, который выносит из леса зеленую веточку ели, подобранную им на месте незаконной вырубке леса!

Поэтому, хотя приведенная в описании этапа 6 программа превосходно работает в “досовском” окне Turbo Pascal, она безнадежно не исполняется в Delphi. Какие же средства рекомендует Windows взамен тех, которые теперь запрещены? Обращение через его штатные драйверы устройств. На первый взгляд все не так плохо: вместо попыток, близких к хакерским, обратиться к “железу” предлагается воспользоваться более удобными “цивилизованными” средствами. Но только на первый взгляд. А на второй и все последующие ситуация не столь радужная. Вместо массы доступных и проверенных временем книг классиков типа Питера Нортона — куца, без единого примера англоязычная помощь (причем для тех, кто *заранее* знает *точные названия* требуемых ему функций!) и безбрежный хаотический Интернет³. Но и это еще не все. В разных версиях Windows обращения к драйверам осуществляются по-разному, т.е. программа для Windows 9X далеко не всегда работает в NT и наоборот! А про то, что программы, которые все же удается где-то разыскать, написаны почти всегда на Си (а не на Паскале) и существенно используют специфические библиотеки этого языка, и говорить не приходится...

Введение получилось чересчур длинным, но иначе читатель не сможет по достоинству оценить ту короткую программу, которая сейчас будет предложена, а также понять, почему ему объяснили не все многочисленные параметры со значением `nil`, как, впрочем, и некоторые другие. Итак, для того чтобы прочитать сектор с дискеты согласно штатной философии Windows, нам предлагается открыть дискету как некий огромный файл, состоящий из последовательности секторов. Те, кто внимательно прочел статью, немедленно поймут, почему такой доступ назван в заголовке логическим. Далее остается просто позиционировать указатель файла на нужное место и читать очередные 512 байт. Плохая наглядность такого подхода будет отчетливо видна, когда используемые функции потребуют от нас задавать все величины в байтах (а не секторах!). Следовательно, подобные действия по своей сути скорее являются некоторой разновидностью блочного чтения (см. вариант С эксперимента 1), чем чтением секторов реального физического устройства.

³ Вот еще пример: в почти девятистотстраничной книге В.Несвижского “Программирование аппаратных средств в Windows” нет приемлемого для нас варианта, хотя там рассказано, как самому написать драйвер.

Приступим к разработке простейшего Windows-приложения в среде Delphi. Прежде всего создадим несколько визуальных компонентов для ввода и вывода информации: однострочное поле `Edit1` для ввода номера логического сектора, кнопка `Button1` для инициализации чтения, многострочное поле `Memo1` для отображения прочитанного содержимого сектора в текстовом виде и метка `Label1` для индикации количества прочитанных байт. Вариант возможного оформления окна показан на *рис. 10*.

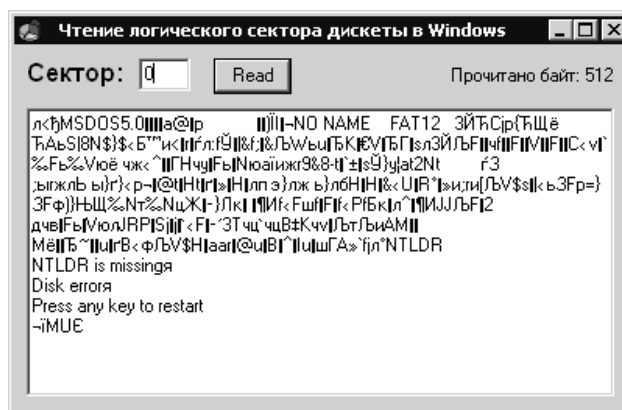


Рис. 10. Вид окна программы для эксперимента 3

Примечание. Очень полезно свойство `Default` кнопки установить в `true`, тогда после ввода номера сектора можно будет просто нажимать `Enter`.

Далее создаем стандартным способом обработчик кнопки `Button1` и вписываем в него следующую программу.

```
procedure TForm1.Button1Click(Sender: TObject);
VAR h: THandle; buf: array [1..512] of char;
    nb: dword;
    sector, i: integer; s: string;
begin
  sector := StrToInt(Edit1.Text); nb := 0;
  h := CreateFile('\\.\A:', GENERIC_READ,
    FILE_SHARE_READ, nil, OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL, 0);
  IF h <> INVALID_HANDLE_VALUE
  THEN BEGIN SetFilePointer(h, 512*sector, nil,
    FILE_BEGIN);
    ReadFile(h, buf, 512, nb, nil);
    CloseHandle(h);
    Memo1.Lines.Clear; s := '';
    FOR i := 1 TO 512 DO
      IF buf[i] <> CHR(0) THEN
        s := s + buf[i];
    Memo1.Lines.Add(s);
  END;
  Label1.Caption := 'Прочитано байт: '+IntToStr(nb);
end;
```

С помощью функции `CreateFile` устанавливается связь с дискетой. Странная запись имени диска `A:` со всевозможными дополнительными служебными символами впереди не должна вас сму-

щать: будучи универсальной функцией, `CreateFile` умеет работать даже с сетевыми адресами, так что все “излишества”, видимо, есть не что иное, как “заготовки” для таких адресов. Всевозможные параметры чтения в скобках при первом знакомстве лучше принять как “научно-медицинский факт”, ибо для обсуждения их смысла и возможных значений потребуется самостоятельная статья, которую, кстати, автор писать не возьмется.

Примечание. Значения с длинными английскими именами типа `GENERIC_READ` — это битовые флаги, о которых подробно рассказывалось в описании предыдущего эксперимента.

Важным результатом выполнения функции является возвращение указателя (`handle`) на “файл”; с помощью значения, лежащего в переменной `h`, будут осуществляться все последующие дисковые операции. На случай, если дисковод “открыть не удалось”, поставлена проверка на недопустимость полученного “хандлера” (`invalid` в компьютерном словаре имеет значение “недопустимый, ошибочный”).

Следующая функция, которая вступает в работу, называется `SetFilePointer`, или по-русски “установить указатель файла”. Она должна установить виртуальный указатель файла-дисковода на нужный байт, пропустив все те, которые входят в предыдущие сектора. Учитывая, что логическая нумерация секторов начинается от 0, то для вычисления смещения достаточно перемножить номер сектора и его размер — 512 байт.

Теперь остается прочесть очередные 512 байт файла. Для этого используется функция `ReadFile`, которая сильно напоминает `BlockRead` (см. эксперимент 1, вариант С). Дополнительный (последний) параметр тоже установим в `nil`, ибо детище Microsoft в нашей задаче прекрасно обходится без него. Вот, собственно, и все — сектор, наконец, прочитан в буфер `buf` и можно завершить работу с дисководом с помощью функции `CloseHandle`.

Остается вывести содержимое буфера в отображаемое на экране поле `Mem01`. Прежде всего оно очищается от старого содержимого с помощью метода `clear`. Далее в строковую переменную `s` заносятся все ненулевые байты; такая дополнительная обработка требуется потому, что все компоненты типа `TMemo` используют нулевой байт как признак окончания текста и игнорируют все последующее. Сформированная в переменной строка добавляется к отображаемому тексту.

Последняя строчка обработчика отображает количество прочитанных байт: при нормальном завершении процесса чтения оно всегда равно 512.

Подчеркнем, что описанная программа может быть названа приложением, т.е. настоящей программой, с огромной натяжкой. Она не производит проверку корректности вводимого номера сектора, не выво-

дит, как это обычно делается в подобных программах, шестнадцатеричные коды байтов и, наверно, не лишена еще кое-каких недостатков. Инициативные читатели, взяв опубликованный вариант за основу, могут самостоятельно усовершенствовать его в соответствии со своими вкусами и потребностями.

Стоит обязательно подчеркнуть, что приведенная программа работоспособна только в Windows NT, 2000 и XP, но в версиях 9X и ME работать не будет.

Если приводимые в статье эксперименты показались вам интересными, можно обратиться к книге [10] и провести следующие дополнительные эксперименты:

- 5.6.1 — логический доступ к сектору дискеты;
- 5.6.2 — физический доступ к сектору дискеты;
- 5.6.3 — чтение секторов CD;
- 5.6.4 — форматирование дискеты;
- 7.10.2 — определение размера кластера;
- 7.10.3 — несколько экспериментов с именами файлов;
- 7.10.4 — изучение расположения файлов на диске.

Некоторые полезные, по мнению автора, дополнительные материалы по поводу многослойного устройства операционной системы можно также найти в статье [11].

Литература

1. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2002, 544 с.
2. Таненбаум Э. Современные операционные системы. СПб.: Питер, 2004, 1040 с.
3. Немет Э., Снайдер Г., Хейн Т. Руководство администратора Linux. М.: Издательский дом “Вильямс”, 2004, 880 с.
4. Гринзоу А. Философия программирования для Windows 95/NT. СПб.: Символ-Плюс, 1997, 640 с.
5. Нортон П. ПК фирмы IBM и операционная система MS-DOS. М.: Радио и связь, 1992, 416 с.
6. Гук М. Аппаратные средства IBM PC. Энциклопедия. СПб.: Питер, 2003, 923 с.
7. Гук М. Дискровая подсистема ПК. СПб.: Питер, 2001, 336 с.
8. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT, AT. М.: Финансы и статистика, 1992, 554 с.
9. Нортон П., Джордейн Р. Работа с жестким диском IBM PC. М.: Мир, 1992, 560 с.
10. Еремин Е.А. Популярные лекции об устройстве компьютера. СПб.: ВHV-Петербург, 2003, 272 с.
11. Еремин Е.А. Коллективное проектирование имитатора операционной системы. / Информатика и образование, 2005, № 7, с. 85–90.

Преподавание программирования с использованием системы автоматической проверки решений

В.А. МАТЮХИН,
Москва

В этой статье мы немного поговорим о том, зачем и как можно учить школьников программированию.

Компьютер — действующее лицо педагогического процесса

Программирование — уникальный предмет. Дело в том, что при изучении почти любого другого предмета учебный процесс строится на взаимодействии ученика и учителя. При изучении же программирования в этот процесс включается еще одно действующее лицо — компьютер.

Что фактически требуется от школьника на первых порах изучения программирования? Взять задачу, которую он отлично понимает, как нужно решать без компьютера (например, задачу поиска минимума среди набора чисел), после чего попытаться осознать, что же он (школьник) делает, решая данную задачу, и сформулировать это словами, а затем перевести этот алгоритм на язык машины и получить программу.

И вот здесь в действие вступает компьютер. Во-первых, если в программе допущены синтаксические ошибки, то компилятор сразу на это укажет. Во-вторых, когда все синтаксические ошибки исправлены, часто программа все равно не работает. И опять же ребенок может увидеть это сам, запустив программу на каком-нибудь примере.

Очень важно, что ученик имеет возможность увидеть, а зачастую и исправить ошибки без всякого вмешательства учителя. Во-первых, в этом случае значительно снижается время получения “обратной связи” школьником (учитель просто физически не имеет возможности с такой скоростью смотреть на работы школьников и указывать на ошибки — как правило, на других предметах школьники сначала в течение какого-то времени выполняют работу, потом сдают и на следующем уроке получают проверенную работу). Во-вторых, когда на ошибку указывает не товарищ, не учитель, а бездушное существо, у школьника исчезает страх ошибку совершить. Вкупе с тем, что можно быстро что-нибудь исправить, снова запустить программу и очень быстро узнать, что и этот вариант неверен (или, наоборот, верен) — школьник довольно активно пробует разные варианты. Тем самым он учится самостоятельно преодолевать проблемы, искать источники ошибок и т.д.

Роль отладчика

Еще один очень полезный инструмент при изучении программирования — отладчик. С помощью него школьник может выполнять программу по шагам и отслеживать значения переменных. Фактически отладчик позволяет школьнику сравнить, что должно происходить при выполнении программы, по мнению школьника, и что происходит на самом деле.

Мне очень нравится сравнение роли отладчика с зеркалом. Когда человек учится танцевать, он делает это перед зеркалом, чтобы видеть, что у него получается. Когда человек учится программировать, фактически он учится излагать свои мысли на языке программирования — абсолютно строгом, формальном языке. Программа — это изложение мыслей школьника о том, что нужно делать для решения задачи. Отладчик позволяет увидеть, где то, что написано в программе, не совпало с тем, что он хотел написать.

Пользоваться отладчиком не всегда просто. Но научить школьников им пользоваться — очень важно. Часто школьнику значительно проще обратиться к учителю с вопросом “а где у меня ошибка?”, и бывает, что учителю достаточно одного взгляда на программу, чтобы эту ошибку увидеть и указать на нее школьнику. Однако обучающий эффект будет значительно больше, если школьник найдет эту ошибку сам.

Система автоматической проверки

Итак, школьник добился того, что программа работает на каком-то примере. То есть вроде бы задача решена. Это далеко не всегда так. В задаче могут быть хитрые случаи, специально спрятанные “грабли” или программа написана так, что работает лишь в случаях специального типа, но именно на таком примере и проверял свое решение школьник.

Теперь, по идее, к школьнику должен подойти учитель и проверить его программу. Запустить ее несколько раз, предложив ей разные входные данные. Однако, во-первых, учитель физически может запустить программу на 2–3 примерах (не больше), и может так почувствовать, что и эти тесты программа пройдет. Во-вторых, входные данные в задаче могут быть достаточно большими, и учителю потребуется довольно много времени на то, чтобы их набить (а ведь надо набить без ошибок!). В-третьих, ответ на этих примерах может быть неоднозначным, и потребуется еще и время на то, что-

бы понять, правильный ли ответ вывела программа школьника или нет. На самом деле все это можно переложить с учителя на компьютер.

Компьютер сам может давать программе школьника наборы входных данных (“подсовывая” ей разные входные файлы или эмулируя ввод с клавиатуры), сам запускать программу школьника и анализировать выданный ей ответ.

Это и есть автоматическая проверка решения. При этом система автоматической проверки — это не универсальная программа, которая может проверять решение любой задачи. Это программа, которая умеет “подсовывать” тесты (т.е. наборы входных данных), запускать решение, анализировать результат и, быть может, как-то вести протокол и поддерживать таблицу результатов. Чтобы система могла проверять некоторую задачу, нужно, чтобы были подготовлены тесты специально для этой задачи.

Причем, в отличие от учителя, компьютер может проверить решение очень качественно — за минуту можно прогнать до сотни разных тестов (в том числе и с достаточно большими по объему входными данными). Если все они прошли, то программа признается правильной. Если какой-то тест не прошел — то, очевидно, программа содержит ошибку.

Дальнейшая тактика может быть разной. Школьнику может выдаваться этот тест (т.е. набор входных данных), чтобы, пользуясь отладчиком, школьник мог найти и исправить ошибку. Так правильно поступать на первых порах изучения программирования. А можно не выдавать тест, а лишь сообщить о том, что в программе имеется ошибка (часто принято сообщать номер теста, чтобы школьник мог сравнить, к чему привели его исправления по сравнению с предыдущей попыткой сдать задачу — программа стала работать хуже и проходить меньше тестов, или, наоборот, ему удалось исправить ошибку, из-за которой программа не работала в прошлый раз, и теперь нужно искать еще какие-то ошибки). В такой ситуации школьник учится анализировать задачу, выделять из нее ключевые случаи, понимать, что в его программе может приводить к ошибке. Это очень непростое, но очень важное умение.

Требования к программам

Для того чтобы программы могли проверяться автоматически, они должны удовлетворять некоторым формальным требованиям. В частности, входные данные им даются в строго определенном формате, выходные данные программа также должна выдавать в строго определенном формате. Например, если программа работает абсолютно правильно, но выдает ответ — два числа — не в требуемом, а в обратном порядке, то такая программа не будет принята.

Однако, как показывает практика, школьники через некоторое время привыкают к такого рода формальным требованиям, и это почти перестает вызывать у них проблемы. Более того, как мне кажется, научить школьников строго следовать некоторой формальной спецификации — это тоже одна из целей обучения школьников, и такой навык пригодится им практически в любой сфере деятельности.

Иногда, правда, при сдаче решений системе автоматической проверки возникают очень специфические проблемы. Например, когда ребенок запускает программу из под среды разработки, все переменные автоматически обнуляются и программа работает верно, а при запуске под тестирующей системой в какой-нибудь ячейке памяти оказывается не ноль, и программа не работает из-за того, что не все переменные в ней инициализируются. Но это уже, скорее, проблема не системы проверки, а программирования в целом. И умение не забывать про такие детали, а тем более — умение понимать, из-за чего может возникнуть проблема, находить и исправлять ошибку даже в том случае, когда воспроизвести ее не удастся, — это очень важное качество профессионального программиста. И, в общем, никуда от таких деталей не деться — ведь программирование складывается, в частности, и из этого.

Авторитет системы

Как уже говорилось, чтобы можно было проверять решения некоторой задачи, должны быть подготовлены тесты для именно этой задачи. Кроме того, когда ответ в задаче неоднозначен, часто требуется написание программы, которая по входным данным и выданному проверяемым решением результату устанавливает его правильность. Это непростая работа, которая требует специальных умений. В последнее время на сайтах (преимущественно посвященных олимпиадам по программированию — т.к. технология автоматической проверки решений изначально применялась именно на олимпиадах по программированию и лишь потом была перенесена на учебный процесс) накоплена довольно большая база задач с тестами, которые можно использовать. Это значительно облегчает подготовку к занятиям. Однако часто требуется адаптация тестов к их использованию под конкретную тестирующей системой.

Важно, чтобы тесты не содержали ошибок. Первая реакция школьника на сообщение системы об ошибочности решения — “у меня все правильно, это у вас неправильные тесты”. И здесь то, что на первых порах школьник получает сам тест, на котором его программа не работает, очень важно. Он начинает смотреть этот тест и действительно находит ошибку в своей программе. Так формируется “авторитет” системы среди школьников.

Если же часто будет оказываться, что решение школьника абсолютно правильное и он был вынужден (из-за ошибок в тестах и в проверяющей программе) искать в нем несуществующую ошибку и потратить на это очень много времени, то это, наоборот, работает на разрушение авторитета. Если же у системы не будет авторитета, убедить школьника искать ошибку и добиваться прохождения всех тестов будет очень-очень сложно.

Еще один важный момент — это то, что система тестов для задачи должна быть полной. То есть, если в решении есть ошибка, то она обязательно должна тестами находиться. Например, если задача формулируется “дано до 100 чисел, найти минимальное из них”, то обязательно должен быть тест именно на 100 чисел (потому что если школьник, например, несколько некорректно работает с массивом и обращается к $(N + 1)$ -му его элементу, то эта ошибка проявится лишь на таком тесте). Пол-

нота системы тестов также работает на укрепление “авторитета” системы — получить от системы сообщение “задача решена” становится престижно.

И, сдавая свое решение на проверку, школьник в этой ситуации уверен, что его решение будет полноценно проверено (в противовес ситуации, когда школьник долго-долго возился с программой, подошел учитель, запустил ее на одном тесте и сказал: “Молодец, Вася!”).

Еще один аспект автоматической проверки — полная беспристрастность. Все решения проверяются абсолютно одинаково. У системы не бывает “любимчиков”, которым прощается то, что не прощается другим, не бывает нелюбимых учеников.

Роль учителя

Может показаться, что при использовании системы автоматической проверки учитель вообще не нужен. Это не так.

Благодаря использованию системы автоматической проверки во время урока у учителя освобождается время, и теперь он может больше времени посвятить тем ученикам, которым нужна его помощь.

Учитель нужен, чтобы помогать ученикам искать нетривиальные ошибки в программах, когда им не удается найти их самостоятельно.

Учитель нужен, чтобы давать подсказки, наталкивающие предложения ученикам, когда у них что-то не получается.

Учитель нужен, чтобы с учениками, решившими задачу, обсуждать, какие еще бывают способы ее решения, как решить задачу более красиво.

Наконец, полезно время от времени после сдачи программы системе автоматической проверки заставлять школьников сдавать программу учителю. Во-первых, при автоматической проверке не отслеживается, чтобы код программы был написан красиво. Конечно, понятие красоты — очень субъективное. Однако нужно требовать от школьников выделения логических блоков, циклов и т.д. Как уже говорилось, программа — это отражение мыслей школьника. Наводя “красоту” в программе, на самом деле мы наводим порядок в его мыслях. А пока в программе каша — в голове будет то же самое.

Кроме того, изучая код программы, можно увидеть и показать школьнику, что и как можно делать красивее и проще.

Чему учатся школьники, изучая программирование

Прежде всего школьники учатся очень точно и очень формально излагать свои мысли.

Во-вторых (и об этом тоже уже говорилось), они учатся работать самостоятельно, экспериментировать, искать пути решения возникшей проблемы.

Школьники учатся анализировать задачу, выделять, какие бывают ключевые случаи, где можно допустить ошибку.

Школьники учатся тестировать свои программы, а значит, учатся относиться критически к продуктам своего труда. Рассмотрим ситуацию, когда школьник получил от системы ответ, что в программе есть ошибки, но не получил теста, на котором эта ошибка происходит. Тог-

да ему прежде всего нужно подобрать тест, на котором его программа работает неправильно (а дальше с помощью отладчика найти, что же именно в ней неверно). Возникает ситуация, когда школьник радуется, если ему удастся найти тест, на котором его программа не работает. Согласитесь, что в педагогическом процессе такая ситуация бывает не часто.

Написание программ позволяет школьнику лучше понять, как устроен и работает компьютер, какие задачи с его помощью можно решать.

Решая уже более сложные задачи, участвуя в олимпиадах, школьник знакомится с теорией алгоритмов. Учитя применять стандартные алгоритмы и адаптировать их для нестандартных задач. Учитя строить математическую модель по словесному описанию условия задачи. Учитя оценивать сложность алгоритма и скорость работы программы (часто до того, как начать эту программу писать).

Участвуя в командных олимпиадах, школьники учатся работать в команде, распределять обязанности, совместно работать на общий результат.

Мне кажется, что большинство этих навыков будут полезны, даже если в дальнейшем жизнь школьника никак не будет связана с программированием.

Благодарности

Эта статья написана по опыту преподавания в московской гимназии на Юго-Западе № 1543, а также по опыту проведения Летних компьютерных школ (www.olympiads.ru/sis). Многие идеи, которые я использую в своей практике, принадлежат моему учителю информатики декану Вятского государственного гуманитарного университета Окулову Станиславу Михайловичу, заведующей кафедрой информатики СУНЦ МГУ Андреевой Елене Владимировне, руководителю Мытищинской школы программистов Шедову Сергею Валерьевичу. Родоначалником систем автоматической проверки в России был Суханов Антон Александрович — многие заложенные им идеи до сих пор используются в архитектуре проверяющих систем. Мне хотелось бы выразить этим людям огромную признательность. А также поблагодарить Чернова Александра Владимировича, Станкевича Андрея Сергеевича, Корнеева Георгия Александровича, Умнова Владимира Игоревича и Каменских Дениса Вячеславовича, чьими тестирующими системами мне неоднократно приходилось пользоваться.

Ссылки

Тестирующие системы:

1. www.ejudge.ru
2. olympiads.ru/school/system/download/t-run/index.shtml
3. olympiads.ru/school/system/download/olympiads/index.shtml

Сайты, где можно найти задачи с тестами:

1. www.olympiads.ru
2. neerc.ifmo.ru/school
3. www.informatics.ru

Задачи для школьных уроков информатики можно найти здесь: www.olympiads.ru/problems

Что и как мы оцениваем?

В.АРСЛАНЬЯН,

Москва

Система оценки знаний вызывает, пожалуй, наибольшие споры между участниками образовательного процесса. Связано это со многими причинами.

Во-первых, школьный учитель пользуется четырехбалльной системой отметок: 2, 3, 4, 5. С ее помощью очень сложно дифференцировать знания учеников, поэтому он вводит еще дополнительные градации: “+” и “-”. Но ребятам непонятно, чем отличается “четверка с плюсом” от “пятерки с минусом”. И ученик просит: “Поставьте мне лучше “5-”, чем “4+”.

Во-вторых, учитель знает, каким образом он оценивает результаты обучения учеников: за что он ставит “3”, за что — “4”, а за что — “5”, и думает, что ученик это тоже знает. Поэтому он не проговаривает систему требований и критерии выставления отметок, не говорит, что для того, чтобы получить “3” за четверть, необходимо выполнить такой-то объем заданий, для “4” — такой-то и т.д.

В-третьих, учитель нередко забывает, что отметка предназначена не только для того, чтобы констатировать результат освоения определенного объема школьной программы, но еще и для того, чтобы мотивировать ученика на дальнейшие успехи и достижения. Отметка чаще выступает как карательный инструмент, а не стимулирующий и развивающий; а еще хуже — как ярлык, который навешивается ученику на все время обучения в школе, ведь дети часто воспринимают отметку не как результат труда, а как показатель ума. Поэтому родители, оправдывая невысокие школьные баллы и дабы поднять статус своего ребенка в глазах окружающих, говорят: “Он умный, но ленивый”, тем самым снимая “ярлык” глупого и бездарного.

Поэтому давайте сформулируем несколько самых важных правил, касающихся выставления школьных отметок.

Правило 1. Ученик должен знать критерии выставления оценок.

Он должен знать, за какие работы, какого уровня сложности и какого объема он получит ту или иную отметку. Каждый учитель оценивает по-своему. Один, например, за арифметическую ошибку при решении уравнения (по невнимательности) может просто поставить минус к отметке, а другой снижает отметку на балл. Один считает, что для

получения за четверть тройки по геометрии ученик должен выучить и ответить формулировки теорем, а другой — что необходимо “сдать” все теоремы с доказательствами и уметь решать несложные задачи. Причем критерии оценки знаний учащимся следует периодически напоминать (неплохо и на стенд вывесить) — учащиеся должны понимать, что это не сиюминутные требования, зависящие от настроения учителя, которые завтра могут измениться, а продуманная система, которую они должны принять и усвоить.

Правило 2. Ученик должен иметь возможность исправить неудовлетворительные оценки.

Если ученик имеет подряд несколько двоек, каждую из которых ему нужно “закрыть”, он вряд ли мобилизует свои силы, чтобы их исправить, — это очень сложно. При этом учитель говорит ученику, что тот совершенно не хочет учиться, ему все равно, какие у него отметки и т.п. Но скажите, какой взрослый не опустит от безнадежности руки в такой ситуации? А ребенок? Поэтому не стоит “заваливать” нерадивых учеников двойками. Нельзя пользоваться только карательными мерами. В арсенале учителя должны быть и другие методы воздействия на учащихся. Например, некоторые учителя никогда не ставят “2” в журнал, а сообщают, когда можно сдать эту тему, тем самым все-таки вынуждая ученика понять, выучить, освоить те или иные навыки.

Правило 3. Отметка должна стимулировать ученика на дальнейшие успехи и вложение своих сил.

Традиционно отметка показывает не только уровень усвоенности знаний и сформированности навыков, но и труд, а его нельзя обесценивать. Например, если школьник проболел, пропустил изучение некоторых тем, но дома самостоятельно разобрал теорию и решал задачи, может быть, не все понял и усвоил. А в первый день после болезни он попадает на контрольную работу, вряд ли в данном случае возможно оценивать знания ученика и относиться к нему со всей строгостью. Нужно поддерживать его стремление заниматься в тот период болезни, иначе в следующий раз он и не сядет за учебник, а будет “болеть в свое удовольствие”. Учитель не должен выступать в роли статиста, который, проводя контрольные срезы, отмечает, кто усвоил материал, а кто — нет. Надо помнить, что

учитель решает не только задачи обучения, но и задачи воспитания.

Правило 4. Школьная отметка — это не показатель ума или личностных качеств ученика. Поэтому учитель, комментируя оценки за выполненную самостоятельную или контрольную работу, должен подчеркнуть не только те моменты, с которыми ученик не справился, над которыми он должен еще поработать, но и то, что ему удалось. Акцент с констатации неуспеха, а он часто ассоциируется с интеллектом или личностными качествами, должен быть перенесен на личный вклад ученика. Но и о том, что пока не получилось, необходимо говорить в конструктивном ключе, например, “чтобы справиться с этой задачей, надо разобрать задачи № 102, 105 и 108”.

Правило 5. Избегайте стереотипов! К сожалению, нередко на учителя оказывают влияние стереотипы, например, что сильный ученик не справился с задачей случайно или по невнимательности, а слабый потому, что ленивый или ему не хватило способностей. А уж если троечник решил задачу, то “непреренно списал”. Поэтому учитель иногда не

выставляет успешному ученику плохую отметку, а слабому хорошую. За одни и те же ошибки разным ученикам выставляются разные оценки. Это вызывает у ребят обиду и чувство несправедливости. Мы боимся хвалить — “чтобы не зазнался, не разленился”, “чтобы не сглазить” и т.п. Но поддержка и похвала действуют гораздо сильнее, особенно на тех ребят, которых в силу их успеваемости вообще редко хвалят.

Правило 6. Не сравнивайте учебные достижения ученика с успехами других учащихся.

Нередко учителя (да и родители тоже) сравнивают успехи одного ребенка с успехами его одноклассников, друзей, братьев или сестер. Но если мы хотим, чтобы у ребят было желание учиться, мы должны сравнивать его достижения только с его же собственными, например, “Эту работу ты выполнил гораздо лучше, чем прошлую” или “Мне кажется, что прошлый раз ты лучше подготовился к контрольной работе”.

Таким образом, у нас появится возможность нашими отметками способствовать развитию ученика, мотивировать его к следующим достижениям и личностному росту.



Фестиваль исследовательских
и творческих работ учащихся

«Портфолио»



Издательский дом «Первое сентября» объявляет о проведении в 2006/07 учебном году Второго всероссийского фестиваля «Портфолио» и приглашает принять в нем участие учащихся учреждений начального, среднего и дополнительного образования и их педагогов.

Участвуя в фестивале, учащиеся могут формировать общедоступное

портфолио своих работ. Также формируется портфолио педагога, в которое входят работы учащихся, выполненные под его руководством.

Все материалы будут опубликованы. По результатам фестиваля будут изданы: книга — сборник тезисов (описаний) работ и компакт-диски с полными версиями работ. Полные версии работ также публикуются на сайте фестиваля <http://portfolio.1september.ru>, который является одним из разделов сайта Издательского дома «Первое сентября» — самого популярного образовательного ресурса русскоязычного Интернета. **Книги и компакт-диски будут высланы всем участникам. Все ученики и их руководители будут отмечены дипломами.**

Заявки на участие принимаются с 1 июля по 15 декабря 2006 г. Заявки можно подавать на сайте <http://portfolio.1september.ru> или по почте, используя бланки, публикуемые в газетах.

РАЗДЕЛЫ ФЕСТИВАЛЯ

- Научно-исследовательские работы:
 - Астрономия и космонавтика
 - Биология
 - География
 - История, археология
 - Искусствоведение
 - Лингвистика
 - Литературоведение
 - Математика
 - Религиоведение
 - Экономика, социология и право
 - Физика
 - Химия
 - Здоровье человека, психология
 - Физкультура и спорт
 - Экология
- Художественное творчество
- Техническое творчество
- Информационные технологии
- Литературное творчество
- Музыкальное творчество
- Краеведение
- В помощь учителю (дополнительный раздел)

НАЧАЛКА

газета-клуб для всех,
кто учит информатике
маленьких детей



№ 12 (16–30 сентября)

Алгоритмическая гимнастика: человечки “бумажные” и заводные

Н.Д. Шумилина,
Тверская гимназия № 6

Движение — жизнь! Психологи утверждают, что в период самого раннего детства физическая активность значительно влияет на интеллектуальное развитие человека. Но и для взрослого человека физическая нагрузка — элемент воздействия на психическое состояние, способ снятия эмоционального напряжения, стресса.

Однако, уважаемые коллеги, дадим себе ответ на вроде бы частный вопрос: выработана ли у нас привычка двигательного перерыва во время напряженной умственной работы? Вряд ли многие ответят положительно. А жаль. Знаем и не реализуем. Почему? Ответы будут разные. А вот результаты похожи. Подолгу, без перерыва сидим за столом или за компьютером.

Культуре умственного труда можно обучить, ею можно овладеть. И, конечно, это лучше делать, начиная с младших классов школы.

Физкультминутка или физкультурная пауза — необходимые элементы урока в начальной школе, снимающие утомление, вызванное учебной деятельностью. Если упражнения выполняются в течение примерно одной минуты, такой отдых называют физкультминуткой, если активный отдых требуется в течение нескольких минут, такую форму занятий специалисты по физической культуре называют физкультпаузой.

Из всех известных методик обучения информатике в начальной школе особое внимание вопросу физкультурной разминки на уроке уделили авторы курсов [2, 3, 4].

В поурочных разработках [4] можно найти множество небольших, простых и понятных упражнений, сопровождаемых стихотворениями, которые с удо-

вольствием воспринимают малыши. В этих методических разработках физкультминутке отведено определенное, логичное место на уроке: после объяснения нового материала, перед работой на компьютере.

В работе [3] физкультминутки названы просто и ясно: зарядка. Составлено отдельное приложение с интересными стихотворными текстами для проведения упражнений не только для разминки тела в целом, но и специально для пальцев рук.

В курсе [2] сделана интересная попытка интеллектуализации, если так можно выразиться, физкультурной паузы. Во многих случаях действия детей зависят от анализа предлагаемой порции информации, фактически это игры (“Угадай предмет”, “Бывает — не бывает”, “Найди по адресу”, отгадывание загадок и др.). Но не всегда предлагаемых действий достаточно для того, чтобы все дети могли размяться. В некоторых случаях встает только первый угадавший, в некоторых дети могут встать (или хлопнуть в ладоши) только в определенном случае.

Давайте посмотрим на столь полезный фрагмент урока со стороны специфики предмета — информатики. Ведь одно из главнейших понятий курса информатики — алгоритм, вряд ли кто из коллег не согласится с этим утверждением. Но алгоритм — это последовательность **действий!** Так попробуем воспользоваться этим замечательным фактом!

Вашему вниманию предлагается вариант упражнений с целью сочетания движения и алгоритмических управляющих конструкций: следование, цикл, ветвление. Наверное, это не самая простая гимнастика. Но если упражнения не слишком усложнять, то интеллектуальная составляющая не будет перегружать детей. Они научатся воспринимать закодированные упражнения “с листа”, выполнять движения вслед за учителем или в зависимости от некоторых условий. То есть разовьется определенный навык выполнения действий по порядку. В подобной коллективной работе

легко исправить ошибку, и, как всегда, всем захочется выполнить задание лучше других!

Для демонстрации действий “бумажных” человечков вводятся элементы формализации, обозначения. Объекты — руки, П — правая, Л — левая.

Движения задаются с помощью стрелочек. Договоримся, что выполнять движение надо так, как видим в записи.

→; ↑; ↓; ←; ↑↑; ↓↓; ; ↑↓, ↓↑, ↖; ↗; ↘; ↙; ⇌

Человечки “бумажные”, потому что они живут на бумажной поверхности! Поэтому движения совершаются в одной плоскости, т.е., например, движение ↑↑ означает подъем рук вверх через стороны.

Если следующее положение можно выполнить через верхнее или нижнее положение рук, то по умолчанию предполагается движение через нижнее.

Исходное положение: ↓↓, т.е. руки вниз.

Запись упражнения можно сделать на доске, можно приготовить плакат. В любом случае перед глазами детей должна быть формализованная запись упражнений.

Однако формализовать все движения трудно. Естественны и необходимы упражнения, которые повторяются после показа, по “живому” образцу. Например, это могут быть движения впереди—сзади, вращательные и др. Их труднее изобразить графически простыми значками. В тексте таким трехмерным движениям соответствуют названия, оформленные с помощью такого шрифта.

Следование и Цикл

Обучение

Правую руку поднимаем в сторону и опускаем вниз

1. П →
2. П ↓

“Машем правым крылом”

Повторить заданное количество раз подъем и опускание правой руки.

ПОВТОРИТЬ 3

1. П →
2. П ↓

или

ПОВТОРИТЬ 3 (П →; П ↓)

Левую руку поднимаем в сторону и опускаем вниз

1. Л ←
2. Л ↓

“Машем левым крылом”

Повторить заданное количество раз подъем и опускание левой руки.

ПОВТОРИТЬ 3

1. Л ←
2. Л ↓

или

ПОВТОРИТЬ 3 (Л ←; Л ↓)

Руки одновременно поднимаем в стороны, потом опускаем вниз

1. ← →
2. ↓↓

“Машем крыльями”

Повторить заданное количество раз одновременный подъем и опускание рук.

ПОВТОРИТЬ 3

1. ← →
2. ↓↓

или

ПОВТОРИТЬ 3 (← →; ↓↓)

Руки по очереди поднимаем в стороны, потом опускаем вниз

1. П →
2. Л ←
3. П ↓
4. Л ↓

“Регулировщик”

Повторить заданное количество раз поочередный подъем и опускание рук.

ПОВТОРИТЬ 3

1. П →
2. Л ←
3. П ↓
4. Л ↓

или

ПОВТОРИТЬ 3 (П →; Л ←; П ↓; Л ↓)

Упражнения на самостоятельное воспроизведение движений по записи

“Круги”

ПОВТОРИТЬ 3

1. ↑↑
2. ↓↓

или

ПОВТОРИТЬ 3 (↑↑; ↓↓)

“Маятник”

ПОВТОРИТЬ 3

1. ⇌
2. ⇌

или

ПОВТОРИТЬ 3 (⇌; ⇌)

“Снежинка”

1. П ↘

2. Л ↙

3. П →

4. Л ←

5. П ↗
6. Л ↖
7. П ↑
8. Л ↑

“Волна”

1. П →
2. Л ←
3. П ↑
4. Л ↑
5. П →
6. Л ←
7. П ↓
8. Л ↓

“Волны”

ПОВТОРИТЬ 3

1. П →
2. Л ←
3. П ↑
4. Л ↑
5. П →
6. Л ←
7. П ↓
8. Л ↓

или

ПОВТОРИТЬ 3 (П →; Л ←; П ↑; Л ↑; П →; Л ←; П ↓; Л ↓)

Упражнения, имитирующие стили плавания кроль, баттерфляй, брасс.**Упражнения, имитирующие работу лыжника палками (одновременный и попеременный ход).**

Далее можно задавать более сложные последовательности команд.

“Приземление”

1. ПОВТОРИ 2 (П ↑; Л ↓)
2. ПОВТОРИ 2 (П ↑; Л ↑; П ↓; Л ↓)
3. П →
4. Л ←
5. П ↓
6. Л ↓







“Марионетка”

1. ← →
2. ПОВТОРИ 2 (П ↙; Л ↘; П →; Л ←; П ↘; Л ↙; П →; Л ←)
3. П

“Заводные теловетки”

Воспользуемся идеей [1, с. 75], но реализуем ее немножечко по-другому. Объясняются позы (схема на доске). Начнем с двух поз, далее количество будем увеличивать, т.е. упражнение можно усложнять до любой степени сложности. Называются по очереди

номера поз. Требуется сразу за названным номером выполнить позу. Последовательность номеров циклически повторяется несколько раз, в это время запоминается последовательность поз.

1	2	3	4	5	6
					

Словесное описание поз:

1. стоя, руки вниз;
2. присед, руки вниз;
3. присед, руки вперед;
4. присед, руки в стороны;
5. стоя, руки в стороны;
6. стоя, руки вперед.

Далее схема стирается. Первый вариант: называется и выполняется номер позы. Второй вариант: использование хлопков как сигнал смены одной позы на следующую, без названия номера.

Начинать можно с простых вариантов, состоящих из двух или трех поз. По желанию можно усложнять. Например:

на 2: руки в стороны, вниз (позы 5, 1).

на 3: руки в стороны, вперед, вниз (позы 5, 6, 1).

на 4: присесть, руки вперед, встать, руки опустить (позы 2, 3, 6, 1).

на 5: по картинке (позы 2, 3, 4, 5, 1).

на 6: по картинке (позы 2, 3, 4, 5, 6, 1).

Прекрасное упражнение, развивающее внимание, координацию! А по алгоритмической сути — великолепное упражнение, задающее последовательность нескольких шагов, команда, которые должны точно выполняться.

“Высший пилотаж”, наиболее сложный вариант — выполнение упражнения каноном. Для этого необходимо разбить детей на группы (ряды, колонки), каждая из которых выполняет упражнение с отставанием от соседней группы на одну позу.

Интересно поработать и со стихотворным сопровождением. Например, использовать упражнения [2], немного преобразовав их.

“Заводные человечки-1”

1. прыг-
2. скок;
3. руки вверх,
4. тянемся до потолка;
5. наклоняемся слегка;
6. выпрямились,
7. руки вниз.

“Заводные человечки-2”

На раз всем руки вверх поднять.

На два — присесть,

на три — всем встать,

Четыре — всем носки достать и снова встать, уже на пять ☺. И, конечно, можно придумать своих заводных человечков!

“Японская машинка”

Базовый вариант описан в [1, с. 73]. В этой игре можно сочетать повторяющиеся команды с элементами передачи управления.

В основе лежит не сложное, ритмичное, легко запоминаемое и воспроизводимое упражнение.

Участники игры, сидя за партами, синхронно и ритмично (темп задает ведущий) повторяют 4 команды:

1. хлопок руками;
2. хлопок по коленям;
3. щелчок пальцами правой руки;
4. щелчок пальцами левой руки.

После усвоения последовательности действий ведущий (учитель) может попробовать менять темп игры по ходу работы “машинки”.

Для большей двигательной активности можно параллельно с действиями рук вставать и садиться на стул (на каждый 4 действия).

Следующее упражнение состоит в том, что каждый из участников получает свой номер. Первый ведущий — учитель. Его номер один. “Машинка” начинает работать. Ведущий на щелчки сначала называет свой номер, один-один, “представляется”, а на следующие щелчки называет сначала свой номер, а потом номер следующего ведущего, например, один-два, передает управление. Следующий ведущий сначала “представляется”, потом опять передает управление. И так далее.

Для начала можно управление передавать по порядку, потом — с конца, при усвоении игры — произвольно.

При освоении циклических действий имеет смысл (именно для понятия циклической последовательности действий) поработать с упражнениями, сочетающими цикл и линейную последовательность в двух вариантах, подкрепленных соответствующими записями. Один вариант — цикл, записанный в свернутом виде, с помощью слова ПОВТОРИ, другой вариант — линейная, развернутая во времени и выполнении запись. Зачем это нужно? Опыт показывает, что не все дети сразу понимают строгость последовательности действий в алгоритме, т.е. свойство дискретности.

Например, запись:

↑; ↓; ↑; ↓; →; ←; ↑; ↓;

некоторые школьники интерпретируют так:

ПОВТОРИ 3 (↑; ↓); →; ←

Практически в каждой учебной группе обнаруживается подобный вариант “понимания”. Именно для того, чтобы с начальной школы суть циклических дейст-

вий была понятна, очень полезно эти действия предложить детям ощутить на себе и соединить их с соответствующей формализованной записью.

Этот материал более пригоден именно в процессе изучения конструкции цикла. С обсуждением того, как можно записать последовательность действий с повторениями? Что будет, если команды переставить, как изменится выполнение? Какие изменения проще сделать, чтобы упражнение включало большее число повторений?

Если учитель почувствовал готовность детей к использованию подобных упражнений в зарядке, можно это сделать. Если сложно — не стоит включать. Иначе может получиться, как с сороконожкой, которая задумалась, как она ходит, и остановилась! А в процессе обучения циклическим действиям использовать этот прием, безусловно, полезно.

Запись может выглядеть, например, так:

Полетели!

- | | | |
|-------|---|----------------------|
| 1. П→ | } | 1. ПОВТОРИ 2(П→; П↓) |
| 2. П↓ | | |
| 3. П→ | | |
| 4. П↓ | | |
| 5. П↑ | | 2. П↑ |
| 6. П→ | | 3. П→; |
| 7. П↓ | | 4. П↓ |

Полетели по-другому!

- | | | |
|-------|--------|----------------------|
| 1. П→ | 1. П→; | |
| 2. П↓ | 2. П↓ | |
| 3. П↑ | 3. П↑ | |
| 4. П→ | } | 4. ПОВТОРИ 2(П→; П↓) |
| 5. П↓ | | |
| 6. П→ | | |
| 7. П↓ | | |

Ветвление

Для начала хороши несложные упражнения, в которых используется слово “Если”:

Задания можно построить, опираясь на зависимость от действий учителя. Условий может быть множество, например, если учитель:

- поднял одну или две руки вверх или в стороны,
- поднял согнутую одну или две руки в локтях,
- хлопнул в ладоши 1, 2 или 3 раза,
- опустил руки,

— то дети выполняют определенные действия. Но удобнее использовать условием хлопок. Это хороший сигнал к изменению действий.

“Внимательные Заводные теловетки”

Если учитель показывает движения рук: любые, например, в стороны, вверх, вперед, их надо повторять. По хлопкам — прыгать (или приседать).

Если ведущий (в дополнение к предыдущему упражнению) отвернулся, присесть. Но после такого действия следует хлопнуть, иначе детям не видно, что делает ведущий!

“Двойное Эхо”

Если учитель хлопнул один раз, дети хлопают два. Если два, то четыре. Но ведь можно задавать различный ритмический рисунок! А это уже просто безграничное поле сочинительства.

“Ветвящиеся” упражнения сложнее линейных и циклических, поэтому имеет смысл использовать в упражнении один вариант ветвления. Однако есть замечательное упражнение, сочетающее два условия, а очень простыми правилами. Рекомендую:

“Руки — ноги” [1, с. 72]

Один хлопок — команда рукам: их надо поднять или опустить, т.е. изменить их положение. Два хлопка — команда ногам. Аналогично: нужно встать или сесть.

Трудное упражнение для безошибочного выполнения! Понять эту трудность можно: сначала идентифицируется задаваемое действие (для рук или для ног). Далее необходимо проанализировать ситуацию (что в данный момент происходит с руками или ногами) и соответственно изменить положение на иное. Поэтому начинать надо с медленного темпа, постепенно его увеличивая. Это упражнение очень хорошо помогает “проснуться” на первом уроке ☺. Поэтому можно именно первый утренний урок начинать с него.

Проводить занятия с использованием элементов алгоритмики в физкультурной паузе можно по-разному, с учетом специфики курса, класса, урока. Но на первом этапе, безусловно, надо обучить детей действиям и порядку их выполнения так, чтобы все поняли и освоили их выполнение. В дальнейшем можно движения усложнять, комбинировать, менять ритм выполнения. Однако можно не ограничиваться только этим подходом. Можно построить работу с физкультурными упражнениями аналогично работе с задачами при изучении алгоритмических разделов информатики. А именно, построить работу на разных уровнях сложности, например: выполнение, поиск ошибок, разработка.

Выполнение алгоритма — это первое по сложности задание. Однако, согласитесь, не всегда оказывается так просто — выполнить заданный (или задаваемый по ходу) порядок действий. Именно этот вариант работы был рассмотрен выше.

Более высокий уровень соответствует работе в усложненных условиях. Учитель специально ошибается, а ребенок должен выполнять упражнение правильно! Этот способ часто используется в модификации: учитель совершает ошибку в действиях, но произносит верную команду.

Аналогично в варианте работы с “бумажными” человечками дети должны внимательно следить за записью и требуемыми движениями и не поддаваться на уловки ведущего. Конечно, такой прием работы следует применять на этапе уверенного выполнения группой детей того или иного упражнения.

Можно попросить детей придумать упражнения для “бумажных” человечков — записать их с помощью обозначений. Конечно, запись должна соответствовать принятой алгоритмической записи: указан порядок команд, на строке записана одна команда. Введение в практику урока упражнений, придуманных учениками, это дополнительная мотивация, которая поддержит интерес детей к подобной деятельности. Этот прием, в большей степени нацеленный именно на освоение понятия алгоритма и способов его записи, дает не оперативный, а отсроченный результат. Однако любому ученику будет приятно, если учитель (после проверки) доверит ему провести физкультурную паузу по собственному сценарию.

Подобные интеллектуальные упражнения можно использовать при пропедевтике, при обучении, а также при повторении алгоритмических разделов информатики. Физкультпауза в каком-то смысле — элемент игры, а играть любят все. Такая форма приобщения к серьезным вопросам обучения — дополнительный путь к успеху в освоении важнейшего раздела информатики. Применять этот подход можно в сочетании с известными и принятыми формами двигательной активности детей на уроке.

Приведенные упражнения, конечно, не исчерпывают всех возможностей сочетания движения человека и команд алгоритма. Тем более не исчерпывают всего многообразия проведения физкультпаузы. Однако автор уверен в безграничных творческих способностях учительства. Поэтому приведенные идеи могут найти другие реализации, другие интерпретации, пусть они получают свое развитие в вашем творчестве. Пусть будет так, как интересно вашим ученикам и вам, уважаемые коллеги.

Литература

1. Букатов В.М., Еришова А.П. Я иду на урок: Хрестоматия игровых приемов обучения: Книга для учителя. М.: Первое сентября, 2002.
2. Горячев А.В., Волкова Т.О., Горина К.И. Информатика в играх и задачах. 1-й класс. Методические рекомендации для учителя. М.: Баласс, 2003.
3. Могилев А.В., Булгакова Н.Н. Методические рекомендации к учебному комплексу “Мир информатики”. Смоленск: Ассоциация XXI век, 2005.
4. Тур С.Н., Бокучава Т.П. Методическое пособие по информатике для учителей 2–4-х классов общеобразовательных школ. СПб.: БХВ-Петербург, 2005.

В мир информатики

77 (16—30 сентября)

Газета для пытливых учеников
и их талантливых учителей

Вычислительная машина с автоматическим управлением

В этой статье мы опишем основные принципы устройства и работы вычислительной машины с автоматическим управлением (примерами такой машины являются современные ЭВМ, в т.ч. персональные компьютеры). Но сначала рассмотрим некоторый вычислительный процесс, осуществляемый человеком. Для решения задачи у него имеются специальное счетное устройство — арифмометр, на котором можно выполнять расчеты, и алгоритм решения некоторой задачи (набор инструкций, который он должен выполнить). Руководствуясь алгоритмом, человек-вычислитель совершает процесс, в котором происходит прием, хранение, переработка и выдача некоторых сведений (некоторой информации). Обычно эти сведения вычислитель изображает (записывает) на бумаге с помощью цифр, букв или других символов. Совокупность этих символов принято называть *алфавитом*. Например, в алгебре, как известно, применяется алфавит, в который входят, кроме букв, цифры, знаки алгебраических операций, скобки и т.д.

Для вычислительного процесса, происходящего с участием человека-вычислителя, характерно наличие следующих факторов (см. рис. 1).

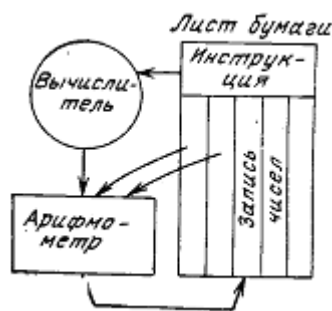


Рис. 1

1. Хранение информации обеспечивается обычно записью всех сведений на листе бумаги, при этом к сведениям от-

носят также инструкцию для решения задачи (алгоритм). Заметим, что фактически вычислитель не записывает буквально все на бумаге; кое-что он запоминает (хранит не на листе бумаги, а в памяти), а некоторые сведения черпает из разных справочников и таблиц. Однако эти обстоятельства не должны затушевывать основного факта, заключающегося в том, что вычислительный процесс предполагает наличие таких средств, которые обеспечивают хранение всех необходимых сведений. Таким образом, под *листом бумаги* следует понимать совокупность всех средств, которые обеспечивают хранение сведений.

2. **Обработка информации** предполагает способность вычислителя выполнять отдельные элементарные операции, предусмотренные в алгоритме, и может осуществляться вычислителем посредством специализированных механизмов, например, арифметические операции над числами могут быть реализованы на арифмометре. Каждая отдельная операция заключается в том, что вычислителем в соответствии с инструкцией извлекаются некоторые сведения (например, числа) из определенных граф листа бумаги, они подаются на обрабатывающее устройство (арифмометр), а результаты обработки (промежуточные, например дискриминант квадратного уравнения, и окончательные) помещаются в определенное место листа.

3. **Управление процессом**, т.е. принятие решений о реализации на данном этапе или иной операции и создание условий для ее выполнения, осуществляется самим вычислителем согласно инструкции.

Теперь перейдем непосредственно к описанию нашей вычислительной машины. Из каких устройств составлена машина и как они взаимодействуют между собой?

Ответ на этот вопрос подсказывается тем, что в ней должны происходить те же процессы, какие были только что описаны, но уже без участия человека-вычислителя.

Для изображения информации машина также должна иметь определенный алфавит; однако вместо обычного графического изображения символов, различаемых друг от друга по их начертанию, в машине различные символы алфавита изображаются по-другому. Как, конечно, известно читателям, в совре-

менных ЭВМ применяется двоичное кодирование информации, т.е. представление ее в виде условных единиц и нулей (которые реализуются, например, различными электрическими напряжениями или различными состояниями намагничивания).

Итак, информация, которая поступает в машину, а также та, которая вырабатывается в ней в процессе ее работы, представлена в виде некоторых физических параметров. В интересующих нас случаях эти сведения, составляющие информацию, будут зашифрованы числами. В частности, в виде набора чисел будет зашифрован и сам алгоритм, которым должна “руководствоваться” машина в своей работе. Алгоритмы, составленные специально для машин, называют *программами*. Программа является важнейшей частью информации, с которой имеет дело машина.

В соответствии со схемой *рис. 1*, в машине имеются устройства (органы), выполняющие функции хранения информации, ее обработки, управления процессом (см. *рис. 2*).

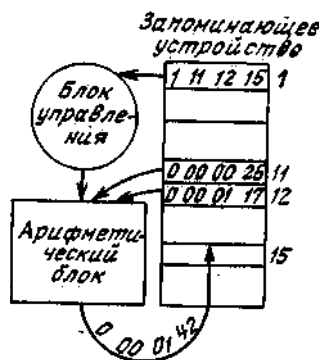


Рис. 2

1. **Запоминающее устройство** играет роль листа бумаги. На условном языке машины в нем фиксируются все нужные сведения, включая программу. Запоминающее устройство (*память машины*) состоит из набора ячеек, занумерованных натуральными числами 1, 2, 3, Эти числа называются *адресами*. Каждая ячейка хранит или может принять на хранение одно закодированное сообщение, представленное в виде числа.

2. **Арифметический блок** играет ту же роль, что и арифмометр, хотя физические принципы, лежащие в основе его работы, сильно отличаются от тех, которые используются в арифмометре. Переработка подаваемых в него данных в нужный результат (например, сложение чисел) происходит путем преобразования в электронном устройстве входных электрических сигналов, изображающих входные данные, в электрические сигналы, изображающие выходные данные. Входные сигналы поступают в арифметиче-

ский блок из памяти, где они хранились, а выходной сигнал направляется в ту ячейку памяти, в которой он будет храниться.

Схематически это изображено на *рис. 2*, где числа из ячеек 11 и 12 складываются, а результат отправляется в ячейку 15. Для того чтобы такая операция была осуществлена в машине в некотором такте, необходимо, чтобы к началу этого такта были произведены соединения ячеек 11 и 12 с арифметическим блоком и арифметического блока с ячейкой 15; также необходимо, чтобы арифметический блок был включен на нужную операцию (в данном случае на сложение). Все это входит уже в “компетенцию” блока управления.

3. **Блок управления** предназначен для выполнения функций, которые в схеме *рис. 1* выполняет вычислитель. Именно на каждом этапе работы машины управляющее устройство создает условия для реализации очередной операции процесса. При этом оно действует как автоматическая телефонная станция, соединяющая тех “абонентов” (узлы и ячейки машины), которые участвуют в данной отдельной операции. Выражаясь образно, блок управления “заглядывает” в программу и в соответствии с ней отправляет распоряжения о срабатывании тех узлов машины, которые должны обеспечить очередную операцию.

Таковы основные три устройства вычислительной машины. Правда, в ней имеется еще ряд важных органов, в частности, для ввода данных в машину и для вывода из нее полученных ею результатов. Однако эти устройства не имеют значения при рассмотрении принципов работы машины и выяснении ее логических и математических возможностей. Поэтому мы не будем привлекать их в дальнейшем к рассмотрению, предполагая, что подача информации в машину и снятие полученной информации производятся непосредственно на запоминающем устройстве.

Машинные команды

Каждая машина характеризуется вполне определенной системой команд (приказов, инструкций), которые она может принимать к исполнению. При этом всякая программа, вкладываемая в машину, представляет собой определенную комбинацию команд и некоторых вспомогательных чисел (параметров), которые помещаются в ячейки памяти. В нашей машине будем применять так называемую “трехадресную систему команд”, каждая из которых является последовательностью четырех чисел: *kabc*, из которых первое указывает номер предписываемой операции, последующие два — адреса двух ячеек, над содержимым которых со-

вершается операция, а последнее — адрес ячейки, в которую следует поместить результат (всего три адреса)¹.

Практически каждая команда записывается в одной ячейке в виде одного числа, цифры которого разбиваются на группы, имеющие соответствующие назначения. На рис. 2 в ячейке 1 запоминающего устройства записано число 1 11 12 15, которое представляет собой зашифрованный приказ: “Сложить (операция № 1) числа из ячеек 11, 12 и результат отправить в ячейку 15”. (Здесь принято разделение чисел на группы справа налево по два разряда, кроме крайней левой группы, в которой указывается одно число — номер команды. Для определенности будем в дальнейшем придерживаться такого метода записи.)

Система команд современных ЭВМ насчитывает сотни, а иногда и тысячи команд. Приведем основные команды нашей машины.

1. Арифметические команды:

- 1) $1abc$ — сложить число из a с числом из b и сумму направить в c ;
- 2) $2abc$ — вычесть из числа, хранящегося в a , число из b и разность отправить в c ;
- 3) $3abc$ — умножить число из a на число из b и произведение отправить в c ;
- 4) $4abc$ — разделить число из a на число из b и частное отправить в c .

2. Команды передачи управления:

- 5) $5 00 00 c$ — переходить к команде, хранящейся в c (безусловная передача управления);

6) $5 01 b c$ — переходить к команде, хранящейся в c , при условии, что в ячейке b хранится положительное число;

7) $5 02 b c$ — переходить к команде, хранящейся в c , при условии, что в ячейке b хранится отрицательное число.

Команды 6–7 называют *условными*; они принимаются к исполнению лишь в том случае, если выполнено соответствующее условие, в противном случае они пропускаются управляющим устройством без выполнения.

3. Команда остановки: 0 00 00 00.

Кроме перечисленных команд, имеются еще и команды так называемых *логических операций* (*логические команды*) и другие, на которых здесь останавливаться не будем. Перечисленных команд вполне достаточно для составления самых разнообразных программ; некоторые примеры будут предложены вам в виде конкурса (см. рубрику “Внимание! Конкурс”). Здесь же в заключение заметим, что работа машины распадается на такты, в течение каждого из которых выполняется одна очередная команда. К началу каждого такта из одной ячейки памяти поступает в блок управления содержащаяся в этой ячейке число-команда. Как только команда поступает туда, управление осуществляет автоматически соответствующие соединения и тем самым обеспечивает выполнение очередной операции процесса. После этого в блок управления поступает следующая команда, и машина срабатывает на следующую операцию — и т.д. до получения требуемого результата, после чего выполняется команда остановки.

“Ломаем” голову

“Тарабарская грамота”

Найдите ключ к “тарабарской грамоте” — тайнописи, применявшейся ранее в России для дипломатической переписки:

Пайцике тсют т “камащамлтой чмароке” — кайпонили, нмирепяшвейля и Моллии цся цинсоракигелтой неменилти

Ключом в данном случае будем называть правила, по которым шифруется исходный текст.

¹ Трехаддресная система команд была принята в выпускавшейся в нашей стране в конце 50-х — начале 60-х годов прошлого века вычислительной машине М-20. В ЭВМ “БЭСМ-6” и в машинах серии “Урал” система команд была одноадресной, в машинах серии “Минск” — двухадресной.



*Ответы,
решения,
разъяснения*

к заданиям,
опубликованным в газете
"В мир информатики"
№ 72 ("Информатика"
№ 7/2006)

1. Статья "Чемпионат по легкой атлетике"

Решение

За первое, второе и третье места в каждом виде соревнований присуждалось различное (но непременно целое и положительное) число очков. Это означает, что за первое место победившая команда не может получить меньше 3 очков. В то же время мы знаем, что соревнования проводились более чем по одному виду спорта (т.к. школа № 1, занявшая первое место по толканию ядра, не стала победителем всего чемпионата) и что школа № 1 (выигравшая соревнования по толканию ядра) набрала 9 очков, поэтому число очков, присуждаемых за первое место, не может быть больше 8.

А может ли оно быть равным 8? Нет, потому что это означало бы, что соревнования проводились только по двум видам спорта и победитель — школа № 3 — не мог бы набрать 22 очка.

Число очков, присуждаемых за первое место, не может быть равным и 7, потому что в этом случае соревнования могли проводиться не более чем по трем видам легкой атлетики, а трех видов недостаточно для того, чтобы школа № 3 набрала 22 очка.

Может ли число очков, присуждаемых за первое место, быть равным 6? Ответ — "нет" — следует из таблицы:

Вид спорта	Толкание ядра	Прыжки в высоту	3	4	Всего
Школа № 3	4	6	6	6	22
Школа № 1	6	1	1	1	9
Школа № 2					9

— школа № 2 в этом случае не может набрать 9 очков.

По той же причине максимальное число очков не может быть равным 4 или 3 (предлагаем читателям убедиться в этом самостоятельно).

Единственно возможным является число 5.

Если команда, занявшая первое место, получает 5 очков, то соревнования должны проводиться по крайней мере по пяти видам спорта (при меньшем числе видов школа № 3 не наберет 22 очка, при большем — школа № 1 наберет более 9 очков). Школа № 2 получает 5 очков за толкание ядра и, следовательно, по 1 очку за участие в соревнованиях по другим видам легкой атлетики. Школа № 3 может набрать 22 очка в

двух случаях: получив 4, 5, 5, 5, 3 очка и получив 2, 5, 5, 5, 5 очков. Первый вариант исключается потому, что тогда школа № 2 набрала бы 17 очков, а мы знаем, что эта команда набрала всего лишь 9 очков:

Вид спорта	Толкание ядра	Прыжки в высоту	3	4	5	Всего
Школа № 3	4	5	5	5	3	22
Школа № 1	5	1	1	1	1	9
Школа № 2	1	4	4	4	4	

Оставшаяся возможность приводит к правильному числу очков у школы № 2, и, таким образом, число очков, полученных каждой командой, восстанавливается однозначно:

Вид спорта	Толкание ядра	Прыжки в высоту	3	4	5	Всего
Школа № 3	2	5	5	5	5	22
Школа № 1	5	1	1	1	1	9
Школа № 2	1	2	2	2	2	9

Правильные ответы прислали:

- Афанасьев Олег, г. Сегежа, Республика Карелия, школа № 5, учитель **Меньшиков В.В.**;
- Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;
- Глижинский Дмитрий, г. Бендеры, Республика Молдова, гимназия № 2, учитель **Глижинская С.Л.**;
- Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;
- Иванова Юлия, Мухарметова Эльвина, Орлов Александр, Сабитова Юлия, Сибгатулина Илюся и Эрштейн Антон, г. Стерлитамак, Республика Башкортостан, школа № 1, учитель **Орлова Е.В.**;
- Морозова Дина, г. Казань, Республика Татарстан, школа № 22, учитель **Осипова А.А.**;
- Урбазаев Батор, село Кижинга, Республика Бурятия, Кижингинский лицей, учитель **Дашиева Д.Р.**

2. Статья "Переложить монеты"

Задача может быть решена несколькими способами. Минимальное число действий в алгоритмах — 4. Полякова Полина (г. Лениногорск, Республика Татарстан, школа № 8, учитель **Кашапова Р.Х.**) привела 18 (!) различных вариантов решения. Вот несколько из них:

- 1. 1 к (3, 6). 2. 6 к (4, 5). 3. 5 к (3, 1). 4. 1 к (5, 6).
- 1. 2 к (5, 6). 2. 4 к (5, 2). 3. 5 к (1, 3). 4. 1 к (5, 4).
- 1. 4 к (1, 2). 2. 5 к (4, 2). 3. 2 к (3, 6). 4. 6 к (2, 5).

Кроме Полины, которую редакция решила наградить дипломом, ответы прислали:

- Гайсина Галия и Ильина Наталья, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Исакова Олеся и Логвиненко Алексей, пос. Лимбяха Новоуренгойского р-на Тюменской обл., школа № 1, учитель **Попова И.С.**;

— Коршунович Елена, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Татаркина Кристина, г. Тобольск, гимназия № 10 им. М.Д. Лицмана, учитель **Махмутова З.К.**;

— Урбазаев Батор, село Кижинга, Республика Бурятия, Кижингинский лицей, учитель **Дашиева Д.Р.**;

— Ягодкин Максим, средняя школа поселка Дзержинский Московской обл., учитель **Зотова М.Н.**

В ряде ответов не учитывались требования условия задачи (сдвигать при перемещении другие монеты нельзя, в новом положении каждая монета должна касаться *двух* других монет, поднимать монеты с поверхности при решении задачи не разрешается).

3. Буквенный ребус

Решение:

$$\begin{array}{r} 1\ 4\ 5\ 8\ 2\ 6 \\ +\ 9\ 4\ 8\ 9\ 4\ 7 \\ \hline 1\ 0\ 9\ 4\ 7\ 7\ 3 \end{array}$$

Правильные ответы прислали:

— Аксаментова Любовь и Парунина Марина, пос. Жатай, Республика Саха (Якутия), школа № 1, учитель **Копылова Л.Ю.**;

— Афанасьев Олег, г. Сегежа, Республика Карелия, школа № 5, учитель **Меньшиков В.В.**;

— Бурцев Анатолий, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Гайсина Галия, Гайсин Рашит и Ильина Наталья, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Идиятов Владимир, г. Ревда Свердловской обл., школа № 28, учитель **Кольцова Е.М.**;

— Кинельский Максим, средняя школа села Пилюгино Бугурусланского р-на Оренбургской обл., учитель **Кинельская С.А.**;

— Клипов Игорь, г. Балашов Саратовской обл., гуманитарно-педагогический лицей-интернат, учитель **Сухорукова Е.В.**;

— Кононов Владимир, средняя школа поселка Ивот Дятковского р-на Брянской обл., учитель **Брин Д.С.**;

— Логвиненко Алексей, пос. Лимбяха Новоуренгойского р-на Тюменской обл., школа № 1, учитель **Попова И.С.**;

— Морозова Дина, г. Казань, Республика Татарстан, школа № 22, учитель **Осипова А.А.**;

— Полякова Полина, г. Лениногорск, Республика Татарстан, школа № 8, учитель **Кашапова Р.Х.**;

— Татаркина Кристина, г. Тобольск, гимназия № 10 им. М.Д. Лицмана, учитель **Махмутова З.К.**;

— Урбазаев Батор, село Кижинга, Республика Бурятия, Кижингинский лицей, учитель **Дашиева Д.Р.**

Правильный ответ на загадку, приведенную в качестве эпиграфа к статье “Старинные русские меры длины”, прислали:

— Афанасьев Олег, г. Сегежа, Республика Карелия, школа № 5, учитель **Меньшиков В.В.**;

— Гайсина Галия и Ильина Наталья, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Глижинский Дмитрий, г. Бендеры, Республика Молдова, гимназия № 2, учитель **Глижинская С.Л.**;

— Ягодкин Максим, средняя школа поселка Дзержинский Московской обл., учитель **Зотова М.Н.**

*Ответы,
решения,
разъяснения*

**к заданиям,
опубликованным в газете
“В мир информатики”
№ 73 (“Информатика”
№ 8/2006)**

1. Статья “0 признаках делимости”

1) число $(a_n a_{n-1} \dots a_1 a_0)_2$;
— делится на 2, если его последняя цифра a_0 равна нулю;

— делится на 4, если обе его последние цифры a_1 и a_0 равны нулю;

2) число $(a_n a_{n-1} \dots a_1 a_0)_3$;
— делится на 2, если сумма его цифр есть четное число;

— делится на 3, если его последняя цифра a_0 равна нулю;

3) число $(a_n a_{n-1} \dots a_1 a_0)_4$;

— делится на 2, если на 2 делится его последняя цифра a_0 ;

— делится на 3, если сумма его цифр кратна трем;

— делится на 4, если его последняя цифра a_0 равна нулю;

4) число $(a_n a_{n-1} \dots a_1 a_0)_6$;

— делится на 2, если на 2 делится его последняя цифра a_0 ;

— делится на 3, если на 3 делится его последняя цифра a_0 ;

— делится на 5, если сумма его цифр кратна пяти;

— делится на 6, если его последняя цифра a_0 равна нулю;

5) число $(a_n a_{n-1} \dots a_1 a_0)_8$;

— делится на 2, если на 2 делится его последняя цифра a_0 ;

— делится на 4, если на 3 делится его последняя цифра a_0 ;

— делится на 7, если сумма его цифр кратна семи;
 — делится на 8, если его последняя цифра a_0 равна нулю;

б) число $(a_n a_{n-1} \dots a_1 a_0)_{12}$:
 — делится на 8, если на 8 делится число $(a_1 a_0)_{12}$, образованное его двумя последними цифрами;
 — делится на 9, если на 9 делится число $(a_1 a_0)_{12}$, образованное его двумя последними цифрами;
 — делится на 11, если на 11 делится сумма его цифр, т.е. $a_n + a_{n-1} + a_{n-2} + \dots + a_1 + a_0$.

2. Буквенный ребус “Как получается молоко”

Обратим внимание на разряд десятков. В нем результат может быть равен К в следующих случаях:

- 1) A = 1 и B = 0;
- 2) A = 1 и B = 5;
- 3) A = 2 и B = 0;
- 4) A = 2 и B = 5;
- 5) A = 3 и B = 0;
- 6) A = 3 и B = 5;
- 7) A = 7 и B = 4;
- 8) A = 7 и B = 9;
- 9) A = 8 и B = 4;
- 10) A = 8 и B = 9;
- 11) A = 9 и B = 4.

Проанализировав каждый из перечисленных вариантов с учетом того, что $M = K + D + 1$ или $M = K + D + 2$, можно убедиться, что возможными являются два варианта решения ребуса:

$$\begin{array}{r} 140498 \quad \text{или} \quad 540498 \\ + 60898 \quad \quad \quad + 60898 \\ \hline 542018 \quad \quad \quad 142058 \\ \hline 743414 \quad \quad \quad 743454 \end{array}$$

Поскольку по условию число КОРОВА больше, чем число ДОЯРКА, первый вариант не подходит.

Правильные ответы прислали:

— Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Диденко Ольга, Занозин Михаил и Пахомов Сергей, г. Нижний Новгород, школа № 77, учитель **Занозина Г.В.** (эти читатели представили развернутое обоснование решения);

— Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Морозова Дина, г. Казань, Республика Татарстан, школа № 22, учитель **Осипова А.А.**

3. Статья “Четыре карточки”

Ложность высказывания “Если на одной стороне карточки — гласная буква, то на другой — четное число” можно проверить, перевернув карточку с буквой **А** — если на другой стороне записано нечетное число, то высказывание будет ложным. Если же там

будет записано четное число, то означает ли это, что высказывание будет истинным? Нет, ведь истинность указанного высказывания должна относиться ко всем карточкам. Для проверки истинности следует перевернуть также карточку с цифрой **5** — если на другой стороне записана гласная, то высказывание будет ложным (на карточке — гласная буква и нечетное число). Если же там записана согласная буква, то это не будет говорить о ложности рассматриваемого высказывания, и по тому факту, что на другой стороне карточки с буквой **А** записано четное число, высказывание будет истинным. Итак, следует перевернуть карточки с буквой **А** и с цифрой **5** (“содержимое” остальных карточек на истинность или ложность высказывания не влияет).

Ответы прислали:

— Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Бурцев Анатолий, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Гайсина Галия и Гайсин Рашид, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Идиятов Владимир, г. Ревда Свердловской обл., школа № 28, учитель **Кольцова Е.М.**;

— Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина Л.И.**;

— Лобода Андрей, г. Старый Оскол Белгородской обл., школа № 24, учитель **Винникова О.Е.**;

— Лось Олеся, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**

4. Статья “Числа в две строки”

Ответы:

- 1) нельзя;
- 2) можно:

1	11	3	9	8	7
12	2	10	4	5	6

Правильные ответы прислали:

— Александров Алексей, г. Новочеркасск, Суворовское училище МВД РФ, преподаватель **Воронкова О.Б.**;

— Бурцев Анатолий, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Васильева Екатерина, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Гайсина Галия и Гайсин Рашид, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Диденко Ольга, Занозин Михаил и Пахомов Сергей, г. Нижний Новгород, школа № 77, учитель **Занозина Г.В.**;



— Иванова Юлия, Мухарметова Эльвина, Орлов Александр, Сибгатуллина Илюся и Эрштейн Антон, г. Стерлитамак, Республика Башкортостан, школа № 1, учитель **Орлова Е.В.**;

— Идиятов Владимир, г. Ревда Свердловской обл., школа № 28, учитель **Кольцова Е.М.**;

— Калюга Антон, пос. Лимбьяха Новоуренгойского р-на Тюменской обл., школа № 1, учитель **Попова Н.В.**;

— Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина А.И.**;

— Лобода Андрей, г. Старый Оскол Белгородской обл., школа № 24, учитель **Винникова О.Е.**;

— Моляров Александр, Новгородов Артур, Пинигина Вероника и Щербинина Ольга, пос. Жатай, Республика Саха (Якутия), школа № 1, учитель **Копылова А.Ю.**;

— Морозова Дина, г. Казань, Республика Татарстан, школа № 22, учитель **Осипова А.А.**;

— Силаев Валерий, Ардатовский аграрный техникум, пос. Ардатов Нижегородской обл., преподаватель **Касаткина С.Ю.**

Анатолий Бурцев предложил аналогичную задачу с тремя строчками. Так, он указал, что первые 9 натуральных чисел можно расставить в три строки так, что суммы чисел в каждой строке равны между собой и суммы чисел в каждом столбце также равны между собой:

9	4	2
1	8	6
5	3	7

А можно ли решить такую же задачу применительно:

- 1) к первым 15 натуральным числам;
- 2) к первым 18 натуральным числам?

Ответы, пожалуйста, присылайте в редакцию.

Ответы, решения, разъяснения

к заданиям,
опубликованным в газете
"В мир информатики"
№ 74 ("Информатика"
№ 9/2006)

1. Статья "Рональд, Жак и Джордж"

На рисунке изображены (слева направо): Джордж, Жак, Рональд.

Правильные ответы прислали:

— Бурцев Анатолий, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Бынина Ольга, Ерилина Алена, Ключникова Ирина, Светлова Анастасия и Чикунова Мария, г. Новохоперск Воронежской обл., гимназия № 1, учитель **Матькин В.Ю.**;

— Гайсина Галия, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Глухов Валентин и Полушкин Евгений, Кемлянская средняя школа Ичалковского р-на Республики Мордовия, учитель **Силантьев О.П.**;

— Потлова Татьяна, средняя школа села Речица Ливенского р-на Орловской обл., учитель **Потлова О.А.**

2. Статья "Непростой кубик"

Решение

Рассмотрим грань, на которой записано число 35. Если на противоположной ей грани стоит нечетное число, то на двух других невидимых гранях стоят разные четные числа. Но существует только единственное простое четное число — 2, следовательно, "напротив" грани с числом 35 и записано это число. Тогда сумма

чисел на противоположных гранях равна 37, и на гранях, противоположных числам 14 и 18, стоят числа 23 и 19.

Правильный ответ прислали:

— Бурцев Анатолий, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Гайсина Галия, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Глухов Валентин и Полушкин Евгений, Кемлянская средняя школа Ичалковского р-на Республики Мордовия, учитель **Силантьев О.П.**;

— Дмитриев Станислав, г. Стерлитамак, Республика Башкортостан, школа № 12, учитель **Дмитриева О.В.**;

— Кирилина София и Кирмасова Юлия, г. Холмск Сахалинской обл., лицей "Надежда", учитель **Романькова С.Н.**;

— Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина А.И.**;

— Кучерова Ирина, Соломеин Антон и Чапаров Зарип, пос. Жатай, Республика Саха (Якутия), школа № 1, учитель **Копылова А.Ю.**;

— Морозова Дина, г. Казань, Республика Татарстан, школа № 22, учитель **Осипова А.А.**;

— Потлова Татьяна, средняя школа села Речица Ливенского р-на Орловской обл., учитель **Потлова О.А.**;

— Серегина Мария и Хамзина Галия, г. Уфа, Республика Башкортостан, гимназия № 3 (фамилия учителя информатики в письме, к сожалению, не указана).

Стародубец Андрей, г. Холмск Сахалинской обл., лицей "Надежда", учитель **Романькова С.Н.**, представил программу на языке Паскаль, с помощью которой можно определить искомые числа.

Цыдыпов Лубсан, село Кижинга, Республика Бурятия, Кижингинский лицей, учитель **Дашиева Д.Р.**, прислал правильное решение буквенного ребуса, опубликованного в газете-вкладке "В мир информатики" № 67 ("Информатика" № 2/2006), а также программу на языке Паскаль, с помощью которой можно получить ответ на вопрос, заданный в статье "Городские слухи" (ответ опубликован в предыдущем номере нашей газеты).

Потлова Татьяна, средняя школа села Речица Ливенского р-на Орловской обл., учитель **Потло-**

ва О.А., представила ответы на задания конкурсов № 44–46.

Спасибо всем приславшим ответы!

* * *

Может ли быть такое?

Один мальчик сказал: "Позавчера мне было 10 лет, а в будущем году мне исполнится 13 лет". Может ли быть такое?

Показываем текущее время в Microsoft Excel

Microsoft Excel углубленно

Если вы хотите показать на листе электронной таблицы Microsoft Excel текущее время:

	E	F	G
Сейчас			
	20	22	1

Рис. 1

— то введите в ячейку E3 формулу: =ЧАС(ТДАТА()), в ячейку F3 формулу: =МИНУТЫ(ТДАТА()), в ячейку G3 формулу: =СЕКУНДЫ(ТДАТА()). Используемая в приведенных формулах функция ТДАТА() возвращает полное значение текущей даты и времени, а другие функции выделяют из него количество полных часов, полных минут, прошедших с начала очередного часа, и полных секунд, прошедших с начала очередной минуты.

Задание для самостоятельной работы

Получите на листе текущее время в виде:

	E	F	G
Сейчас			
	20 час	22 мин	1 сек

Рис. 2

Указание по выполнению. Для "склейки" значений времени и текста используйте функцию СЦЕПИТЬ или оператор "&".

Нетрудно убедиться, что приведенные формулы показывают текущее время только один раз — после ввода формулы в ячейку. Для "обновления" времени следует нажать функциональную клавишу **F9**. Можно также разместить на листе кнопку, по щелчку на которой будет показываться текущее время:

	E	F	G
Сейчас			
	20	37	30
	Показать время		

Рис. 3

Делается это так.

1. Выберите пункт меню **Вид**, в нем — подпункт **Панели инструментов** и в появившемся перечне щелкните на строке **Формы**.

2. На появившейся панели **Формы** выберите элемент **Кнопка** и мышью, методом буксировки, "нарисуйте" кнопку в нужном месте листа.

3. В появившемся после рисования кнопки диалоговом окне **Назначить макрос объекту** в поле **Имя макроса** дайте имя, например, **Время** (можно оставить и имя, предложенное программой) и щелкните на кнопке **Создать**.

4. На листе модуля (программы) между заголовком макроса (процедуры обработки события — щелчка на кнопке) с именем **Время ()** и его завершающей конструкцией **End Sub** запишите:

```
Calculate
```

Эта команда проводит перерасчет всех формул на листе, в результате чего будет считано новое значение текущего времени и в ячейках E3:G3 выведутся новые значения.

Чтобы выполнить макрос и увидеть текущее время, необходимо, как обычно, щелкнуть на кнопке.

А можно ли сделать так, чтобы текущее время показывалось постоянно, а не только при нажатии клавиши **F9** или щелчке на кнопке? Можно! Для этого в теле процедуры **Время ()** вместо команды **Calculate** запишите:

```
Application.OnTime Now +  
TimeValue("00:00:01"), "ОбновитьЗначения"
```

Метод OnTime объекта Application (приложения Microsoft Excel) позволяет назначить выполнение процедуры на заданное время. В приведенном случае через 1 секунду (Now + TimeValue("00:00:01")) вызывается процедура, имя которой ОбновитьЗначения(). Такую процедуру нужно создать, записав здесь же, на листе модуля (программы):

```
Sub ОбновитьЗначения()
Calculate
Время
End Sub
```

Процедура Sub ОбновитьЗначения():

1) обеспечивает вывод новых значений времени (по команде Calculate);

2) вызывает процедуру Время, которая затем опять назначит вызов процедуры ОбновитьЗначения через 1 секунду для очередного обновления значений, и т.д.²

Задание для самостоятельной работы

Получите на листе постоянно обновляемое значение текущего времени в виде, приведенном на рис. 2.

Литература

1. Макросы / "В мир информатики" № 48 ("Информатика" № 1/2005).
2. Медведев А.Н. Как создать мультфильм с помощью макроса. / "В мир информатики" № 49 ("Информатика" № 2/2005).

Внимание! Конкурс

Итоги конкурса № 47 для учащихся

Напомним, что необходимо было разработать устройство, которое обеспечивает вывод на экран монитора шестизначного десятичного числа по его известным цифрам, состоящее из девяти перечисленных в условии конкурса элементов.

Ответ

Прежде чем описывать оптимальное устройство, которое решает поставленную в условии конкурса задачу, проанализируем так называемую "развернутую" форму записи шестизначного десятичного числа, состоящего из цифр c_1, c_2, c_3, c_4, c_5 и c_6 :

$$c_1 \cdot 10^5 + c_2 \cdot 10^4 + c_3 \cdot 10^3 + c_4 \cdot 10^2 + c_5 \cdot 10 + c_6.$$

Вынесем множитель 10 за скобки — получим:

$$(c_1 \cdot 10^4 + c_2 \cdot 10^3 + c_3 \cdot 10^2 + c_4 \cdot 10 + c_5) \cdot 10 + c_6.$$

Повторим процедуру еще несколько раз:

$$\begin{aligned} & ((c_1 \cdot 10^3 + c_2 \cdot 10^2 + c_3 \cdot 10 + c_4) \cdot 10 + c_5) \cdot 10 + c_6 = \\ & = (((c_1 \cdot 10^2 + c_2 \cdot 10 + c_3) \cdot 10 + c_4) \cdot 10 + c_5) \cdot 10 + c_6 = \\ & = (((c_1 \cdot 10 + c_2) \cdot 10 + c_3) \cdot 10 + c_4) \cdot 10 + c_5) \cdot 10 + c_6. \end{aligned}$$

Проанализировав полученную формулу, можно увидеть, что она описывает последовательное умножение предыдущего результата на 10 и прибавление очередной цифры c_i . Как известно, такая схема вычислений называется "схема Горнера".

В устройстве, которое обеспечивает решение задачи с использованием вычислений по схеме Горнера, достаточно применить только две ячейки памяти — в одной из них можно хранить очередную цифру, в другой — результат предыдущих вычислений.

Обозначим используемые элементы следующим образом:

- 1 и 2 — ячейки памяти для хранения неотрицательных целых чисел (до шестизначных включительно);
- 3 — устройство для считывания цифры, набранной на клавиатуре;

- 4 — устройство для считывания чисел из ячеек памяти;
- 5 — устройство для записи неотрицательных целых чисел в любую из ячеек памяти;
- 6 — устройство для умножения двух неотрицательных целых чисел;
- 7 — устройство для сложения двух неотрицательных целых чисел;
- 8 — устройство для вывода числа из любой ячейки памяти на экран монитора (при выводе имевшаяся на экране информация удаляется).

Алгоритм работы такого устройства:

- 1) нажатие первой цифры c_1 на клавиатуре;
- 2) считывание цифры устройством 3;
- 3) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 4) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 5) запись полученного произведения в ячейку памяти 2 с помощью устройства 5;
- 6) нажатие второй цифры c_2 на клавиатуре;
- 7) считывание цифры устройством 3;
- 8) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 9) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 10) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 11) умножение числа в ячейке 2 на 10 с помощью устройства 6;
- 12) запись полученного произведения в ячейку памяти 2 с помощью устройства 5;
- 13) нажатие третьей цифры c_3 на клавиатуре;

² Описанная ситуация, когда процедура вызывает саму себя через вызов другой процедуры, называется косвенной, или непрямой, рекурсией. Подробно о рекурсии см. газету-вкладку "В мир информатики" № 5-6 ("Информатика" № 37-38/2003).

- 14) считывание цифры устройством 3;
- 15) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 16) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 17) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 18) умножение числа в ячейке 2 на 10 с помощью устройства 6;
- 19) запись полученного произведения в ячейку памяти 2 с помощью устройства 5;
- 20) нажатие четвертой цифры c_4 на клавиатуре;
- ...
- 26) запись полученного произведения в ячейку памяти 2 с помощью устройства 5;
- 27) нажатие пятой цифры c_5 на клавиатуре;
- ...
- 33) запись полученного произведения в ячейку памяти 2 с помощью устройства 5;
- 34) нажатие шестой цифры c_6 на клавиатуре;
- 35) считывание цифры устройством 3;
- 36) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 37) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 38) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 39) считывание числа в ячейке памяти 2 с помощью устройства 4;
- 40) вывод считанного числа на экран с помощью устройства 6.

Возможен также вариант устройства, использующего только две ячейки памяти, но реализующий расчет по развернутой форме записи числа (см. выше). В нем в ячейке памяти 1 будет храниться значение (промежуточное или окончательное) в том или ином разряде (с учетом весомости), в ячейке памяти 2 — сумма уже рассчитанных значений в разрядах.

Алгоритм его работы:

- 1) нажатие первой цифры c_1 на клавиатуре;
- 2) считывание цифры устройством 3;
- 3) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 4) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 5) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 6) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 7) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- ...
- 12) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 13) запись полученного произведения в ячейку памяти 2 с помощью устройства 5;
- 14) нажатие второй цифры c_2 на клавиатуре;
- 15) считывание цифры устройством 3;
- 16) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 17) умножение числа в ячейке 1 на 10 с помощью устройства 6;

- 18) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 19) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 20) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- ...
- 23) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 24) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 25) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 26) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 27) нажатие третьей цифры c_3 на клавиатуре;
- 28) считывание цифры устройством 3;
- 29) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 30) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 31) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- ...
- 34) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 35) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 36) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 37) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 38) нажатие четвертой цифры c_4 на клавиатуре;
- 39) считывание цифры устройством 3;
- 40) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 41) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 42) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 43) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 44) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 45) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 46) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 47) нажатие пятой цифры c_5 на клавиатуре;
- 48) считывание цифры устройством 3;
- 49) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
- 50) умножение числа в ячейке 1 на 10 с помощью устройства 6;
- 51) запись полученного произведения в ячейку памяти 1 с помощью устройства 5;
- 52) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
- 53) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
- 54) нажатие шестой цифры c_6 на клавиатуре;



- 55) считывание цифры устройством 3;
 56) запись считанной цифры в ячейку памяти 1 с помощью устройства 5;
 57) сложение чисел в ячейках памяти 1 и 2 с помощью устройства 7;
 58) запись полученной суммы в ячейку памяти 2 с помощью устройства 5;
 59) считывание числа в ячейке памяти 2 с помощью устройства 4;
 60) вывод считанного числа на экран с помощью устройства 6.
- Преимущества первого варианта устройства (точнее — первого алгоритма) очевидны.

Ответы, в которых предложено устройство, аналогичное описанному, прислали Лобода Андрей, г. Старый Оскол Белгородской обл., школа № 24, учитель **Винникова О.Е.**, и Лось Олеся, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.** Андрей и Олеся будут награждены дипломами. Поздравляем! Кроме них, ответы прислали:

- Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;
- Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;
- Юргин Андрей, основная школа села Кубайка Красноярского края, учитель **Семеновский В.А.**

Итоги конкурса № 48 для учащихся

Напомним, что необходимо было предложить способ, по которому можно прочитать и редактировать в текстовом редакторе Microsoft Word на компьютере типа IBM документ, созданный на компьютере "Macintosh".

Ответ: нужно распечатать документ на компьютере "Macintosh", а затем отсканировать его на компьютере типа IBM, отсканированный текст распознать с помощью программ распознавания текста (ABBY FineReader или др.) и передать его для редактирования в текстовый редактор Microsoft Word.

Котов Александр, г. Ревда Свердловской обл., школа № 28, учитель **Кольцова Е.М.**, предложил использовать

для решения проблемы конвертеры текстовых файлов (входящие в состав пакета Microsoft Office или специализированные). Александр будет награжден дипломом. Поздравляем! Кроме него, ответы прислали:

- Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;
- Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина Л.И.**;
- Невицкий Сергей, Москва, гимназия № 1530, учитель **Шамшев М.В.**

Обращаем внимание на то, что в условии конкурса речь шла о несовместимости форматов подготовленного документа и формата документа Microsoft Word.

Конкурс № 50 для учащихся

Для ЭВМ, описанной в статье "Вычислительная машина с автоматическим управлением", разработать программы решения трех задач:

- 1) нахождения корней линейного уравнения $ax + b = 0$;
- 2) расчета значения $\frac{x^2 + y^3}{x^3 + y^2}$ при заданных значениях x и y ;
- 3) нахождения корней системы уравнений:

$$\begin{aligned} ax + by &= c; \\ dx + ey &= f. \end{aligned}$$

Указания по выполнению. Принять для определенности, что:

- 1) в задаче 1 коэффициенты a и b размещены соответственно в ячейках 21 и 22, а результат будет записываться в ячейку номер 101;
- 2) в задаче 2 все исходные значения записаны в ячейках 40–41, а для промежуточных и окончательного результатов отводятся ячейки, начиная с номера 51;
- 3) в задаче 3 коэффициенты a, b, c, d, e и f помещены подряд в ячейки памяти, начиная с номера 51:

Адрес	Содержимое
51	a
52	b
53	c
54	d
55	e
56	f

чательных результатов отводятся ячейки, начиная с номера 31.

Команды программы во всех случаях записывать последовательно в ячейки с номерами 1, 2, ...

Ответы отправьте в редакцию до 15 октября по адресу: 121165, Москва, ул. Киевская, д. 24, "Первое сентября", "Информатика", или по электронной почте: inf@1september.ru. Пожалуйста, четко укажите в ответе свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики.

За активное участие в конкурсах для учащихся Похвальной грамотой газеты "Информатика" награждены:

- Александров Алексей, г. Новочеркасск, Суворовское училище МВД РФ, преподаватель **Воронкова О.Б.**;
- Глухов Валентин и Полушкин Евгений, Кемлянская средняя школа Ичалковского р-на Республики Мордовия, учитель **Силантьев О.П.**;
- Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина Л.И.**;
- Морозова Дина, г. Казань, Республика Татарстан, школа № 22, учитель **Осипова А.А.**;
- Полякова Полина, г. Лениногорск, Республика Татарстан, школа № 8, учитель **Кашапова Р.Х.**;
- Потлова Татьяна, средняя школа села Речица Ливенского р-на Орловской обл., учитель **Потлова О.А.**;
- Сабитова Юлия и Эрштейн Антон, г. Стерлитамак, Республика Башкортостан, школа № 1, учитель **Орлова Е.В.**;
- Урбазаев Батор и Цыдыпов Лубсан, село Кижинга, Республика Бурятия, Кижингинский лицей, учитель **Дашиева Д.Р.**

“КАК ЭТО ДЕЛАЮ Я”

Методический конкурс для учителей информатики

Дорогие коллеги!

В новом учебном году мы продолжаем творческий конкурс для учителей информатики “Как это делаю я”. Для наших новых подписчиков кратко напомним его суть.

Для каждого тура предлагается типичная задача (проблема), с которой приходится иметь дело большинству учителей информатики. Фиксируется формат и сроки представления материалов. Все работы по мере поступления публикуются в специальном разделе на сайте “Информатики” <http://inf.1september.ru>. По завершении каждого тура избранные работы публикуются на страницах газеты. В конце учебного года все участники (даже если они принимали участие всего в одном туре) получают диплом и компакт-диск со всеми работами (разумеется, и то и другое бесплатно). В прошлом учебном году было проведено шесть туров. Напомним их темы:

- “Введение в предмет”;
- “Соблюдаем технику безопасности”;
- “Задачи, которые по силам лишь компьютеру”;
- “Домашние задания”;
- “Кабинет — наш второй дом”;
- “Внеклассная работа по информатике в школе”.

Отметим, что, хотя формально “Как это делаю я” является конкурсом (при всем желании мы не можем опубликовать на страницах газеты все работы и вынуждены субъективно отбирать наиболее интересные), соревновательная составляющая в нем вторична. Мы рассматриваем конкурс “Как это делаю я” как открытую площадку для обмена опытом решения актуальных и насущных задач. Надеемся, что материалы, опубликованные в прошлом году, были интересными и полезными.

ЗАДАНИЕ СЕДЬМОГО ТУРА “ИНФОРМАТИКА В ПРОФИЛЬ”

Эксперимент по переходу на профильное обучение шагает по стране. До большинства из нас он уже так или иначе дошагал, кому-то все еще только предстоит. И тем и другим интересен опыт коллег.

Пожалуйста, расскажите о ходе эксперимента по переходу на профильное обучение в вашей школе. Мы знаем, что наработано уже немало — разработаны программы, учебные планы, имеется опыт преподавания и разработки элективных курсов. Поделитесь с коллегами!

Нас очень интересуют вопросы профильного преподавания информатики (и здесь в течение нескольких десятилетий накоплен большой опыт), но особенно — все, что связано с преподаванием нашего предмета в непрофильных классах. Ведь многие справедливо считают, что преподавать информатику, например, гуманитариям сложнее, чем математикам. С содержательной точки зрения здесь нужны особые программы, с методической — специальные приемы обеспечения мотивации.

ФОРМАТ ПРЕДСТАВЛЕНИЯ КОНКУРСНЫХ МАТЕРИАЛОВ

Материалы принимаются только в электронном виде в формате Microsoft Word. Объем основного текста (с учетом пробелов) — не более 6000 знаков. Дополнительные материалы (развернутые программы курсов, учебные планы и т.п.) можно оформлять в виде приложений. Объем приложений не ограничен.

Материалы можно присылать на электронных носителях или по электронной почте.

Почтовый адрес: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика”.
Электронный адрес: inf@1september.ru.

В теме (subject) электронных писем, пожалуйста, указывайте “Методический конкурс, седьмой тур”.

Вместе с материалами, пожалуйста, присылайте краткую информацию о себе, в обязательном порядке включающую: фамилию, имя, отчество, полное название учебного заведения, в котором вы преподаете, стаж работы. Пожалуйста, указывайте также максимально полную контактную информацию: полный почтовый адрес (с индексом), электронный адрес, телефоны (с кодом населенного пункта). (Эта информация не будет опубликована, но она может потребоваться редакции для оперативной связи.)

КЛЮЧЕВЫЕ СРОКИ

Срок отправки материалов седьмого тура — до 30 октября 2006 г. (для “бумажных” писем дата отправки фиксируется по штемпелю предприятия-отправителя).

Результаты седьмого тура будут опубликованы в № 2/2007. На сайте “Информатики” материалы будут размещаться по мере поступления.

Задание следующего тура будет опубликовано в № 19.

**ИЗДАТЕЛЬСКИЙ ДОМ
“ПЕРВОЕ СЕНТЯБРЯ”**
главный редактор —
А.С. Соловейчик

ГАЗЕТЫ
ИЗДАТЕЛЬСКОГО ДОМА
Первое сентября

гл. ред. — Е.В. Бирюкова,
индекс подписки — 32024;

Английский язык

гл. ред. — Е.В. Громушкина,

индекс подписки — 32025;

Библиотека в школе

гл. ред. — О.К. Громова,

индекс подписки — 33376;

Биология

гл. ред. — Н.Г. Иванова,

индекс подписки — 32026;

География

гл. ред. — О.Н. Коротова,

индекс подписки — 32027;

Дошкольное образование

гл. ред. — М.С. Аромштам,

индекс подписки — 33373;

Здоровье детей

гл. ред. — Н.В. Сёмина,

индекс подписки — 32033;

Информатика

гл. ред. — С.Л. Островский,

индекс подписки — 32291;

Искусство

гл. ред. — М.Н. Сартан,

индекс подписки — 32584;

История

гл. ред. — А.Л. Савельев,

индекс подписки — 32028;

Литература

отв. сек. — С.Ф. Дмитренко,

индекс подписки — 32029;

Математика

и. о. гл. ред. — Л.О. Рослова,

индекс подписки — 32030;

Начальная школа

гл. ред. — М.В. Соловейчик,

индекс подписки — 32031;

Немецкий язык

гл. ред. — М.Д. Бузоева,

индекс подписки — 32292;

Русский язык

гл. ред. — Л.А. Гончар,

индекс подписки — 32383;

Спорт в школе

гл. ред. — О.М. Леонтьева,

индекс подписки — 32384;

Управление школой

гл. ред. — Я.А. Сартан,

индекс подписки — 32652;

Физика

гл. ред. — Н.Д. Козлова,

индекс подписки — 32032;

Французский язык

гл. ред. — Г.А. Чесновицкая,

индекс подписки — 33371;

Химия

гл. ред. — О.Г. Блохина,

индекс подписки — 32034;

Школьный психолог

гл. ред. — И.В. Вачков,

индекс подписки — 32898.

Гл. редактор
С.Л. Островский
Зам. гл. редактора
А.И. Сенокосов
Редакция
Е.В. Андреева
Д.М. Златопольский (редактор
вкладки “В мир информатики”)
Л.Н. Картвелишвили
С.Б. Кишкина
Н.П. Медведева
Ю.А. Первин (редактор вкладки
“Началка”)
Корректор Дизайн и верстка
Е.Л. Володина Н.И. Пронская

©ИНФОРМАТИКА 2006
Выходит два раза в месяц
При перепечатке ссылка
на ИНФОРМАТИКУ обязательна,
рукописи не возвращаются

Адрес редакции и издателя:

Киевская, 24, Москва, 121165
тел. 8-499-249-48-96
Отдел рекламы: 8-495-249-98-70

ИНДЕКС ПОДПИСКИ

для индивидуальных подписчиков 32291
комплекта изданий 32744

Тел.: 8-495-249-31-38, 249-33-86. Факс 8-495-249-31-84

Учредитель: ООО “Чистые пруды”

Зарегистрировано в Министерстве РФ по делам
печати. ПИ № 77-7230 от 12.04.2001.
Отпечатано в ОИД “Медиа-Пресса”,
ул. Правды, 24, Москва, ГСП-3, А-40, 125993
Тираж 6500 экз.
Срок подписания в печать по графику 24.08.2006.
Номер подписан 24.08.2006.
Заказ № 615518
Цена свободная

Internet: inf@1september.ru
WWW: <http://www.1september.ru>