

С 1 по 30 апреля 2002 года в Московском городском доме учителя состоится

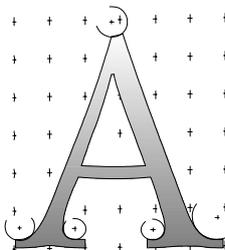
# Московский педагогический марафон учебных предметов



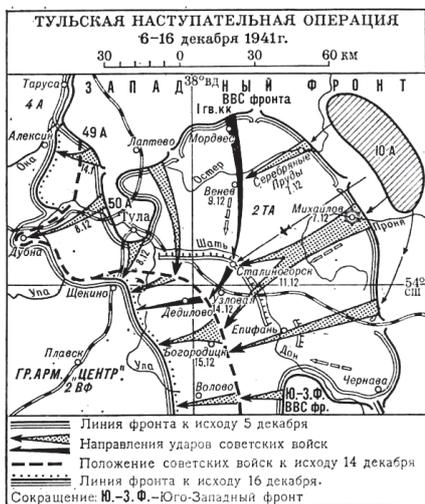
• 1 апреля – День учителя русского языка • 2 апреля – День учителя литературы • 3 апреля – День учителей мировой художественной культуры, музыки и изо • 4 апреля – День учителя истории • 5 апреля – День школьного библиотекаря • 8 апреля – День учителя географии • 9 апреля – День учителя биологии • 10 апреля – День учителя химии • 11 апреля – День учителя физики • 12 апреля – День учителя математики • 15 апреля – **ДЕНЬ УЧИТЕЛЯ ИНФОРМАТИКИ** • 16 апреля – День учителя английского языка • 17 апреля – День учителя немецкого языка • 18 апреля – День учителя французского языка • 19 апреля – День учителей технологии, профориентации и ОБЖ • 22 апреля – День учителя физкультуры • 23 апреля – День здоровья детей • 24 апреля – День дошкольного образования • 25 апреля – День учителя начальной школы • 26 апреля – День логопедов и коррекционных педагогов • 29 апреля – День школьного психолога • 30 апреля – День школьной администрации •

№ 8 (345) 23–28 февраля 2002

# ИНФОРМАТИК



Еженедельная газета Издательского дома «ПЕРВОЕ СЕНТЯБРЯ»  
ПОДПИСКА: (095) 249-47-58



## Искусство давать плохие ответы

60 лет назад, в начале 1940-х годов, исследование операций оформилось как самостоятельное научное направление, а в начале 1950-х годов появился метод динамического программирования, ставший одним из эффективных методов исследования операций

*«Исследование операций представляет собой искусство давать плохие ответы на те практические вопросы, на которые даются еще худшие ответы другими способами».*

Томас Саати, один из создателей исследования операций

Первые публикации, посвященные этой научной дисциплине, появились в 1939–40 гг. В них методы исследования операций применялись для решения военных задач, в частности, для анализа и исследования боевых операций. Отсюда и возникло название. Позднее принципы и приемы исследования операций стали использоваться в области промышленно-финансового управления.

Большой вклад в формирование и развитие исследования операций внесли зарубежные и отечественные ученые: Р.А. Коф, Р.Беллман, Г.Данциг, Г.Кун, Т.Саати, Р.Черчмен (США), А.Кофман, Р.Фор (Франция), Н.П. Бусленко, Ю.Б. Гермейер, Б.В. Гнеденко, Л.В. Канторович, Д.Б. Юдин и другие.

Для задач исследования операций часто выделяют следующие основные классы.

**Задачи управления запасами.** Самый распространенный класс задач. С увеличением запасов увеличиваются расходы на их хранение, но уменьшаются потери из-за возможной их нехватки. Одна из задач управления запасами заключается в определении такого уровня запасов, который минимизирует сумму затрат, связанных с хранением запасов, и потерь из-за их дефицита. (Кстати, динамическое программирование, возникшее и сформировавшееся благодаря работам Ричарда Беллмана и его сотрудников, обязано своим появлением динамическим задачам управления запасами.)

**Задачи распределения ресурсов.** Возникают, когда существует определенный набор работ (операций), которые необходимо выполнить, а ресурсов для выполнения каждой работы наилучшим образом не хватает. К данному классу относится, например, так называемая «транспортная задача» (совсем недавно рассмотренная в нашей газете, см. № 5).

**Задачи ремонта и замены оборудования.** Появляются в тех случаях, когда работающее оборудование изнашивается, устаревает и со временем подлежит замене.

**Задачи массового обслуживания.** Здесь рассматриваются вопросы образования и функционирования очередей (напри-

мер, очередей самолетов, идущих на посадку, клиентов в банке, абонентов телефонной станции), возникающих из-за того, что поток требований или клиентов неуправляем и случаен.

**Задачи составления расписаний.**

**Задачи сетевого планирования и управления.** Актуальны при разработке сложных и дорогостоящих проектов. Тут рассматривается соотношение между сроком окончания крупного комплекса операций и моментами начала всех операций комплекса.

**Задачи выбора маршрута, или сетевые задачи.**

Среди сетевых задач наиболее распространенными являются: задача выбора кратчайшего пути между произвольными пунктами сети, задача коммивояжера, задача о максимальном потоке.

**Комбинированные задачи.** Включают в себя несколько типовых моделей задач одновременно.

При этом некоторые задачи исследования операций «не укладываются» ни в один из известных классов (и представляют наибольший интерес с научной точки зрения).

Важной особенностью исследования операций является стремление найти оптимальное решение поставленной задачи. Однако часто такое решение оказывается недостижимым...

## Читайте в номере

### Страницы повышения квалификации ..... 4–15

**Е.В. Андреева. Олимпиады по информатике. Пути к вершине**

Знакомство с классическими алгоритмами является необходимым (но, разумеется, не достаточным) условием успешного выступления на олимпиадах. В очередной лекции рассматривается алгоритм получения различных разбиений числа  $N$  на натуральные слагаемые, а также алгоритм поиска кратчайшего пути в графе.

**А.Г. Гейн. Введение в профессию “учитель информатики”**

“В современном обществе нельзя прожить без знаний... А дети видят, что многие малообразованные люди живут значительно лучше учителей и преподавателей вузов...”

Тема сегодняшней лекции — “Проблема мотивации”.

### На стенд в кабинете информатики ..... 16–17

**Winchester**

“Современный компьютер невозможно представить без основного носителя информации — жесткого диска”.

Все о винчестере, который фактически сам превратился в специализированный компьютер, обеспечивающий обмен информацией.

### Учебники..... 18–23

**Н.Турлынович. Моделирование**

Что общего между игрушечным корабликом и рисунком на экране компьютера?

Представляем вам главу из нового казахстанского учебника информатики, посвященную моделированию.

### Уроки ..... 24–32

**А.А. Дуванов. Азы информатики. Материалы Роботландского университета**

О чем может “рассказать” стол? Почему из семечка кабачка не вырастет елка? Картошка хранится в подвале. Хранится ли вместе с картошкой информация? Что такое информационный носитель? Какой самый важный информационный носитель придумал человек? Как хранят изображения? Где хранится информация в Интернете? Сегодня речь пойдет о хранении информации.

#### Читайте в следующих номерах

В следующем номере начнется публикация билетов для проведения итоговой аттестации по информатике в 9-м классе. Существенная особенность этого материала и его принципиальное отличие от публикаций в других изданиях заключаются в том, что мы предложим нашим подписчикам не просто “голые” вопросы, но и примерные ответы на них, специально подготовленные по заказу “Информатики”.

В № 10 будет опубликована первая статья из цикла материалов, посвященных реализации интересных и, что крайне важно, доступных школьникам проектов в среде электронной таблицы Excel с использованием VBA. Весь проект занимает менее трех десятков строчек на языке Visual Basic, но выглядит очень эффектно и может быть реализован учениками (разумеется, при помощи учителя) в течение одного-двух уроков.

Материалы рубрики “На стенд в кабинете информатики” в следующих двух номерах будут посвящены вопросам, связанным с историей и структурой сети Интернет. Наши постоянные читатели, несомненно, обратили внимание на то, что “стенды” вдвое увеличились в объеме, мы также стараемся делать их познавательными и разнообразными.

#### День учителя информатики в Москве

15 апреля,

Московский городской дом учителя  
(улица Пушкинская, дом 4, строение 2,  
станция метро “Кузнецкий мост”)



Дорогие коллеги! Пожалуйста, постарайтесь найти возможность прийти 15 апреля в Дом учителя. Мы готовим замечательную программу, будет очень интересно. Подробная информация о Дне учителя информатики будет опубликована в № 10.



# “ЖАРКОЕ ЛЕТО-2002”

УНИКАЛЬНЫЙ КОМПЛЕКТ  
ЗАМЕЧАТЕЛЬНЫХ МАТЕРИАЛОВ  
К НОВОМУ УЧЕБНОМУ ГОДУ

Как всегда, планируя очередное “Жаркое лето”, мы учитывали пожелания наших читателей и их мнения о сериях тематических номеров прошлых лет. При этом мы стремились отдавать предпочтение материалам, которые наши подписчики смогут использовать на всем протяжении следующего учебного года. Исключение составляет, пожалуй, лишь номер, посвященный А.Г. Звенигородскому. Но мы считаем принципиально важным беречь память о тех, кто сделал для школьной информатики в России неизмеримо много. Обращаем ваше внимание на необычный номер — “Интеллектуальные игры”. Не станем скрывать, что непосредственно к предмету “информатика” этот номер имеет мало отношения. Но все мы работаем в школе и знаем, что школьная жизнь не заканчивается звонком с последнего урока. Номер “Интеллектуальные игры” пригодится вам именно в этой “второй” жизни.

**Д.М. Златопольский.**

### Задачник по электронным таблицам (Excel)

Вопросы, связанные с обработкой информации с помощью электронных таблиц, занимают важное место в школьном курсе информатики. Но специализированного школьного задачника по электронным таблицам нет. Вернее, не было, а теперь есть. Его первая часть будет опубликована в летних номерах нашей газеты.

Задачник  
по электронным  
таблицам

**А.И. Терентьев.**

### Организация школьного web-сайта

Как сделать школьный web-сайт “с нуля”? Как связать компьютеры в сеть, какое программное обеспечение выбрать и как его установить? Как, наконец, заставить все это “хозяйство” работать? В одном из летних номеров мы познакомим наших читателей с ответами на эти (и не только на эти!) вопросы.

Организация  
школьного  
web-сайта

**А.И. Сенокосов.**

### Информатика и информатизация школы.

#### Практические решения

Допустим, вы сделали школьный web-сайт. Протянули провода, настроили локальную сеть, установили программное обеспечение. Все работает. А что же дальше? Как организовать информационное наполнение сайта? Как “встроить” web-сайт в школьную жизнь? Все, кто решал эти вопросы, знают, что они-то и являются настоящими **вопросами**.

Информатика  
и информатизация  
школы

**Д.М. Златопольский.**

### Внеклассная работа по информатике.

#### Избранные задания

Летом мы предложим подписчикам целый номер с заданиями, которые можно использовать на викторинах, конкурсах и других внеклассных мероприятиях по информатике. В него войдут множество новых заданий, а также лучшие из опубликованных ранее материалов популярной рубрики нашей газеты.

Внеклассная  
работа  
по информатике

**О.Г.А. Звенигородском.**

### Редактор-составитель — Н.А. Юнерман

История становления школьной информатики не может быть написана без страниц, посвященных исследованиям и разработкам О.Г.А. Звенигородского. И сегодня остается актуальным воплощенное в них единство тематических и педагогических сторон работы с учащимися, живой практики и теоретического осмысления материала. 9 августа 2002 г. О.Г.А. Звенигородскому исполнилось бы 50 лет.

О.Г.А. Звенигородский

**А.А. Дуванов.**

### DHTML-конструирование

“Бумажная” версия электронного учебника продолжает роботландский курс гипертекстового конструирования (ранее были опубликованы “HTML-конструирование” и “JavaScript-конструирование”). Новый учебник посвящен созданию динамических интерактивных приложений. В нем изложены основы CSS (каскадные таблицы стилей) и показаны способы управления содержимым страницы при помощи воздействий на гипертекстовую модель документа.



**Л.О. Сергеев.**

### Уроки по теме “Базы данных”

В этом тематическом выпуске мы предложим вниманию читателей цикл уроков по теме “Базы данных”. В качестве основного инструмента для изучения этой темы автор предлагает использовать язык SQL. Теоретический материал подкрепляется большим количеством разноуровневых заданий.



### Интеллектуальные игры

Что такое спортивное “Что? Где? Когда?” и чем оно отличается от телевизионного? Какие головоломки решают на чемпионатах мира? Во что можно поиграть без компьютера? Это и многое другое, а также вопросы, вопросы, вопросы... в специальном выпуске “Интеллектуальные игры”.



**А.Г. Гейн.**

### “Рыба” для учителя информатики

Авторы учебников обычно много говорят о том, что преподавать, и мало про то, как это делать. Свой взгляд на тематическое планирование предлагает один из авторов учебников по информатике.



**А.А. Дуванов.**

### Азы информатики.

#### Книга 3 — “Пишем на компьютере”

“Бумажная” версия третьей книги нового интерактивного курса для малышей “Азы информатики” (первая книга — “Знакомство с компьютером” — была опубликована в № 1, 2, публикация второй — “В мире информации” продолжается в текущих номерах). Заглавная тема третьей книги (современная обработка текстов) нагружена “анатомией” трех китов информатики: хранение, передача и обработка информации.



*Дорогие коллеги! “Информатика” распространяется только по подписке.*

*Подписаться на нашу газету можно по каталогу “Роспечати”, индекс подписки для индивидуальных подписчиков — 32291.*

# Олимпиады по информатике.

## Пути к вершине

Лекции читает Е.В. Андреева

### Лекция 8. Динамическое программирование, алгоритмы на графах

В прошлой лекции мы рассмотрели две группы задач, при решении которых используется динамическое программирование. Подзадачи, через которые выражалось решение, по формулировке представляли собой аналоги исходной задачи, только меньшей размерности. Однако иногда решение основной задачи приходится формулировать в терминах несколько модифицированных подзадач. Именно такие проблемы рассматриваются в следующем разделе.

#### Алгоритмы, использующие решение дополнительных подзадач

**Задача 9.** Требуется подсчитать количество различных разбиений числа  $N$  на натуральные слагаемые. Два разложения считаются различными, если одно нельзя получить из другого путем перестановки слагаемых.

**Решение.** Для того чтобы подсчитать количество различных разбиений числа  $N$  на произвольные натуральные слагаемые, предварительно подсчитаем количества разбиений на следующие группы слагаемых: 1) разбиения только на единицы (очевидно, что для любого числа такое разбиение единственно); 2) разбиения на единицы и двойки такие, что хотя бы одна двойка в разбиении присутствует, и т.д. Последнюю группу представляет само число  $N$ . Очевидно, что каждое разбиение числа  $N$  можно приписать только к одной из рассмотренных групп, в зависимости от значения  $j$  — максимального слагаемого в том или ином разбиении. Обозначим количество разбиений в каждой из групп  $t(N, j)$ . Тогда искомое количество разбиений равно сумме разбиений по всем возможным группам. Введение таких подзадач приводит к несложному способу подсчета числа разбиений. А именно: так как в любом из разбиений  $j$ -й группы присут-

ствует число  $j$ , то мы можем вычесть из  $N$  число  $j$  и сложить разбиения уже числа  $N - j$  на слагаемые, не превосходящие  $j$ . То есть мы пришли к следующему рекуррентному соотношению:

$$t(N, j) = \sum_{k=1}^j t(N - j, k), \quad N > j;$$

$$t(N, 1) = t(N, N) = 1; \quad (*)$$

$$t(N, j) = 0, \quad N < j.$$

Теперь очевидно, что если мы имеем возможность завести двумерный массив размером  $N \times N$  и будем заполнять его в порядке возрастания номеров строк, то задача будет легко решена. Однако легко заметить, что решения части подзадач никак не участвуют в формировании решения. Например, при вычислении количества разбиений числа 10 на слагаемые будут получены, но не использованы, значения  $t(9, j)$  для  $j = 2..9$  и т.д. Для того чтобы не производить лишних вычислений, применим динамическое программирование “сверху вниз” (все предыдущие задачи решались “снизу вверх”). Для этого задачу будем решать все же рекурсивно, используя формулу (\*), но ответы к уже решенным подзадачам будем запоминать в таблице. Первоначально таблица пуста (вернее, заполним элементы, значение которых по формуле (\*) равно 0 или 1, а остальные значения, например, числом  $-1$ ). Когда в процессе вычислений подзадача встречается первый раз, ее решение заносится в таблицу. В дальнейшем решение этой подзадачи берется из таблицы. Таким образом, мы получили прием улучшения рекурсивных алгоритмов, а “лишние” подзадачи теперь решаться не будут.

Приведем программу реализующую этот алгоритм.

```

var i, j, k, n : byte;
    sum : longint;
    table : array[1..120, 1..120] of longint;
function t(n, k : byte) : longint;
var i, s : byte;
begin
    if table[n, k] < 0 then
        {остальные элементы не пересчитываем}
        begin
            table[n, k] := 0;
            for i := 1 to k do
                inc(table[n, k], t(n - k, i))
            end;
            t := table[n, k]
        end;
end;
begin
    read(n);
    fillchar(table, sizeof(table), 0);
    for i := 1 to n do
        begin
            table[i, i] := 1;

```

План публикаций лекций курса “Олимпиады по информатике. Пути к вершине” на “Страницах повышения квалификации”.

Номер лекции	Номер газеты
1	38/2001
2	40/2001
3	42/2001
4	44/2001
5	46/2001
6	48/2001
7	6/2002
8	8/2002
9	10/2002
10	12/2002
11	14/2002
12	16/2002

```

    table[i, 1] := 1
  end;
} (неопределенные элементы метим -1)
  for i := 2 to n do
    for j := 2 to i - 1 do
      table[i, j] := -1;
    end;
  end;
sum := 0;
for i := 1 to n do
  sum := sum + t(n, i);
end;
writeln('sum=', sum);
end.

```

**Задача 10.** Плитки (Чемпионат школьников по программированию, Санкт-Петербург, 1999 г.).

У Пети имеется неограниченный набор красных, синих и зеленых плиток размером  $1 \times 1$ . Он выбирает ровно  $N$  плиток и выкладывает их в полосу. Например, при  $N = 10$  она может выглядеть следующим образом:

К	К	К	С	З	К	К	З	К	С
---	---	---	---	---	---	---	---	---	---

(Буквой “К” обозначена красная плитка, “С” — синяя, “З” — зеленая.)

После этого Петя заполняет следующую таблицу, которая в данном примере выглядит так:

	Красный	Синий	Зеленый
Красный	Y	Y	Y
Синий	Y	N	Y
Зеленый	Y	Y	N

В клетке на пересечении строки, отвечающей цвету  $A$ , и столбца, отвечающего цвету  $B$ , он записывает “Y”, если в его полоске найдется место, где рядом лежат плитки цветов  $A$  и  $B$ , и “N” в противном случае. Считается, что плитки лежат рядом, если у них есть общая сторона. (Очевидно, что таблица симметрична относительно главной диагонали — если плитки цветов  $A$  и  $B$  лежали рядом, то рядом лежали и плитки цветов  $B$  и  $A$ .) Назовем такую таблицу *диаграммой смежности* данной полосы.

Так, данная таблица представляет собой диаграмму смежности приведенной выше полосы.

Помогите Пете узнать, сколько различных полосок имеют определенную диаграмму смежности (заметьте, что полоски, являющиеся отражением друг друга, но не совпадающие, считаются разными. Так, полоска

С	К	З	К	К	З	С	К	К	К
---	---	---	---	---	---	---	---	---	---

не совпадает с полоской, приведенной в начале условия).

Первая строка входного файла содержит число  $N$  ( $1 \leq N \leq 100$ ). Следующие три строки входного файла, содержащие по три символа из набора {“Y”, “N”}, соответствуют трем строкам диаграммы смежности. Других символов, включая пробелы, во входном файле не содержится. Входные данные корректны, т.е. диаграмма смежности симметрична.

Выведите в выходной файл количество полосок длины  $N$ , имеющих приведенную во входном файле диаграмму смежности.

Ниже дан пример входного и выходного файлов.

Input.txt	Output.txt
10	4596
YYY	
YNY	
YYN	

*Решение.* Очевидно, что перебор всех возможных полосок в данной задаче невозможен, так как их количество может составить  $2^{100}$ , поэтому следует попытаться найти динамическую схему решения. Понятно, что, для того чтобы подсчитать количество полосок длины  $N$ , удовлетворяющих заданной диаграмме смежности, необходимо знать количество допустимых полосок длины  $N - 1$ , а также количество полосок, в диаграмме смежности которых один диагональный элемент или два симметричных недиагональных элемента равны “N” вместо “Y” в исходной диаграмме. Далее, при рассмотрении полосок длины  $N - 2$  потребуются знать количество полосок, удовлетворяющих еще большему количеству диаграмм смежности, и т.д. В результате на каком-то шаге нам может понадобиться информация о количестве полосок практически со всеми возможными диаграммами. Общее количество последних составляет  $2^6 = 64$  (уникальными, то есть не повторяющимися, а, значит, определяющими количество различных диаграмм, являются только 6 элементов). Так как при увеличении длины полоски диаграмма может измениться в зависимости от сочетания цветов в последнем (новом) и предпоследнем элементах, подсчитывать полоски следует отдельно для трех различных конечных элементов. Таким образом, количество хранимой информации возрастает до  $64 \times 3 = 192$  значений. Столько же значений будет получено в результате пересчета. Но благодаря тому, что количество полосок длины  $i$  выражается только через количество полосок длины  $i - 1$ , хранить нужно лишь эти  $2 \times 192 = 384$  значения. Несмотря на малый размер таблицы (массив `total` в программе), следует отметить, что ее размер экспоненциально зависит от одного из входных параметров — количества цветов  $k$ , а именно:  $2 \times k \times 2^{k(k+1)/2}$ . Например, для 8 цветов необходимо было бы хранить  $2^{40}$  элементов, что нереально. Этим данная задача отличается от рассмотренных ранее.

Осталось обсудить некоторые технические приемы, позволяющие написать довольно простую программу, реализующую описанный алгоритм. Если мы поставим в соответствие каждому из уникальных мест в диаграмме смежности свою степень двойки от  $2^0$  до  $2^5$  (см. массив констант `magic` в программе), то каждой диаграмме может быть поставлен в соответствие номер от 0 до 63, равный сумме тех степеней двоек, которые соответствуют значениям “Y” (см. процедуру `findcode`). Если мы подсчитываем количество полосок для диаграммы с номером  $j$ , то совместимость добавляемого цвета  $k$  стоявшему ранее последним цвету  $l$  согласно диаграмме  $j$  можно проверить так: `magic[k, 1] and j <> 0`. Данное условие, построенное с помощью битовой операции над целыми числами `and`, означает наличие в  $j$ -й диаграмме смежности элемента “Y” на пересечении  $k$ -й строки и  $l$ -го столбца (соответствующая степень двойки массива

magic содержится в двоичном представлении числа  $j$ ). Выражение  $j - \text{magic}[k, 1]$  соответствует замене в диаграмме с номером  $j$  упомянутого элемента “Y” на “N” (по-другому это выражение можно было бы записать как  $j \text{ xor } \text{magic}[k, 1]$ ). Подробнее о битовых операциях над целыми числами можно прочитать в [1]. Последний прием заключается в том, что мы не будем на каждом шаге переписывать полученные значения элементам массива, предназначенного для хранения результатов предыдущего шага. Для этого результаты для полосок четной длины  $i$  будем помещать в половину массива `total` с первым индексом 0, а нечетные — с индексом 1. В любом из этих случаев значения предыдущего шага доступны по индексу  $[1 - i \bmod 2]$ . Кроме того, ответ на решение этой задачи при всех  $N$ , удовлетворяющих условию, требует самостоятельной организации вычислений с помощью так называемой “длинной арифметики” (см., например, [1, 3]).

Приведем программу для решения этой задачи, но использующую вместо “длинной арифметики” тип данных `extended`, сохраняющий максимально возможное количество значащих цифр (попробуйте модернизировать программу самостоятельно). То есть не для всех значений  $N$  ответ будет вычислен точно. Но, так как для получения результата используется только сложение целых чисел, потери точности при промежуточных вычислениях не будет, по крайней мере пока ответ не станет превышать  $2^{63}$ .

```
{N+}
const magic: array [1..3, 1..3] of byte =
  ((1, 2, 4),
   (2, 8, 16),
   (4, 16, 32));
var n, i, j, k, l, code : longint;
    can : array [1..3, 1..3] of boolean;
    total : array [0..1, 1..3, 0..63] of extended;
    answer : extended;
procedure readdata;
var s : string;
    i, j : byte;
begin
  readln(n);
  fillchar(can, sizeof(can), false);
  for i := 1 to 3 do
  begin
    readln(s);
    for j := 1 to 3 do
      if upcase(s[j]) = 'Y' then
        can[i, j] := true
    end
  end;
procedure findcode;
var i, j : byte;
begin
  {переводим диаграмму смежности в число}
  code := 0;
  for i := 1 to 3 do
    for j := i to 3 do
      if can[i, j] then
        code := code + magic[i, j]
    end;
  end;
assign(input, 'input.txt');
reset(input);
```

```
assign(output, 'output.txt');
rewrite(output);
readdata;
findcode;
fillchar(total, sizeof(total), 0);
{количество полосок длины l}
for i := 1 to 3 do
  total[1, i, 0] := 1;
for i := 2 to n do
for j := 0 to 63 do
for k := 1 to 3 do
  {считаем полоски длины i,
   с диаграммой смежности j
   и оканчивающиеся цветом k}
  begin
    total[i mod 2, k, j] := 0;
    for l := 1 to 3 do
      {цикл по конечному цвету полосок длины i - 1}
      if magic[k, l] and j <> 0 then
        {цвета l и k могут соседствовать
         согласно диаграмме смежности j}
        begin
          total[i mod 2, k, j] :=
            total[i mod 2, k, j] +
            total[1 - i mod 2, l, j];
          total[i mod 2, k, j] :=
            total[i mod 2, k, j] +
            total[1 - i mod 2, l, j - magic[k, l]]
        end
      end;
    answer := 0;
    {суммируем количество полосок с диаграммой
     смежности code и различными окончаниями}
    for i := 1 to 3 do
      answer := answer + total[n mod 2, i, code];
    writeln(answer:0:0)
  end.
```

Похожая задача (“Симпатичные узоры”) предлагалась и на I Всероссийской командной олимпиаде по программированию. Ее условие и решение можно прочитать в [2].

**Задача 11.** Паркет (Задача VI Всероссийской олимпиады по информатике, 1994 г.).

Комнату размером  $N \times M$  единиц требуется покрыть одинаковыми паркетными плитками размером  $2 \times 1$  единицу без пропусков и наложений ( $1 \leq N \leq 20, 1 \leq M \leq 8$ ). Требуется определить количество всех возможных способов укладки паркета для конкретных значений  $N$  и  $M$ .

*Решение.* Пусть  $M$  — ширина комнаты, которую мы зафиксируем. Попытаемся выразить искомое количество упаковок паркета для комнаты длины  $N$  через количество упаковок для комнаты длиной  $N - 1$ . Однако очевидно, что сделать это не удастся, так как существует еще множество упаковок, в которых часть плиток пересекает границу между такими комнатами. Следовательно, нам опять придется решать дополнительное число подзадач. А именно, введем обобщенное понятие укладки комнаты длиной  $N - 1$ : первая часть комнаты длиной  $N - 2$  уложена плотно, а в  $(N - 1)$ -й единице измерения длины комнаты могут находиться пустоты (в  $N$ -й единице измерения паркета нет). Если наличие плитки в

$(N - 1)$ -й единице измерения обозначить 1, а ее отсутствие — 0, то количество различных окончаний подобных укладок можно пронумеровать двоичными числами от 0 до  $2^M - 1$ . Если количество укладок для каждого из окончаний нам известно (часть из них могут оказаться нереализуемыми, то есть соответствующее количество укладок будет равно 0), то мы сможем подсчитать количество различных укладок комнаты длины  $N$ . При этом придется проверять совместимость окончаний. Окончания будем считать совместимыми, если путем добавления целого числа плиток к укладке длиной  $N - 1$  с окончанием  $j$ , таких, что каждая из них увеличивает длину укладки до  $N$ , мы можем получить окончание  $i$  укладки длиной  $N$ . Если способ совмещения укладок существует, то по построению он единственен. Тогда для определения количества укладок с окончанием  $i$  длиной  $N$  необходимо просуммировать количества укладок длиной  $N - 1$  с совместимыми окончаниями. Для комнаты нулевой длины будем считать количество укладок равным 1. Формирование динамической схемы закончено. Количество хранимых в программе значений при этом равно  $2 \times 2^M = 2^{M+1}$ , то есть оно экспоненциально зависит от одного из параметров задачи и существенно его увеличить не представляется возможным. В нашем случае оно равно 512, то есть применение табличного метода решения оказывается реальным. Ответ на вопрос задачи будет получен на  $N$ -м шаге алгоритма в элементе таблицы с номером  $2^M - 1$ . При максимальном по условию задачи размере комнаты для получения ответа опять потребуется “длинная арифметика”.

Схему программы для решения этой задачи, которая проще предыдущей, можно найти, например, в [3].

После рассмотрения задач 9—11 может сложиться впечатление, что к данному классу относятся лишь задачи подсчета количеств тех или иных конфигураций, в том числе комбинаторных. Конечно же это не так. Примером оптимизационной задачи, решение которой основано на аналогичных идеях, служит задача “Бизнес-классики”, предлагавшаяся на XIII Всероссийской олимпиаде по программированию (см. [4]).

Многие прикладные и олимпиадные задачи легко сформулировать в терминах такой структуры данных, как граф. Для ряда подобных задач хорошо изучены эффективные (полиномиальные) алгоритмы их решения. Рассмотрим в данной лекции те из них, которые используют идеи динамического программирования. Но прежде необходимо познакомиться с некоторыми терминами, встречающимися при описании этой структуры.

## Основные определения теории графов

*Графом* называется пара  $G = \langle V, E \rangle$ , где  $V$  — некоторое множество, которое называют множеством *вершин* графа, а  $E$  — отношение на  $V$  ( $E \subset V \times V$ ) — множество *ребер* графа. То есть все ребра из множества  $E$  соединяют некоторые пары точек из  $V$ .

Если отношение  $E$  симметричное (т.е.  $(u, v) \in E \Leftrightarrow (v, u) \in E$ ), то граф называют *неориентированным*, в противном случае граф называют *ориентированным*. Фак-

тически для каждого из ребер ориентированного графа указаны начало и конец, то есть пара  $(u, v)$  упорядочена, а в неориентированном графе  $(u, v) = (v, u)$ .

Если в графе существует ребро  $(u, v)$ , то говорят, что вершина  $v$  *смежна* с вершиной  $u$  (в ориентированном графе отношение смежности несимметрично).

*Путем* из вершины  $u$  в вершину  $v$  длиной  $k$  ребер называют последовательность ребер графа

$$U = \langle (u, v_1), (v_1, v_2), \dots, (v_{k-1}, v) \rangle.$$

Часто тот же путь обозначают последовательностью вершин  $\langle u, v_1, \dots, v_{k-1}, v \rangle$ . Если для данных вершин  $u, v$  существует путь из  $u$  в  $v$ , то говорят, что вершина  $v$  *достижима* из  $u$ . Путь называется *простым*, если все вершины в нем различны. *Циклом* называется путь, в котором начальная вершина совпадает с конечной. При этом циклы, отличающиеся лишь номером начальной точки, отождествляются.

Граф называется *связанным*, если для любой пары его вершин существует путь из одной вершины в другую.

Если каждому ребру графа приписано какое-то число (*вес*), то граф называют *взвешенным*.

При программировании вершины графа обычно сопоставляют числам от 1 до  $N$ , где  $N = |V|$  — количество вершин графа, и рассматривают  $V = \{1, 2, \dots, N\}$ . Ребра нумеруют числами от 1 до  $M$ , где  $M = |E|$ . Для хранения графа в программе можно применять различные методы. Самым простым является хранение *матрицы смежности*, т.е. двумерного массива, скажем,  $A$ , где для невзвешенного графа  $A[i][j] = \text{true}$  (или 1), если  $(i, j) \in E$ , и  $A[i][j] = \text{false}$  (или 0) в противном случае. Для взвешенного графа  $A[i][j]$  равно весу соответствующего ребра, а отсутствие ребра в ряде задач удобно обозначать бесконечностью. Для неориентированных графов матрица смежности всегда симметрична относительно главной диагонали ( $i = j$ ). С помощью матрицы смежности легко проверить, существует ли в графе ребро, соединяющее вершину  $i$  с вершиной  $j$ . Основной же ее недостаток заключается в том, что матрица смежности требует, чтобы объем памяти был достаточен для хранения  $N^2$  значений, даже если ребер в графе существенно меньше, чем  $N^2$ . Это не позволяет построить алгоритм со временем порядка  $O(N)$  для графов, имеющих  $O(N)$  ребер.

Этого недостатка лишены такие способы хранения графа, как одномерный массив длины  $N$  списков или множеств вершин. В таком массиве каждый элемент соответствует одной из вершин и содержит список или множество вершин, смежных ей.

Для реализации некоторых алгоритмов более удобным является описание графа путем перечисления его ребер. В этом случае хранить его можно в одномерном массиве длиной  $M$ , каждый элемент которого содержит запись о номерах начальной и конечной вершин ребра, а также его весе в случае взвешенного графа.

Наконец, при решении задач на графах, в том числе и с помощью компьютера, часто используется его графическое представление. Вершины графа изображают на плоскости в виде точек или маленьких кружков, а ребра — в виде линий (не обязательно прямых), соеди-

няющих соответствующие пары вершин, для неориентированного графа и стрелок — для ориентированного (если ребро направлено из  $u$  в  $v$ , то из точки, изображающей вершину  $u$ , проводят стрелку в вершину  $v$ ).

Графы широко используются в различных областях науки (в том числе в истории!!!) и техники для моделирования отношений между объектами. Объекты соответствуют вершинам графа, а ребра — отношениям между объектами). Подробнее об этой структуре данных можно прочитать в [5—7].

## Поиск пути между парой вершин невзвешенного графа

Пусть мы имеем произвольный граф, ориентированный или неориентированный. Если в невзвешенном графе существует путь, то назовем длиной пути количество ребер в нем. Если пути нет вообще, то расстояние считается бесконечным. Путь минимальной длины при этом называется *кратчайшим путем* в графе. Легко показать, что любые части кратчайшего пути также являются кратчайшими путями между соответствующими вершинами. Ведь если это не так, то есть существует отрезок кратчайшего пути, между концами которого можно построить более короткий путь, то мы можем заменить этот отрезок кратчайшего пути между вершинами  $u$  и  $v$  на более короткий, тем самым уменьшив и длину кратчайшего пути между  $u$  и  $v$ , что невозможно. Это свойство кратчайших путей позволяет решать задачу их нахождения методом динамического программирования. Покажем сначала, как можно записать “волновой алгоритм” так, что задача поиска кратчайшего пути между двумя вершинами графа будет решаться за  $O(N^2)$  действий.

**Задача 12.** Для линий метрополитена некоторого города известно, между какими парами линий есть пересадочная станция. Необходимо определить, за сколько пересадок можно добраться с линии  $m$  на линию  $n$ , или сообщить, что сделать это невозможно.

**Решение.** Такой метрополитен удобно описывать с помощью графа, вершины которого есть линии метрополитена (а не станции!!!), а наличие ребра между вершинами  $i$  и  $j$  графа соответствует наличию пересадочной станции между линиями с номерами  $i$  и  $j$ . Представим этот граф с помощью массива множеств (переменная  $ss$  в программе), в  $i$ -м элементе этого массива содержится множество всех линий, на которые можно попасть с линии  $i$  за одну пересадку. Результат будем получать с помощью множества  $s$  на каждом шаге алгоритма, содержащего номера всех линий, на которые можно попасть с исходной линии  $m$  за  $k$  пересадок. Заметим, что если вершина  $n$  нашего графа достижима из вершины  $m$  (говорят, что они находятся в одной *компоненте связности*), то искомое число пересадок меньше общего количества линий  $nn$ . Так как даже если после каждой из первых  $nn - 1$  пересадок мы попадали на новую линию, то после следующей пересадки мы обязательно окажемся на какой-то из линий повторно, ведь

их всего  $nn$ . Поэтому если наш алгоритм не завершился за  $nn - 1$  шаг, то граф не связан и дальнейший поиск пути бесполезен (заметим, что наличие пути между двумя конкретными вершинами не доказывает его связность, а исследовать все пары вершин с помощью предложенного алгоритма для анализа связности неэффективно).

Программа для решения задачи представлена ниже.

```
const nn = 200; {число линий}
type myset = set of 0..nn;
var i, m, n, k : byte;
    ss : array[1..nn] of myset;
    s, s1 : myset;
begin
  ...{считываем входные данные}
  s := [m]; k := 0;
  while not (n in s) and (k < nn - 1) do
    begin
      k := k + 1;
      s1 := s; s := [];
      for i := 1 to nn do
        if i in s1 then
          {добавляем к s вершины,
           достижимые из i}
          s := s + ss[i]
        end;
      if n in s then writeln(k) else
        writeln('it is impossible')
      end.
end.
```

Заметим, что предложенный при решении задачи 12 алгоритм можно модифицировать так, чтобы он находил длину кратчайшего пути от исходной вершины до всех других вершин графа, причем асимптотическое время его работы не изменится. Несмотря на хорошие временные характеристики, область применения алгоритма ограничена размером типа “множество” в Паскале. Избежать этого ограничения можно, используя такой способ представления графа, как массив списков вершин, смежных с данной. О способах реализации динамических структур данных и, в частности, списков см., например, [8].

Пусть теперь требуется определить наличие пути сразу для всех пар вершин графа. Такая задача для невзвешенного графа называется *транзитивным замыканием*. Рассмотрим ее решение на примере следующей проблемы.

**Задача 13.** Пусть для некоторых пар переменных известно, что значение одной из них не больше значения другой. Выписать остальные пары из упомянутых переменных, для которых, используя транзитивность операции “ $\leq$ ”, можно также сказать, что значение одной из них не превосходит значения другой.

**Решение.** Обозначим данными переменными вершины графа, а знание о наличии между двумя переменными отношения “ $\leq$ ” — ориентированными ребрами. Для некоторой пары вершин справедливо, что значение одной  $\leq$  значения другой, если в построенном ориентированном графе существует путь из первой из упомянутых вершин во вторую. Тогда для решения задачи можно воспользо-

ваться следующим алгоритмом Уоршола [5, 6]. Пусть  $A$  — матрица, изначально равная матрице смежности графа, записанной с помощью логических констант *true* и *false*. На  $k$ -м шаге алгоритма значение *true* в элементе матрицы  $A[i, j]$  будет означать, что из вершины  $i$  в вершину  $j$  существует путь, который проходит через некоторые вершины с номерами, не превосходящими  $k - 1$ . Если же через упомянутые вершины пути нет ( $A[i, j] = \text{false}$ ), но существует путь из вершины  $i$  в вершину  $k$  и путь из вершины  $k$  в вершину  $j$ , то значение данного элемента матрицы становится равным *true*. Покажем, как написать фрагмент программы, реализующий этот алгоритм без использования условных операторов:

```
c := a; {запоминаем матрицу смежности}
for k := 1 to nn do
  for i := 1 to nn do
    for j := 1 to nn do
      a[i, j] := a[i, j] or a[i, k] and a[k, j];
```

Краткость говорит здесь сама за себя. В результате выполнения трех вложенных циклов (то есть мы имеем алгоритм, работающий за  $N^3$  операций), порядок которых очень важен, в матрице  $a$  мы фактически получим ответ на вопрос задачи. Распечатать его можно так:

```
for i := 1 to nn do
  for j := 1 to nn do
    if a[i, j] xor c[i, j] then writeln(i, ' ', j);
```

Если же требуется найти длины кратчайших путей для всех пар вершин, то, если каждому ребру графа приписать вес, равный единице, решение задачи будет полностью совпадать с решением той же задачи для взвешенного графа (см. далее). Поэтому отдельно мы рассматривать его не будем.

## Пути минимальной длины во взвешенном графе

Длиной пути между двумя вершинами во взвешенном графе называется сумма весов ребер, составляющих этот путь. В отличие от невзвешенного графа наличие ребра между двумя вершинами не гарантирует, что кратчайший путь между ними состоит из этого ребра. Зачастую суммарный вес пути, состоящего из двух, трех и более ребер, может оказаться меньше веса одного ребра, поэтому даже для *полного* графа (то есть графа, между каждой из пар вершин которого существует ребро, а в случае ориентированного графа — два ребра, направленных в противоположные стороны) задача поиска кратчайших путей имеет смысл. Понятие кратчайшего пути пока будем рассматривать только для графов, все ребра которых имеют неотрицательный вес.

Наиболее просто найти кратчайший путь между каждой из пар вершин можно с помощью алгоритма Флойда [5—7], основанного на той же идее, что и алгоритм Уоршола. Пусть в элементе матрицы  $A[i, j]$  хранится длина кратчайшего пути из вершины  $i$  в вершину  $j$ , который проходит через некоторые вершины с номерами, не превосходящими  $k - 1$ . Если же длины пути из вершины  $i$  в вершину  $k$  и пути из вершины  $k$  в вершину  $j$

таковы, что их сумма меньше, чем значение данного элемента матрицы, то его следует переопределить. То есть в реализации алгоритма Уоршола следует заменить операцию *and* на “+”, а *or* — на нахождение минимума из двух величин. Для реализации алгоритма массив  $a$  первоначально следует заполнить элементами матрицы смежности, обозначая отсутствие ребра между двумя вершинами “бесконечностью” — числом, заведомо превосходящим длину любого пути в рассматриваемом графе, но меньшим, чем максимальное значение используемого типа данных, чтобы было возможно выполнять с ним арифметические операции. В этом случае можно избежать дополнительных проверок.

Если же нам требуется найти сам кратчайший путь, а не его длину, то при каждом улучшении пути между двумя вершинами мы в соответствующем элементе вспомогательного массива (в программе —  $w$ ) будем запоминать номер той вершины, через которую кратчайший путь проходит, а затем с помощью элегантной рекурсивной процедуры будем его печатать. Идея рекурсии заключается в том, что если мы знаем, что кратчайший путь от вершины  $i$  до вершины  $j$  проходит через вершину  $k$ , то мы можем обратиться к той же самой процедуре с частями пути от  $i$  до  $k$  и от  $k$  до  $j$ . Рекурсивный спуск заканчивается, когда на кратчайшем пути между двумя вершинами не окажется промежуточных вершин.

Приведем фрагмент программы, реализующий алгоритм Флойда и печатающий кратчайшие пути между всеми парами вершин графа.

```
procedure way(i, j : integer);
{печатает путь между вершинами i и j}
begin
  if w[i, j] = 0 then write(' ', j)
  {печатаем только вершину j,
  чтобы избежать повторов}
  else
    begin
      way(i, w[i, j]); way(w[i, j], j)
    end
end;
begin
  ...{заполняем матрицу смежности}
  for k := 1 to nn do
    for i := 1 to nn do
      for j := 1 to nn do
        if a[i, k] + a[k, j] < c[i, j] then
          begin
            a[i, j] := a[i, k] + a[k, j];
            w[i, j] := k
          end;
        for i := 1 to nn do
          for j := 1 to nn do
            begin
              write(i);
              if i <> j then way(i, j);
              writeln
            end
          end
        end
      end
end.
```

Алгоритм Флойда хорош всем, кроме одного: он требует хранить матрицу смежности, а это не всегда возможно. Кроме того, для определения длины кратчайшего

пути между двумя конкретными вершинами его упростить невозможно (то есть все равно придется считать пути между всеми парами вершин). Если вес любого ребра в графе вычисляется по некоторой формуле (например, как расстояние между двумя точками на плоскости, если таковыми являются вершины нашего графа), то матрицу смежности можно не создавать вообще, а в процессе выполнения программы обращаться к функции вычисления веса ребра, соединяющего вершины  $i$  и  $j$ :  $a(i, j)$ .

В этом случае для определения кратчайшего пути между вершинами  $s$  и  $t$  используют алгоритм Дейкстры [5—7]. Все вершины в процессе работы этого алгоритма разбиты на два множества: те, до которых кратчайший путь из вершины  $s$  уже известен (в программе они помечены значениями `true` одномерного булевого массива  $b$ ), и все остальные. Сначала в первом множестве находится только вершина  $s$ . На каждом шаге к нему добавляется одна из вершин, текущее известное расстояние до которой минимально среди всех вершин из второго множества, обозначим ее  $p$ . Первоначально текущие расстояния (в программе они хранятся в одномерном массиве  $l$ ) от  $s$  до остальных вершин равны  $\infty$ , а расстояние до  $s$  равно 0,  $p$  также равна  $s$ . На очередном же шаге мы пытаемся улучшить длину пути до каждой из вершин второго множества, сравнивая выражения  $l[p] + a(p, i)$  и  $l[i]$ . Нужно показать, почему минимальное из значений  $l$ , рассматриваемых на текущем шаге, является длиной кратчайшего пути до соответствующей вершины, а, следовательно, этот путь содержит только вершины из первого множества. Если это не так, то есть кратчайший путь до этой вершины содержит и вершины из второго множества, то минимальной оказалась бы длина пути от  $s$  до одной из этих вершин. Значит, кратчайший путь до вершины  $p$  проходит только через вершины первого множества и больше его пересчитывать не нужно.

Приведем схему программы, реализующей этот алгоритм (функцию  $a(i, j)$  и значение “бесконечности” определять не будем):

```
for i := 1 to nn do
  begin
    l[i] = ∞;
    b[i] := false
  end;
p := s; l[s] := 0;
b[s] := true;
```

```
f := true; {стоит ли искать дальше}
while (p <> t) and f do
  begin
    f := false;
    for i := 1 to nn do
      if not b[i] then
        if l[p] + a(p, i) < l[i] then
          l[i] := l[p] + a(p, i);
    min := t; {важно, что b[t] = false}
    for i := 1 to n do
      if (not b[i]) and (l[i] < l[min]) then
        min := i;
    if l[min] < ∞ then
      begin
        p := min; b[p] := true; f := true
      end
  end;
```

Несложно подсчитать, что трудоемкость алгоритма составляет  $O(N^2)$ , что окупает некоторые сложности в его реализации. Как и в случае применения “волнового” алгоритма в невзвешенном графе, асимптотическая оценка не изменится, если нам потребуется подсчитать длину пути от  $s$  до каждой из вершин графа. Поэтому, как и в алгоритме Флойда, длины кратчайших путей между всеми парами вершин могут быть рассчитаны за  $O(N^3)$  операций.

В следующей лекции будут рассмотрены другие алгоритмы на графах.

### Литература

1. Андреева Е., Фалина И. Системы счисления и компьютерная арифметика. М.: Лаборатория базовых знаний, 2000.
2. Станкевич А.С. Решение задач I Всероссийской командной олимпиады по программированию. “Информатика” № 12, 2001.
3. Окулов С.М. 100 задач по информатике. Киров: Изд-во ВГПУ, 2000.
4. Андреева Е.В. Решение задач XIII Всероссийской олимпиады по информатике. “Информатика” № 19, 2001.
5. Ахо А.А., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. М.: Вильямс, 2000.
6. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦНМО, 2000.
7. Липский В. Комбинаторика для программистов. М.: Мир, 1988.
8. Вирт Н. Алгоритмы и структуры данных. СПб.: Невский диалект, 2001.

### Калейдоскоп

Свободный доступ в Интернет получили 500 заключенных воспитательной колонии для несовершеннолетних в поселке Ново-Гришино (Дмитровский район Московской области). 26 января здесь открылся учебный Интернет-класс, оборудованный специалистами центра новых информационных тех-

### Из тюрьмы можно совершить побег, но виртуальный

нологий при Новгородском государственном университете имени Ярослава Мудрого.

Как передает ИА “Росбалт”, работы в колонии, продолжавшиеся два месяца, выполнили программисты из Великого Новгорода по заказу республиканской Федерации Интернет Образования”. Стоимость работ

составила более 40 тыс. долларов. Преподавать информатику несовершеннолетним заключенным будут четыре программиста, которые прошли специальную подготовку в Новгородском центре новых информационных технологий.

По материалам сайта [www.regions.ru](http://www.regions.ru)

# Введение в профессию “учитель информатики”

Лекции читает А.Г. Гейн

*На уроки информатики школьники любых классов идут с удовольствием, и связано это пока с тем, что компьютер сам по себе уже является стимулом к изучению предмета. Но проникновение компьютеров во многие сферы жизнедеятельности человека со временем притупят этот фактор, это, дорогие коллеги, надо учитывать.*

И.Н. Фалина. “Современные педагогические технологии и частные методики обучения информатике”

## Лекция 2. Проблема мотивации

В предыдущей лекции мы обсуждали, какие интересы преследует общество, принуждая школьников изучать информатику. Слово “принуждая” употреблено совсем не случайно — ведь у школьника нет права выбора учить этот предмет или не учить, значит, он *принужден* его учить. Общество, конечно, старается всячески убедить школьника, как важно ему этот предмет изучать. В основном, правда, методом запугивания. Судите сами по тем выдержкам, которые приведены ниже.

“Чем выше уровень развития страны, тем более интенсивно происходит процесс, получивший название “информатизация”. Его конечной целью является формирование общества, в котором живут и работают информационно грамотные люди, умело использующие компьютерные системы и технологии. Если вы не будете обладать необходимым уровнем информационной культуры, то жить в информационном обществе будет нелегко” ([4], с. 5).

Итак, государство создает общество определенного типа и требует от его членов соответствия этому обществу, а не то, грозят авторы учебника, плохо будет.

“Человек, не знакомый с компьютером, может оказаться совершенно не приспособленным к жизни в наш век новых информационных, компьютерных технологий” ([2], с. 5).

Чуть мягче, но, по существу, то же самое.

“Сегодня информатика и вычислительная техника, проникнув во многие сферы человеческой деятельности человека, постепенно становятся неотъемлемой частью практически всех профессий, прочно входят в наш быт, образование, культуру. Именно поэтому знание информатики, умение использовать компьютер совершенно необходимы любому образованному человеку в современном обществе” ([5], с. 15).

То же, что в [2], с одной особенностью: знание компьютера нужно не каждому, а только тому, кто хочет быть образованным. Ну а кто ж не хочет? И даже если не хочет, закон об обязательном общем образовании никто не отменял.

Все, что говорят авторы приведенных выше цитат, верно. Как и то, что человек не может жить вне общества и должен подчиняться его требованиям. Но это, повторю еще раз, принудительная мотивация. Конечно, это не розги, которыми наказывали нерадивых учеников прошлого века, но все равно здесь нет обращения к внутренней, личной потребности учащегося изучать данный школьный предмет.

Отметим и то, что с подобной мотивацией учащийся сталкивается при изучении практически каждой школьной дисциплины: “В современном обществе нельзя прожить без знаний физики (химии, биологии, истории, ... — подставляйте сюда любой предмет из школьного расписания)”. А в реалии дети видят, что многие малообразованные люди живут куда лучше учителей и преподавателей вузов. Так что данный прием создания мотивации представляется малоэффективным.

Однако внутренняя мотивация у школьников к изучению информатики есть. Ведь, как отмечалось в предыдущей лекции, редко можно услышать от школьника фразу “Зачем мне информатика, я не собираюсь быть тем-то и тем-то?”. Обычно такой протест у учеников по отношению к информатике возникает, когда в ее изучении есть какой-то перекоп, например, в сторону математических аспектов информатики (глубокое изучение таких разделов, как *теория* алгоритмов, математическая логика, методы вычислений и т.п.).

По-видимому (по крайней мере это мнение разделяется многими исследователями), мотивом к изучению информатики в первую очередь выступает интерес к компьютеру. Он завораживает детей тайной своей могущественности и демонстрацией все новых возможностей. Он готов быть другом и помощником, он способен развлечь и связать со всем миром.

И авторы многих сегодняшних учебников хорошо знают, что привлекательность курса информатики обеспечивается именно компьютерной составляющей. Поэтому в мотивационных введениях к своим учебникам они вместо объявления целей и задач курса информатики, продиктованных социальным заказом, делают ставку именно на компьютерную компоненту. Вот соответствующие цитаты.

“О чудесных возможностях компьютеров слышаны все. Да и сами они уже давно не в диковинку. Конечно, вам хочется познакомиться с ними поближе. И в первую очередь научиться применять компьютер для решения разнообразных задач” ([2], с. 3).

План публикации лекций курса “Введение в специальность “учитель информатики” на “Страницах повышения квалификации”.

Номер лекции	Номер газеты
1	6/2002
2	8/2002
3	10/2002
4	12/2002
5	14/2002
6	16/2002

“Вы начинаете знакомство с новым для вас предметом — информатикой. Часто думают, что информатика — это о том, как работать на компьютере. Да, конечно, и об этом тоже, но только на первый взгляд, если не пытаться разобраться, зачем надо уметь работать на компьютере и почему компьютер стал предметом чуть ли не первой необходимости для ваших родителей, старших братьев и сестер, многих знакомых вам людей. На самом деле дисциплина эта настолько многогранна, что четко определить ее границы практически невозможно.

*Информатика — область человеческой деятельности, связанной с процессами хранения, преобразования и передачи информации.*

Ваша задача в процессе изучения информатики — не только научиться пользоваться компьютером, но и осознать его возможности, определить роль, которую он играет в нашей жизни, понять, какую помощь он может оказать вам при исследовании окружающего мира” ([3], с. 3).

Эти фрагменты мотивационного введения из разных учебников цитировались в задании 2 предыдущей лекции; там же отмечалось, что при такой мотивации истинные цели и задачи изучения курса информации остаются скрытыми от школьников. Вопрос в том задании собственно и состоял в том, почему это происходит. Выше мы фактически сформулировали ответ на этот вопрос: авторы хотят достичь максимального эффекта в создании мотивации, а таковой дает обращение к внутренним мотивам обучения. Для школьников сегодня таким внутренним мотивом является изучение возможностей компьютерной технологии.

Но перечитайте еще раз эпиграф, и станет ясно, что эта мотивация не вечна.

Впрочем, некоторые учебники, следуя, по-видимому, вузовским канонам, вообще не содержат мотивационного введения, не предпринимая никаких попыток объяснить, зачем ученику надо изучать данный предмет<sup>1</sup>. Это означает, что решение проблемы мотивации просто перекладывается на плечи учителя.

При этом мы пока обсуждали мотивацию курса в целом, а ведь необходимо мотивировать изучение каждого раздела, каждой темы и даже введение отдельных понятий. Но при всем этом разнообразии основными мотивирующими элементами остаются интерес и практическая значимость предлагаемого к освоению материала. Сразу скажем, что, несмотря на то что декларации “Я не буду это учить, потому что это никогда не понадобится”, звучат в устах учеников гораздо чаще, чем “Я не буду это учить, потому что это не интересно”, в создании мотивации интерес всегда имеет приоритет над прагматикой. Каждый сам легко приведет примеры готовности любого человека тратить значительные усилия на усвоение совершенно бесполезного, но интересного, но как трудно, только осознавая необходимость того или иного дела, им заниматься<sup>2</sup>.

Начнем с обсуждения путей создания мотивации курса в целом. Конечно, есть много разных вариантов. Вот один из них (см. [1], с. 3).

<sup>1</sup> Для вузовских учебников это естественно — человек уже выбрал специальность и теперь без лишних вопросов обязан изучать программу, предусмотренную данной специальностью.

<sup>2</sup> Литературное описание этого феномена всем известно — история побелки забора Томом Сойером.

“Вся наша жизнь — постоянное решение задач. Как починить велосипед и как вывести на орбиту космический корабль, как воспитать подаренного вам щенка и что предпринять для спасения редкой породы птиц, как организовать школьный вечер и как рационально использовать запасы нефти и газа — многообразие жизненных задач беспредельно. Конечно, каждый школьный предмет предлагает свой подход к решению задач. Скажем, проблемы охраны природы можно обсуждать и на уроках математики, и на уроках химии, и на уроках литературы. Есть свой подход к решению задач и у информатики. Суть его в использовании компьютера...”

Если заявка на жизненную направленность не будет подтверждаться в курсе, а все применение компьютера сведется к вычислению суммы последовательности чисел и другим подобным задачам, то мотивационная ценность подобной декларации весьма мала. Конечно, весьма непросто подбирать задачи с жизненным содержанием, которые бы допускали доступное для школьника решение, да еще и были бы направлены на достижение требуемых целей обучения. Но если такие задачи подобрать удастся, то учащиеся понимают, что их в информатике учат тому, что может им потребоваться в реальной жизни. Приведем несколько примеров таких задач.

Пример 1 ([2], с. 30; автор задачи — А.И. Сеносков)

Представьте себе, что в вашей школе организуется какая-нибудь автобусная экскурсия. Администрация школы, зная, сколько человек желает принять в ней участие, заказала соответствующее число автобусов. Естественно, суммарное количество мест в них оказалось чуть больше количества участников экскурсии. Еще более естественно, что ученики из одного класса хотели бы ехать в одном автобусе. Но количество желающих из 6-го “А” — 23 человека, из 8-го “В” — 15, а есть и такой класс, где желающих всего лишь трое. Нельзя ли использовать электронную таблицу для того, чтобы разместить участников экскурсии почти так, как им хочется?

Жизненность ситуации очевидна; учебная цель — освоение электронной таблицы — тоже прозрачна. Чтобы не было сомнений в том, что учащимся по силам решить эту задачу, приведем возможное обсуждение ее решения с учениками.

Прежде всего обсудим, какая информация необходима для решения задачи и как она должна быть представлена. Ясно, что для этой задачи исходная информация — это численность экскурсантов от каждого класса. Такую информацию удобнее всего иметь в виде таблицы.

Класс	Едут на экскурсию
6-й “А”	23
6-й “Б”	17
6-й “В”	14
6-й “Г”	7
7-й “А”	18
7-й “Б”	6
7-й “В”	19
7-й “Г”	22
8-й “А”	21
8-й “Б”	11
8-й “В”	15

Кроме того, надо, конечно, знать вместимость автобусов. Пусть в каждый автобус входит ровно 45 пассажиров<sup>3</sup>. Просуммировав общее число желающих поехать на экскурсию и разделив на 45, получаем, что заказать надо 4 автобуса.

Чтобы использовать электронную таблицу, надо решить, как исходная информация будет в ней представлена. Поскольку контролировать надо заполнение автобусов, естественно для каждого автобуса выделить столбец и в конце подсчитывать количество человек, размещаемых в данном автобусе. Вот заготовка таблицы:

	А	В	С	Д	Е	Ф
1	<b>Класс</b>	<b>Едут на экскурсию</b>	<b>Первый автобус</b>	<b>Второй автобус</b>	<b>Третий автобус</b>	<b>Четвертый автобус</b>
2	6-й "А"	23				
3	6-й "Б"	17				
4	6-й "В"	14				
5	6-й "Г"	7				
6	7-й "А"	18				
7	7-й "Б"	6				
8	7-й "В"	19				
9	7-й "Г"	22				
10	8-й "А"	21				
11	8-й "Б"	11				
12	8-й "В"	15				
13		<b>Итого в автобусе :</b>				

<sup>3</sup> Условность исходных данных никак не мешает учащимся воспринимать эту задачу как жизненную, все они прекрасно понимают, что не в числах дело. Так что учителю нет никакой необходимости привязывать числовые данные к конкретным условиям своей школы.

Теперь будем пытаться с помощью электронной таблицы подобрать такой вариант, чтобы было как можно меньше недовольных.

Эта задача отнюдь не простая. Ключевой момент обсуждения с учащимися — что должно стоять в ячейках таблицы на пересечении номеров автобусов и классов? Количество человек? Но если класс целиком садится в один автобус, оно и так нам известно.

Немного отвлечемся от автобусной экскурсии и сформулируем одно из самых главных правил информатики:

*Любая задача, связанная с обработкой информации, требует в первую очередь осмысления связи между исходными данными и результатами.*

Итак, что общего имеют между собой первый автобус и, скажем, 6-й "А" класс? Ну конечно же 6-й "А" класс либо едет в этом автобусе, либо нет. Поскольку мы хотим, чтобы класс ехал целиком, то в клетке на пересечении строки и столбца удобно ставить 1, если класс садится в данный автобус, и 0 в противном случае. А для контроля заполняемости автобуса необходимо добавить еще одну строку. Кроме того, поскольку все дети должны быть рас-

сажены хоть в какой-то автобус, можно в столбце последнего автобуса поручить выставлять нули и единицы самому компьютеру. Вот как будет выглядеть таблица, подготовленная к решению задачи:

	А	В	С	Д	Е	Ф
1	<b>Класс</b>	<b>Едут на экскурсию</b>	<b>Первый автобус</b>	<b>Второй автобус</b>	<b>Третий автобус</b>	<b>Четвертый автобус</b>
2	6-й "А"	23				1 – С2 – Д2 – Е2
3	6-й "Б"	17				1 – С3 – Д3 – Е3
4	6-й "В"	14				1 – С4 – Д4 – Е4
5	6-й "Г"	7				1 – С5 – Д5 – Е5
6	7-й "А"	18				1 – С6 – Д6 – Е6
7	7-й "Б"	6				1 – С7 – Д7 – Е7
8	7-й "В"	19				1 – С8 – Д8 – Е8
9	7-й "Г"	22				1 – С9 – Д9 – Е9
10	8-й "А"	21				1 – С10 – Д10 – Е10
11	8-й "Б"	11				1 – С11 – Д11 – Е11
12	8-й "В"	15				1 – С12 – Д12 – Е12
13		<b>Итого в автобусе :</b>	$C2 * B2 + C3 * B3 + C4 * B4 + C5 * B5 + C6 * B6 + C7 * B7 + C8 * B8 + C9 * B9 + C10 * B10 + C11 * B11 + C12 * B12$	$D2 * B2 + D3 * B3 + D4 * B4 + D5 * B5 + D6 * B6 + D7 * B7 + D8 * B8 + D9 * B9 + D10 * B10 + D11 * B11 + D12 * B12$	$E2 * B2 + E3 * B3 + E4 * B4 + E5 * B5 + E6 * B6 + E7 * B7 + E8 * B8 + E9 * B9 + E10 * B10 + E11 * B11 + E12 * B12$	$F2 * B2 + F3 * B3 + F4 * B4 + F5 * B5 + F6 * B6 + F7 * B7 + F8 * B8 + F9 * B9 + F10 * B10 + F11 * B11 + F12 * B12$

Решение задачи сводится теперь к расстановке в клетках таблицы нулей и единиц так, чтобы в строке “Итого” во всех автобусах было бы примерно поровну, но не более 45. Для тех данных, которые приведены, ответ может выглядеть так:

	А	В	С	Д	Е	Ф
1	<b>Класс</b>	<b>Едут на экскурсию</b>	<b>Первый автобус</b>	<b>Второй автобус</b>	<b>Третий автобус</b>	<b>Четвертый автобус</b>
2	6-й "А"	23	1	0	0	0
3	6-й "Б"	17	0	0	0	1
4	6-й "В"	14	0	0	0	1
5	6-й "Г"	7	0	0	0	1
6	7-й "А"	18	0	0	1	0
7	7-й "Б"	6	0	0	0	1
8	7-й "В"	19	0	1	0	0
9	7-й "Г"	22	1	0	0	0
10	8-й "А"	21	0	1	0	0
11	8-й "Б"	11	0	0	1	0
12	8-й "В"	15	0	0	1	0
13		<b>Итого в автобусе:</b>	45	40	44	44

Конечно, если в каждом классе число желающих оказалось не меньше 23, то целиком в автобус может войти не более одного класса. Ясно, что тогда придется все равно какие-то классы распределять между разными автобусами. Можно предложить учащимся решить данную задачу и в такой постановке: рассадить учащихся по автобусам так, чтобы каждый класс ехал либо целиком в одном автобусе, либо был разделен не более чем на две части, причем число классов, которые оказались разделенными, должно быть минимальным из возможных.

Обсуждение этой задачи преследует еще одну важную дидактическую цель. Она демонстрирует применение электронных таблиц к эвристическому, а не алгоритмическому решению задач. Попытка записать решение этой задачи посредством алгоритма требует организации полного перебора возможных вариантов размещения классов по автобусам. Здесь же, глядя каждый раз в итоговую строку, мы просто перемещаем 1 из столбца, где оказалось превышенное суммарное число 45, в тот столбец, где еще нет переполнения. Обучение эвристическим методам обработки информации и тем более с помощью компьютера — одна из важных, но трудно реализуемых задач преподавания курса информатики.

**Пример 2** ([1], с. 142; [2], с. 186; автор задачи — В.Ф. Шолохович)

В одном районе расположены несколько населенных пунктов. По территории района проходит железная дорога. По просьбе жителей района планируется построить железнодорожную станцию и проложить дороги от нее до каждого населенного пункта. Требуется определить наиболее удобное расположение железнодорожной станции.

Ситуация, описываемая в этой задаче, хотя взята и не из школьной жизни, но ее жизненность вполне понятна учащимся.

Учебные цели:

- построение компьютерной модели и эксперименты с компьютерной моделью;
- усвоение учащимися алгоритмов поиска максимального элемента в массиве;

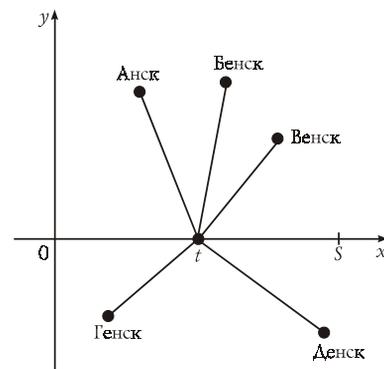
— знакомство учащихся с понятиями оптимального решения и критерия оптимизации.

Доступность учащимся мы продемонстрируем, приведя и в этом случае возможный вариант обсуждения этой задачи с учащимися.

К моменту решения этой задачи учащиеся уже хорошо усвоили, что для применения компьютера необходимо построить компьютерную модель задачи. И начинать надо с высказывания упрощающих предположений.

Прежде всего допустим, что участок дороги, проходящий по территории района, прямолинеен и в любом месте участка можно построить станцию и соединить ее прямолинейными дорогами с каждым населенным пунктом (см. рисунок). Нарисуем оси координат на карте района так, чтобы ось абсцисс проходила по

интересующему нас участку железной дороги, а начало координат совпадало с его левым концом. Населенные пункты будем изображать точками — тогда их положение просто задается координатами.



Теперь надо уяснить смысл слов “наиболее удобное расположение станции”. Это можно понимать по-разному. Если стремиться к экономии средств на строительство дорог, то станцию надо расположить так, чтобы сумма длин дорог, соединяющих станцию с населенными пунктами, была наименьшей. Если же стремиться к максимальной справедливости, то место для станции надо выбрать так, чтобы наибольшее из расстояний от нее до населенных пунктов было как можно меньше. Например, если в самом дальнем от станции населенном пункте заболел человек, то его надо доставить на станцию за самое короткое время. Мы за справедливость, поэтому выберем второй принцип.

Определим исходные данные и результат для нашей модели. Исходными данными являются количество населенных пунктов  $N$ , их координаты и длина  $S$  отрезка, изображающего на карте участок железной дороги, пролегающий по району. Результат — абсцисса точки, где будет строиться станция.

Остается найти соотношения между результатом и исходными данными. Исходные данные — координаты наших населенных пунктов — естественно организовать в форме массива  $K[N, 2]$ .

Возьмем любую точку на отрезке, изображающем железную дорогу. Пусть  $t$  — ее абсцисса. Через  $z$  обозначим максимальное из расстояний между этой точкой и каждым из населенных пунктов. Иначе говоря,

$$z = \max_{1 \leq i \leq N} \left\{ \sqrt{(t - K(i, 1))^2 + (K(i, 2))^2} \right\}.$$

Чтобы определить, где построить станцию, надо узнать, при каком  $t$  из отрезка  $[0; S]$  переменная  $z$  принимает наименьшее значение. Как же найти это  $t$ ? Ведь невозможно перебрать все числа от 0 до  $S$ : их бесконечно много.

Выход один — искать приближенное значение  $t$ . Для этого разобьем отрезок  $[0; S]$  на равные части, длину каждой из них обозначим буквой  $r$ . Затем найдем значения  $z$  при  $t = 0, r, 2r, \dots$  и так далее, пока  $t \leq S$ . Выберем из этих значений наименьшее и определим, при каком  $t$  оно достигается. Это значение  $t$  будет отличаться от искомого результата не больше, чем на  $r$ . Поэтому мы и возьмем его в качестве результата<sup>4</sup>.

Итак, для решения задачи требуется из достаточно большого количества чисел найти минимальное. Это одна из стандартных задач, и мы не будем останавливаться на организации ее решения с помощью компьютера. Здесь возможно как составление алгоритма, так и использование электронной таблицы.

Выбор другого критерия оптимальности расположения станции приведет к иному ответу для  $t$ .

Полученные при решении этой задачи результаты должны обсуждаться не только с точки зрения правильности проведенных вычислений и построений, но с содержательной точки зрения, т.е. с позиций критериев оптимальности полученного решения. Рассматриваемая нами задача обладает тем важным достоинством, что она многокритериальна: критерием оптимальности решения могут выступать и принцип минимума суммарной длины строящихся дорог, и принцип социальной справедливости, и, быть может, еще какой-либо принцип, выдвинутый учащимися в ходе обсуждения данной задачи. К примеру, таким принципом может выступать принцип минимизации эксплуатационных расходов. Ясно ведь, что из большего по численности населенного пункта будет больше людей ездить на станцию, поэтому потребуются больше автобусов. Количество этих автобусов можно считать пропорциональным числу жителей и продолжительности поездки (т.е. длине пути). Иными словами, чтобы обойтись заданным числом автобусов, нужно, чтобы населенные пункты с большей численностью были к станции ближе других.

Многокритериальные задачи (в том числе и предлагаемая нами) характеризуются тем, что, как правило, нельзя добиться наилучшего решения по всем критериям сразу. Приходится искать компромисс, состоящий в том, что решение, неоптимальное по каждому отдельно взятому критерию, все же оказывается приемлемым с точки зрения совокупности этих критериев. Один из путей поиска такого компромиссного решения описан в [1], с. 147, и [2], с. 190. Но здесь важно само обсуждение с учащимися

<sup>4</sup> Вообще говоря, не для каждой функции полученное таким методом значение  $t$  действительно будет отстоять от точки минимума менее чем на  $r$  (попробуйте привести пример такой функции). Однако можно строго доказать, что в данном случае это верно.

необходимости компромисса, причем компромисса не политического (о каковом только и можно услышать в нашей стране), а экономического, да еще и проанализированного вычислительными средствами информатики.

И снова учащиеся убеждаются в жизненной применимости, а значит, и важности изучения информатики.

**Пример 3** ([1], с. 30; автор задачи — М.В. Сапир)

Поработав летом, школьник купил себе дорогой магнитофон. К сожалению, с покупкой магнитофона затраты на него не кончаются: чем старше магнитофон, тем больше денег “съест” его ремонт. Со временем ремонт может обходиться дороже, чем замена старого магнитофона на новый магнитофон той же марки. Вот и возникла у школьника задача: через сколько лет после покупки магнитофона его наиболее выгодно обменять на новый магнитофон той же марки?

Однако “звонит звонок”, оповещающий об окончании лекции, — исчерпан объем в газете, отводимый для одной лекции, и размышления о том, какие учебные цели можно достичь, разбирая эту задачу, и как организовать обсуждение ее решения с учащимися, остаются уважаемым слушателям-читателям “на дом”.

### Вопросы и задания

1. Как вы считаете, при изучении какой темы курса информатики могла бы использоваться приведенная выше задача о магнитофоне? Каким дидактическим целям может служить ее решение?

2. В этой лекции в качестве основного подхода к созданию мотивации к изучению курса информатики рассматривалась жизненная направленность данного курса. Реализуется ли тот же подход в создании мотивации следующим фрагментом введения, взятым из некоторого действующего сегодня учебника информатики? Ответ постарайтесь аргументировать.

“Содержание учебника должно помочь вам реализовать свои творческие возможности. Если вы имеете склонность к сочинительству, то учебник предоставляет прекрасную возможность научиться красиво оформлять и иллюстрировать вашу работу в программных средах Word и PageMaker. Кто-то мечтает попробовать себя в роли режиссера, художника, сценариста, звукорежиссера. Изучив среды PowerPoint и Access, вы всегда сможете воплотить любой свой замысел, создавая видеоклип или мультимедийный продукт. Кто-то другой готовит себя в менеджеры. В этом случае вам необходимо внимательно изучить раздел, в котором рассказывается о делопроизводстве. В этой книге каждый сможет найти что-то полезное для себя”.

3. Можете ли вы предложить свой подход к созданию мотивации изучения курса информатики в целом?

### Литература

1. Гейн А.Г., Линецкий Е.В., Сапир М.В., Шолохович В.Ф. Информатика: Учебник для 8—9-х классов общеобразовательных учреждений. М.: Просвещение, 1994, 256 с.
2. Гейн А.Г., Сенокосов А.И., Шолохович В.Ф. Информатика: Учебник для 7—9-х классов общеобразовательных учебных заведений. М.: Дрофа, 1997, 224 с.
3. Информатика: Учебник для 6—7-х классов средней школы / Под ред. Н.В. Макаровой. СПб.: Питер Ком, 1998, 256 с.
4. Информатика: Учебник для 10—11-х классов средней школы / Под ред. Н.В. Макаровой. СПб.: Питер Ком, 1999, 304 с.
5. Кузнецов А.А., Апатова Н.В. Основы информатики. 8—9-е классы: Учебник для общеобразовательных учебных заведений. М.: Дрофа, 1999, 176 с.



Современный компьютер невозможно представить без основного носителя информации — жесткого диска. Это один из основных компонентов системы, вытеснить который не смогли никакие новомодные концепции, вроде популярной не так давно идеи бездисковых интернет-терминалов, использующих в качестве постоянного хранилища данных ресурсы Всемирной сети. Наоборот, в наши дни жесткие диски все чаще находят применение в устройствах, далеких от персональных компьютеров, — это и цифровые видеомонофоны, и бытовые mp3-плееры, mp3-автоматитолы и так далее. Даже в, казалось бы, такой специфической области, как цифровая фотография, где одним из важнейших факторов является мобильность, только жесткий диск (например, IBM "Microdrive") способен на сегодняшний день предоставить емкость, измеряемую сотнями мегабайт. Однако в большинстве случаев термин "жесткий диск" ассоциируется с 3,5-дюймовым устройством (см. рисунок).

Первый жесткий диск был представлен фирмой IBM в 1956 году. Он назывался RAMAC, обладал 5-мегабайтной емкостью и состоял из 50 24-дюймовых пластин. Безусловно, кроме названия и основных принципов работы, это устройство не имеет ничего общего с тем, что мы привыкли подразумевать под жестким диском. В то время еще даже не существовало названия "винчестер", которое прочно вошло в лексикон всех, кто имеет отношение к информатическим технологиям. Этот термин появился лишь в 1973 году, когда все та же IBM представила модель 3340, имевшую неофициальное название "Винчестер". Это был 60-мегабайтный жесткий диск, состоявший из двух 14-дюймовых пластин.

Много теорий и слухов ходит по поводу того, почему инженеры из IBM назвали свой первый накопитель на жестких магнитных дисках (НЖМД) именно Winchester. Общепринятая версия гласит, что изобретатель ружья Winchester построил усадьбу в Сан-Хосе, которая находится как раз недалеко от лаборатории IBM, где и был изобретен Winchester. Еще одна теория связана с лабораторией IBM близ местечка Winchester, что в Англии. А правильный ответ на вопрос "Почему винчестер?" таков. Лаборатория IBM в Сан-Хосе находится недалеко от Winchester House — в 16 км. А жесткий диск был назван так из-за того, что он содержал две пластины по 30 Мб каждая (30—30), что напомнило руководителю проекта Кену Хотону (Ken Haughton) про ружье Winchester (два ствола калибра 30). (Эта история описана в книге Magnetic Recording: the First 100 Years, авторы — Eric D. Daniel, С. Denis Mee, Mark H. Clark.)

Нужно сказать, что физический размер накопителей на ранних этапах развития компьютерной индустрии был далеко не самым важным фактором при их проектировании и производстве, так как основной упор делался на емкость и скорость. Однако со временем достаточно остро встал вопрос об уменьшении размеров жестких дисков. И вот через 6 лет после выпуска "Винчестера", в 1979 году, IBM была анонсирована модель 3310 — первый жесткий диск с 8-дюймовыми пластинами. Этот формат пришел на смену 14-дюймовому, который являлся стандартом де-факто на протяжении почти десяти лет. Практически вслед за этим событием (в 1980 году) Seagate представила жесткий диск ST-506, имевший форм-фактор 5,25 дюйма (5 мегабайт, 4 пластины). Достижение столь малых по тем временам размеров накопителей позволило использовать их в первых персональных компьютерах. Данный форм-фактор приобрел небывалую популярность и использовался в ПК на протяжении многих последующих лет. Даже появление в 1984 году первого жесткого диска с более прогрессивным размером 3,5 дюйма, ставшим впоследствии одним из важнейших стандартов всей индустрии информационных технологий, практически не повлияло в тот момент на ситуацию на рынке, где продолжали доминировать устройства 5,25 дюйма.

Компанией, представившей первый 3,5-дюймовый жесткий диск, была небольшая шотландская фирма Rodime plc. Модель называлась R0352, имела емкость 10 мегабайт и две 3,5-дюймовые пластины. Накопитель на жестком диске относился к наиболее совершенным и сложным устройствам современного персонального компьютера. Его диски способны вместить многие гигабайты информации,

передаваемой с огромной скоростью. В то время, как почти все элементы компьютера работают бесшумно, жесткий диск ворчит и поскрипывает, что позволяет отнести его к тем немногим компьютерным устройствам, которые содержат как механические, так и электронные компоненты.

Взглянув на накопитель на жестком диске, вы увидите только прочный металлический корпус. Он полностью герметичен и защищает дисковод от частичек пыли, которые при попадании в узкий зазор между головкой и поверхностью диска могут повредить чувствительный магнитный слой и вывести диск из строя. Кроме того, корпус экранирует накопитель от электромагнитных помех.

Внутри корпуса находятся все механизмы и некоторые электронные узлы.

Механизмы — это сами диски, на которых хранится информация (1), головки (4), которые записывают и считывают информацию с дисков, а также двигатели, приводящие все это в движение.

Диск представляет собой круглую металлическую или стеклянную пластину с очень ровной поверхностью, покрытую тонким ферромагнитным слоем. Во многих накопителях раньше использовался слой оксида железа (который покрывается обычной магнитной лентой), но теперь модели жестких дисков работают со слоем кобальта толщиной порядка десятка микрон. Такое покрытие более прочно и, кроме того, позволяет значительно увеличить плотность записи. Технология его нанесения близка к той, которая используется при производстве интегральных микросхем.

Количество дисков может быть различным — от двух до двенадцати, количество рабочих поверхностей соответствующим образом больше (по две на каждом диске). Последнее (как и материал, использованный для магнитного покрытия) определяет емкость жесткого диска.

Иногда наружные поверхности крайних дисков (или одного из них) не используются, что позволяет уменьшить высоту накопителя, но при этом количество рабочих поверхностей уменьшается и может оказаться нечетным.

Магнитные головки считывают и записывают информацию на диски. Принцип записи в общем схож с тем, который используется в обычном магните.

Цифровая информация преобразуется в переменный электрический ток, поступающий на магнитную головку, а затем передается на магнитный диск, но уже в виде магнитного поля, которое может воспринять и "запомнить".

Магнитное покрытие диска представляет собой множество мельчайших областей самопроизвольной (спонтанной) намагниченности. Для наглядности представьте себе, что диск покрыт слоем очень маленьких стрелок от компаса, направленных в разные стороны. Такие частицы-стрелки называются "доменами". Под воздействием внешнего магнитного поля события в соответствии с его направлением ориентируются в соответствии с его направлением. После прекращения действия внешнего поля на поверхности диска образуются зоны остаточной намагниченности. Таким образом, сохраняются записанная на диск информация. Участки остаточной намагниченности, оказавшись при вращении диска напротив зазора магнитной головки, наводят в ней электропроводящую катушку, изменяющуюся в зависимости от величины намагниченности.

Пакет дисков, смонтированный на оси-шпинделе (2), приводится в движение специальным двигателем, компактно расположенным под ним. Количество оборотов шпинделя напрямую связано с вращением носителей информации — дисков. Скорость вращения дисков, как правило, составляет от 3600 об/мин. до 7200 об/мин. Безусловно, этот параметр влияет на количество данных, считываемых в единицу времени. Сейчас на рынке доступны диски со скоростями вращения до 10 000 (пока серийно не выпускаются) оборотов в минуту, однако среди них более распространенные 5400 и 7200 оборотов в секунду.

Диски "5400" более надежные и долговечные. При работе они меньше греются. "7200", как правило, более шумные и менее долговечные, но в целом они обеспечивают 10—12%-ный прирост производительности всей системы в целом.

Для того чтобы сократить время выхода накопителя в рабочее состояние, двигатель при включении некоторое время работает в форсированном режиме. Поэтому источник питания компьютера должен иметь запас по пиковой мощности. Теперь о работе головок (4). Они перемещаются с помощью прецизионного шагового двигателя (3) и как бы "плывут" на расстоянии доли микрона от поверхности диска, не касаясь его. Это обеспечивается специальной аэродинамикой голо-

вок, которая создает эффект "воздушной подушки". Именно поэтому головки передвигаются в рабочей области дисков только после набора теми достаточной скорости вращения.

На поверхности дисков в результате записи информации образуются намагниченные участки в форме концентрических окружностей. Они называются "магнитными дорожками". Совокупность таких окружностей одинакового радиуса, полученная при фиксированном положении позиционера головок, называется "цилиндром".

Электроника жесткого диска расположена снизу винчестера. Она расширярует команды контроллера жесткого диска и передает их в виде изменяющегося напряжения на шаговый двигатель, перемещающий магнитные головки к нужному цилиндру диска. Кроме того, она управляет приводом шпинделя, стабилизируя скорость вращения пакета дисков, генерирует сигналы для головок при записи, усиливает эти сигналы при чтении и управляет работой других электронных узлов накопителя. Практически все современные жесткие диски имеют еще и дополнительную кэш-память, позволяющую многократно ускорить повторный доступ к информации.

Жесткие диски имеют различный объем встроеного кэша. Он заметно влияет на скорость работы с данными, особенно при работе с маленькими файлами баз данных, когда обновление информации носит "локальный характер" и происходит достаточно часто (например, работа по вводу бухгалтерской информации). Размер кэша для современных дисков составляет от полумегабайта до двух мегабайт.

Естественно, что такие сложные функции невозможно реализовать без специализированного процессора. Фактически "винчестер" превратился в специализированный компьютер, обеспечивающий чтение-запись информации с магнитных дисков.

Несколько слов об интерфейсе современных винчестеров. В настоящий момент наиболее распространены два из них.

#### Ultra ATA (ATA-33, ATA-66, ATA-100)

Это самый последний вариант IDE-интерфейса. Жесткие диски характеризуются относительной дешевизной и идеально подходят для большинства домашних и офисных компьютеров. Указанные в конце цифры говорят о максимальной скорости передачи данных по шине (Мбайт/с) от контроллера диска к материнской плате. Работает, реально скорость передачи данных может быть существенно ниже. Это зависит от скорости работы диска, скорости работы электроники, работы памяти и процессора.

#### SCSI

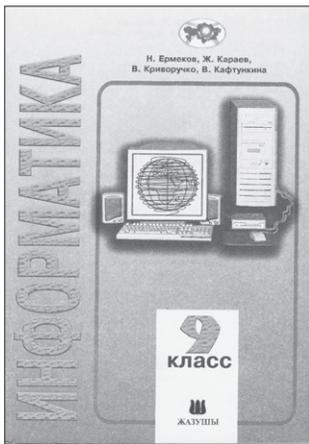
Этот интерфейс предназначен не только для использования жестких дисков. Контроллер, который вставляется в специальный слот материнской платы, может поддерживать до 15 различных устройств (сканеры, CD-ROM, жесткие диски и прочее). Производители SCSI-винчестеров разработали устройства со скоростью вращения дисков 15 000 оборотов в минуту. Скоростные характеристики таких жестких дисков с легкостью будут самые шустрые IDE-приводы. Основным преимуществом SCSI можно считать их малую нагрузку на процессор и более высокую скорость работы. К сожалению, как контроллер, так и сами жесткие диски достаточно дороги.

#### Источники информации

1. Андрей Чапка ([www.km.ru](http://www.km.ru)).
2. Михаил Заруцкий, Андрей Рябцев ([www.ixbt.com](http://www.ixbt.com)).







# Моделирование

Н. Турлынович,  
Казахстан

В этом и следующем номерах мы предлагаем вниманию наших читателей главу из нового учебника по информатике, который вышел в Казахстане. При подготовке материала к печати в него были внесены минимальные изменения, вызванные необходимостью переворота из книжного формата в газетный.

## Понятие и виды моделей

Компьютеры являются прекрасным средством для научных исследований — моделирования процессов, выполнения всевозможных экспериментов.

Что же такое модель? Что общего между игрушечным корабликом и рисунком на экране компьютера, изображающим строгую математическую абстракцию?

И все же общее есть: и в том, и в другом случае мы сталкиваемся с образом реального объекта или явления, “заместителем оригинала”, воспроизведенного с той или иной достоверностью или подробностью. Другими словами, модель является представлением объекта в форме, отличной от реальной.

Практически во всех науках о природе и обществе построение и использование моделей является мощным оружием познания. Реальные объекты и процессы бывают столь многогранны и сложны, что лучшим способом их изучения часто является построение и исследование модели, отражающей лишь какую-то грань реальности, и потому более простую, чем эта реальность. Многовековой опыт развития науки доказал на практике плодотворность такого подхода.

**Модель** — это представление объекта, процесса реального или вымышленного мира, которое отражает существенные стороны его существования.

Модель нужна, чтобы дать представление об объекте (процессе), какова его структура, основные свойства, законы развития и взаимодействия с окружающим миром, определить наилучшие способы управления и прогнозировать поведение. Один и тот же объект может иметь несколько моделей, разные объекты могут описываться одной моделью. Это объясняется тем, что выбор существенных признаков зависит от целей исследования.

Различают следующие основные виды моделей:

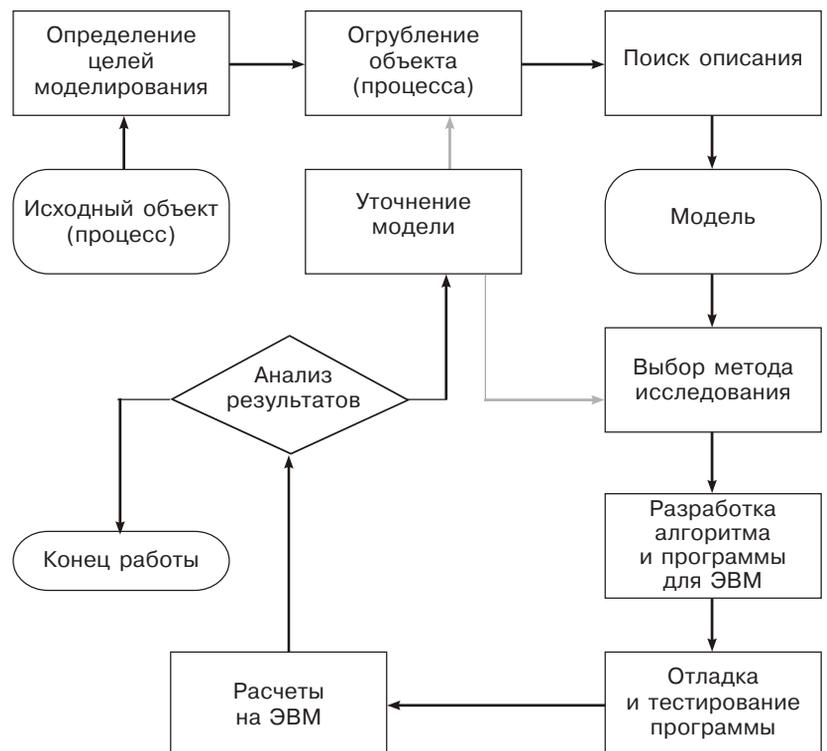
- 1. Графическое представление.** Примером могут служить фотографии, рисунки, схемы, чертежи.
- 2. Словесное описание.** Это словесное перечисление основных составных частей объекта, наиболее важных признаков и свойств. Подобной моделью

является, например, описание путешественником какой-либо местности.

- 3. Информационно-логические модели.** Набор величин, содержащий всю необходимую информацию об исследуемом объекте (процессе), называется информационной моделью. Примером может служить таблица — универсальное средство для представления информации, блок-схема.
- 4. Математическая модель.** Подобные модели строятся на основе уравнений и функций, выражающих существенные черты объекта (процесса). На основе математического и информационно-логического моделирования выполняется компьютерное моделирование.
- 5. Натурная модель.** К ним относятся технические модели, например, модель автомобиля, самолета.

## Этапы компьютерного моделирования

Основные этапы компьютерного моделирования показаны на рисунке.



Этапы компьютерного моделирования

### Первый этап — определение целей моделирования

Основные из них:

- понять, как устроен конкретный объект, каковы его структура, основные свойства, законы развития и взаимодействия с окружающим миром;
- при необходимости — научиться управлять объектом (или процессом) и определить наилучшие способы управления при заданных целях и критериях;
- уметь прогнозировать прямые и косвенные последствия воздействия на объект.

### Второй этап — огрубление исходного процесса (объекта)

На этом этапе происходит отделение информации от реального объекта, фиксируется существенная информация, отбрасывается несущественная, составляется список величин, от которых зависит поведение объекта или ход процесса (*входные величины*), а также тех величин, которые желательно получить в результате моделирования (*выходные величины*). Важно отметить, что информация, несущественная для одной задачи, может оказаться существенной для другой и потеря ее приведет к неверному решению или не позволит получить нужное решение. Учет несущественной информации вызывает неоправданные сложности, создает препятствия на пути к решению. Очень важно разделить (*ранжировать*) входные параметры по степени важности влияния их изменения на выходные.

### Третий этап — поиск описания

На этом этапе информация о моделируемом объекте структурируется и подготавливается к обработке на компьютере. Необходимо перейти от абстрактной описательной формулировки модели к формулировке, имеющей конкретное наполнение. Когда модель сформулирована, выбирается метод ее исследования.

### Четвертый этап — разработка алгоритма и составление программы для ЭВМ

Это творческий и трудноформализуемый процесс. В настоящее время наиболее распространены приемы структурного и объектно-ориентированного программирования. Выбор языка программирования обычно определяется опытом программиста, наличием стандартных подпрограмм и доступных библиотек. В некоторых случаях расчеты удобно провести, используя готовые программные продукты, например, электронные таблицы или специальные математические пакеты.

После составления программы решаем с ее помощью простейшую тестовую задачу (желательно с заранее известным ответом) с целью устранения грубых ошибок.

Модель адекватна реальному процессу, если некоторые характеристики процесса, полученные на ЭВМ, с заданной степенью точности совпадают с экспериментальными. В случае несоответствия модели реальному процессу возвращаемся к одному из предыдущих этапов. Возможные точки возврата указаны на

схеме: либо в процессе огрубления были отброшены какие-то важные факторы, либо выбор метода исследования оказался не слишком удачным и нужно использовать более сложный и точный. После внесения изменений вновь проходим по части технологической цепочки и повторяем это, пока не получим приемлемые результаты.

Если результаты соответствуют экспериментальным данным или нашим интуитивным представлениям, проводим расчеты по программе. Результаты накапливаются и затем обрабатываются.

**Пример.** Имеется классный журнал. Для проведения контроля и качественного анализа за успеваемостью разработайте для классного руководителя модель отчета.

*Простейшая модель отчета.*

**I этап.** Определение целей моделирования.

В самом общем приближении можно выделить следующие цели:

- хранение информации;
- возможность упорядочивания информации;
- предоставление информации в удобном для пользователя виде;
- возможность выбора данных по некоторым критериям;
- возможность изменения информации пользователем в процессе функционирования модели.

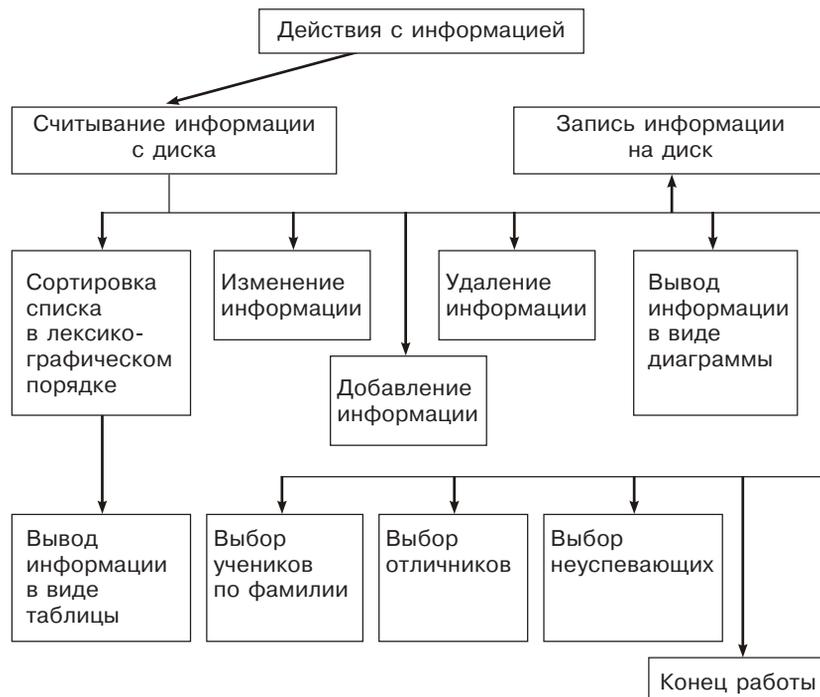
**II этап.** Выделим из разнообразной информации, характеризующей объект, ту, которая обусловлена целями моделирования. Такой информацией будем считать список учащихся, их количество, итоговые оценки (четвертные или годовые, как того требует директор) по четырем профилирующим предметам, например, по физике, математике, информатике, родному языку.

**III этап.** Для работы с информационной моделью лучше всего воспользоваться таблицами, хранить информацию на диске, предоставлять информацию пользователю в виде таблиц и диаграмм для качественного анализа по предметам (доля “4” и “5” по предмету в зависимости от их общего количества).

Представим простейшую модель в виде схемы, представленной на с. 20.

Подобную модель можно реализовать с помощью СУБД, электронной таблицы и языка программирования.

**IV этап.** Выберем язык программирования QBasic. В случае выключения и включения компьютера приходилось каждый раз вводить исходные данные. В QBasic есть возможность хранить эти данные на диске в файле. Для этого перед началом работы файл для данных необходимо открыть, а затем записывать или считывать данные с диска. Далее приводятся описания некоторых команд языка QBasic для операций с файлами, а также некоторых других полезных команд, которые нам понадобятся.



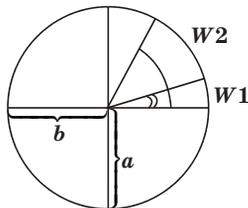
**OPEN** OPEN имя файла FOR режим AS # N  
 OPEN имя файла FOR INPUT AS # N  
 OPEN имя файла FOR OUTPUT AS # N

**INPUT** INPUT # N, список переменных  
**PRINT** PRINT # N, список переменных  
**CLOSE** CLOSE # N  
**EOF** EOF (N)

Пример: WHILE NOT EOF (N)  
 операторы  
 WEND

**INKEY\$** X\$=INKEY\$  
 Пример: WHILE INKEY\$=""  
 WEND

**CIRCLE** CIRCLE (X, Y), R, C, W1, W2, K  
 CIRCLE (X, Y), R, C, -W1, -W2, K



Открыть файл данных:  
 для чтения,  
 для записи,  
 $1 \leq N \leq 255$  — логический номер файла.  
 Чтение данных из файла.  
 Вывод в файл.  
 Закрывать файл.  
 Проверка конца файла.  
 Следит за тем, когда данные в файле закончатся.

Ввод символа с клавиатуры.  
 При вводе не отображается знак "?".

Рисование дуги эллипса.  
 Рисование сектора эллипса.  
 $W1, W2$  — начальный и конечный углы дуги в радианах,  
 $K = b/a$  — коэффициент сжатия эллипса.

С помощью программы **ЗАПИСЬ НА ДИСК** запишем следующую информацию на диск:

Фамилия, имя	Оценки			
	Физика	Математика	Информатика	Родной язык
Зайцев Толя	5	4	3	2
Каиров Кайрат	5	5	5	5
Юн Лена	4	5	5	4

```

REM ЗАПИСЬ НА ДИСК
OPEN "KCLASS" FOR OUTPUT AS #1
INPUT "КОЛИЧЕСТВО УЧЕНИКОВ:"; N
FOR J = 1 TO N
INPUT "СВЕДЕНИЯ ОБ УЧЕНИКЕ (ФАМИЛИЯ, ОЦЕНКИ ПО ФИЗИКЕ, МАТЕМАТИКЕ, ИНФОРМАТИКЕ,
РОДНОМУ ЯЗЫКУ):"; A1$, F1, M1, I1, R1
PRINT #1, A1$
  
```

```

PRINT #1, F1
PRINT #1, M1
PRINT #1, I1
PRINT #1, R1
NEXT J
CLOSE #1

```

Вышеописанная программа записала введенные с клавиатуры данные в файл с именем "KLASS" на диск. Эта программа для дальнейшей работы не нужна и может понадобиться только в том случае, если кто-то уничтожит файл "KLASS".

Для работы с данными на диске необходимо составить новую программу. Программа **БАЗА ДАННЫХ** рассчитана на список из 20 учеников ( $N = 20$ ). При желании значение этой величины можно изменить.

Программа **БАЗА ДАННЫХ** представляет собой компьютерную модель:

```

DECLARE SUB sap(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB sort(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB writ(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB bukw(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB ot1(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB DWA(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB ISM(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB DOB(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB UD(a$( ), f!( ), m!( ), i!( ), r!( ), n!)
DECLARE SUB diagr(a$( ), f!( ), m!( ), i!( ), r!( ), n!)

```

Организует запись данных на диск (режим "8").  
 Сортировка списка в лексикографическом порядке.  
 Вывод информации (режим "1").  
 Выбор учеников по фамилии (режим "2").  
 Выбор отличников (режим "3").  
 Выбор неуспевающих (режим "4").  
 Изменение информации (режим "5").  
 Добавление информации (режим "6").  
 Удаление информации (режим "7").  
 Построение диаграммы (режим "9").

```

REM БАЗА ДАННЫХ
n = 20
DIM a$(n), f(n), m(n), i(n), r(n)
OPEN "KLASS" FOR INPUT AS #1
n = 0
WHILE NOT EOF(1)
n = n + 1: INPUT #1, a$(n)
INPUT #1, f(n)
INPUT #1, m(n)
INPUT #1, i(n)
INPUT #1, r(n)
WEND
CLOSE #1
CLS : LOCATE 2, 30: PRINT "ЗАДАЙТЕ РЕЖИМ!"
10 x$ = INKEY$
IF x$ = "" THEN 10
CLS : LOCATE 2, 30: PRINT "ЗАДАЙТЕ РЕЖИМ!"
SELECT CASE x$
CASE "1"
CALL writ(a$( ), f( ), m( ), i( ), r( ), n)
CASE "2"
CALL bukw(a$( ), f( ), m( ), i( ), r( ), n)
CASE "3"
CALL ot1(a$( ), f( ), m( ), i( ), r( ), n)
CASE "4"
CALL DWA(a$( ), f( ), m( ), i( ), r( ), n)
CASE "5"
CALL ISM(a$( ), f( ), m( ), i( ), r( ), n)
CASE "6"
CALL DOB(a$( ), f( ), m( ), i( ), r( ), n)
CASE "7"
o CALL UD(a$( ), f( ), m( ), i( ), r( ), n)
CASE "8"
CALL sap(a$( ), f( ), m( ), i( ), r( ), n)
CASE "9"
CALL diagr(a$( ), f( ), m( ), i( ), r( ), n)
CASE "0"
END
END SELECT
GOTO 10

```

Вначале необходимо считать информацию с диска в массивы:

a\$ — фамилия;  
 f — оценка по физике;  
 m — оценка по математике;  
 i — оценка по информатике;  
 r — оценка по родному языку.

Печатать данные.

Организовывать поиск по первой букве фамилии.

Выбирать из списка информацию об отличниках.

Выбирать из списка информацию о неуспевающих.

Изменять сведения о каком-либо ученике.

Добавлять сведения о новом ученике в список.

Удалять сведения о каком-либо ученике.

Записывать новые данные на диск.

Строить диаграмму.

```
SUB bukw (a$( ), f( ), m( ), i( ), r( ), n)           Поиск ведется по первой букве фамилии.
```

```
INPUT "ВВЕДИТЕ БУКВУ:"; V$
FOR j = 1 TO n
IF LEFT$(a$(j), 1) = V$ THEN PRINT a$(j), f(j), m(j), i(j), r(j)
NEXT j
END SUB
```

```
SUB diagr (a$( ), f( ), m( ), i( ), r( ), n)
DIM b(4)
SCREEN 9
CLS
LOCATE 2, 20: PRINT "КАЧЕСТВЕННЫЙ АНАЛИЗ УСПЕВАЕМОСТИ"
CIRCLE (300, 150), 100, 15, , , 1: pi = 3.14159
kf = 0: km = 0: ki = 0: kr = 0
```

Определение количества оценок "4" и "5" по каждому предмету.

```
FOR j = 1 TO n
IF f(j) = 4 OR f(j) = 5 THEN kf = kf + 1
IF m(j) = 4 OR m(j) = 5 THEN km = km + 1
IF i(j) = 4 OR i(j) = 5 THEN ki = ki + 1
IF r(j) = 4 OR r(j) = 5 THEN kr = kr + 1
NEXT j
s = kf + km + ki + kr: c = 0: w1 = 0
b(1) = kf / s: b(2) = km / s
b(3) = ki / s: b(4) = kr / s
```

В s хранится количество всех оценок "4" и "5". Доля оценок "4" и "5" по предмету от общего количества.

```
FOR j = 1 TO 4
c = c + b(j): w2 = c * 2 * pi
CIRCLE (300, 150), 100, 15, -w1, -w2, 1
x = 98 * COS(w2 - .05)
y = 98 * SIN(w2 - .05)
PAINT (x + 300, -y + 150), j + 1, 15: w1 = w2
NEXT
```

Рисование секторов и их закрашка.

```
LOCATE 23, 4: PRINT "    -ФИЗИКА          -МАТЕМАТИКА      -ИНФОРМАТИКА      -РОДНОЙ ЯЗЫК"
LINE (35, 310)-(45, 320), 2, BF
LINE (145, 310)-(155, 320), 3, BF
LINE (265, 310)-(275, 320), 4, BF
LINE (390, 310)-(400, 320), 5, BF
END SUB
```

Оформление диаграммы.

```
SUB DOB (a$( ), f( ), m( ), i( ), r( ), n)           Добавление сведений о новом ученике.
```

```
n = n + 1
INPUT "ВВЕДИТЕ НОВЫЕ ДАННЫЕ:"; a$(n), f(n), m(n), i(n), r(n)
END SUB
```

```
SUB DWA (a$( ), f( ), m( ), i( ), r( ), n)           Выбор неуспевающих учеников.
```

```
FOR j = 1 TO n
IF f(j) = 2 OR m(j) = 2 OR i(j) = 2 OR
r(j) = 2 THEN PRINT a$(j), f(j), m(j), i(j), r(j)
NEXT j
END SUB
```

```
SUB ISM (a$( ), f( ), m( ), i( ), r( ), n)           Изменение сведений об ученике.
```

```
CALL writ(a$( ), f( ), m( ), i( ), r( ), n)
INPUT "ВВЕДИТЕ НОМЕР:"; N1
INPUT "ВВЕДИТЕ НОВЫЕ ДАННЫЕ:"; a$(N1), f(N1), m(N1), i(N1), r(N1)
END SUB
```

```
SUB ot1 (a$( ), f( ), m( ), i( ), r( ), n)           Если сумма оценок равна 20, это - отличник!
```

```
FOR j = 1 TO n
IF f(j) + m(j) + i(j) + r(j) = 20 THEN PRINT a$(j), f(j), m(j), i(j), r(j)
NEXT j
END SUB
```

```
SUB sap (a$( ), f( ), m( ), i( ), r( ), n)           Запись новых данных в файл "KLASS" из массивов
a$, f, m, i, r.
```

```
OPEN "KLASS" FOR OUTPUT AS #1
FOR j = 1 TO n
PRINT #1, a$(j)
PRINT #1, f(j)
PRINT #1, m(j)
PRINT #1, i(j)
```

```
PRINT #1, r(j)
NEXT j
CLOSE #1
END SUB
```

```
SUB sort (a$( ), f( ), m( ), i( ), r( ), n)
FOR c = 1 TO n - 1
FOR k = 1 TO n - c
```

При сортировке литерных величин их значения располагаются в лексикографическом порядке.

```
IF a$(k) > a$(k + 1) THEN SWAP a$(k), a$(k + 1): SWAP f(k), f(k + 1): SWAP m(k), m(k + 1):
SWAP i(k), i(k + 1): SWAP r(k), r(k + 1)
NEXT k: NEXT c
END SUB
```

```
SUB UD (a$( ), f( ), m( ), i( ), r( ), n)
CALL writ(a$( ), f( ), m( ), i( ), r( ), n)
INPUT "ВВЕДИТЕ НОМЕР:"; N1
a$(N1) = "": f(N1) = 0: m(N1) = 0
i(N1) = 0: r(N1) = 0
CALL sort(a$( ), f( ), m( ), i( ), r( ), n)
FOR j = 1 TO n - 1
a$(j) = a$(j + 1): f(j) = f(j + 1)
m(j) = m(j + 1): i(j) = i(j + 1)
r(j) = r(j + 1)
NEXT j
n = n - 1
END SUB
```

Из списка выбирается номер позиции ученика, все числовые величины обнуляются, а литерная величина а\$ принимает значение пустого текста.

После сортировки эти сведения оказываются в начале списка. Все значения элементов массивов сдвигаются влево на один элемент, уничтожая ненужные сведения.

На одного ученика в списке становится меньше.

```
SUB writ (a$( ), f( ), m( ), i( ), r( ), n)
CALL sort(a$( ), f( ), m( ), i( ), r( ), n)
FOR k = 1 TO n
PRINT k; " "; a$(k), f(k), m(k), i(k), r(k)
NEXT k
END SUB
```

Печать сведений в виде таблицы с предварительной сортировкой.

Самостоятельно протестируйте программу, проведите анализ экспериментов.

Протокол исполнения для процедуры SUB diagr:

Пусть  $kf = 3, km = 3, ki = 2, kr = 2, w1 = 0,$   
тогда  $s = 10, b(1) = 3/10, b(2) = 3/10, b(3) = 2/10, b(4) = 2/10.$

*Продолжение следует*

## Калейдоскоп

### Лучше больше, да лучше

Компания Fujitsu разработала новый магнитный слой, благодаря которому трехдюймовые дискеты смогут хранить до трех гигабайт информации.

Новая технология называется Nano3, некоторые промежуточные результаты, полученные исследователями в процессе ее создания, уже нашли применение в ZIP-дискетах объемом 250 Мб и ленточных носителях.

Фактически новая технология Nano3 состоит из трех различных решений, разработанных компанией ранее: это двухкомпонентный ферромагнитный сплав и две технологии равномерного нанесения частиц сплава на несущую основу.

Теперь осталось дело за малым: начать продвигать на рынке оборудование, способное писать и считывать информацию с таких носителей.

*По материалам сайта [www.avanova.com](http://www.avanova.com)*

### Голосовать через Интернет реально

Таким способом, возможно, будут проходить следующие выборы в Великобритании, ожидающиеся через три года. Об этом заявил лидер палаты общин Робин Кук. Было объявлено о проведенных во время последних выборов испытаниях, во время которых при помощи Интернета проголосовало 120 тысяч студентов Лондонского университета.

Испытания признаны успешными. Теперь все дело за специальной комиссией, она должна представить результаты своей деятельности. После встанет вопрос о разрешении онлайн-голосования.

Основное, что беспокоит англичан, — возможность изменения результатов при помощи всевозможных высокотехнологичных махинаций, так что в конечном счете все упирается в создание надежной защиты передачи данных при проведении голосования.

*По материалам сайта [www.computeruser.com](http://www.computeruser.com)*

### Статистика

Примерно каждый десятый россиянин (9%) и четверть москвичей (26%) ежедневно или несколько раз в неделю пользуются дома, на работе или в других местах персональным компьютером. Примерно раз в неделю используют компьютер 4% россиян и 3% москвичей. По одному проценту как россиян в целом, так и жителей столицы работают на компьютере примерно раз в месяц. Тех, кто использует компьютер 2—3 раза в месяц, среди россиян москвичей оказалось также равное число — 1%. Столько же москвичей и жителей России пользуются компьютером менее одного раза в месяц. Никогда не пользуются компьютером 84% россиян и 69% москвичей. Об этом свидетельствуют данные, которые предоставили во Всероссийском центре изучения общественного мнения. Опрос также показал, что сейчас 23% семей москвичей и 6% семей россиян имеют дома персональные компьютеры.

*По материалам газеты "Метро"*

# Азы информатики

А.А. ДУВАНОВ

МАТЕРИАЛЫ  
РОБОТЛАНДСКОГО  
УНИВЕРСИТЕТА

Книга 1. См. № 1, 2/2002  
Книга 2. См. № 5, 6, 7/2002

Демо-версию гипертекстовой книги (700 Кб) можно скопировать с адреса:  
<ftp://ftp.botik.ru/rented/robot/univer/azinfd.zip>

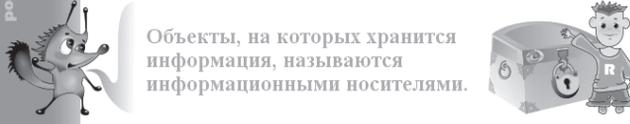
## Книга 2. В мире информации (продолжение)

*Книга для ученика*

### 4. ХРАНЕНИЕ ИНФОРМАЦИИ

**4 Хранение информации**  
Цифровой зал Роботландии  
РОБОТЛАНДИЯ.RU © А.А.ДУВАНОВ

Объекты, на которых хранится информация, называются информационными носителями.



#### Читальный зал

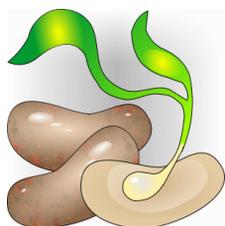
#### Где хранится информация

#### Информация хранится в природе

Посмотрев вокруг, мы увидим очень много примеров хранения информации.

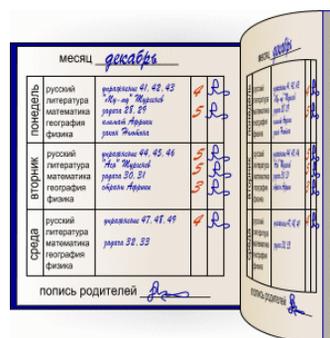
Если спилить дерево, то по кольцам на стволе можно определить, сколько ему лет, дождливым или засушливым был каждый год его жизни, и многое другое. Значит, дерево хранит информацию обо всей своей жизни.

Если посадить желудь, вырастет дубок, а из пшеничного зерна всегда вырастает пшеничный колос. Значит, в семенах хранится вся информация о растениях.



#### Информацию сохраняет человек

Давным-давно, тысячи лет назад, когда на земле жили первобытные люди, возникла необходимость хранить сведения о способах охоты, рыбной ловли, земледелия. Для этого люди использовали наскальные рисунки, зарубки на палках или узелки на веревках. По рисункам древних людей мы узнаем, как они жили.



Заучивая правила математики или стихи, человек хранит их в своей памяти. Часто, не полагаясь на память, он записывает информацию на листок бумаги, в записную книжку или тетрадь. Например, расписание уроков и домашнее задание школьник хранит в дневнике.

Любимую песенку о Черепашке и Лявкенке можно записать на ленту магнитофона. Песня — это тоже информация, и ее сохраняют, чтобы послушать, когда очень захочется. Человек придумал много способов хранения информации. Это книги, газеты, журналы. Из них люди узнают много ин-



тересных сведений об окружающем мире. Информация хранится на кинолентах, магнитных лентах, лазерных дисках.

### Информационные носители

Объекты, на которых хранится информация, называются **информационными носителями**.

Всем знаком такой информационный носитель, как книга. Это самое важное изобретение человека. Без книг нельзя передать знания большому числу людей.



Дверной косяк, на котором родители ежегодно отмечают, насколько вырос их ребенок, — это тоже информационный носитель.



Информация для компьютера записывается на магнитной дискете или лазерном диске.

Информация для компьютера записывается

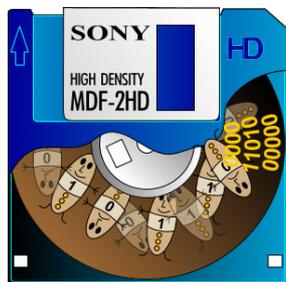


### Как хранится информация



Для записи информации люди изобретают разные способы. Так, для записи человеческой речи был изобретен алфавит. В русском языке алфавит содержит 33 буквы и несколько знаков препинания.

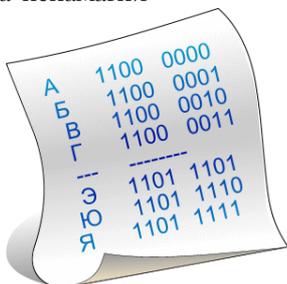
Музыканты записывают информацию о музыкальном произведении специальными значками — нотами.



Для записи информации на магнитный диск используют магнитные свойства материала, из которого сделан диск: намагниченный участок поверхности диска означает число 1, а ненамагниченный — число 0.

Комбинируя нули и единицы, можно обозначить любой знак, значит, на диске можно записать и сохранить любую информацию.

Современные информационные носители — лазерные диски. Они значительно пре-



восходят магнитные дискеты по объему: на одном лазерном диске можно поместить информацию, записанную на сотнях обычных дискет, тогда как по размерам они одинаковы. Лазерный диск намного дольше сохраняет информацию.



Информация на лазерный диск записывается в виде микроскопических углублений на его поверхности. Яркость отражения лазерного луча уменьшается, когда он попадает в углубления. Таким образом, удается хранить и считывать информацию: углубление соответствует числу 0, а отсутствие углубления — числу 1.

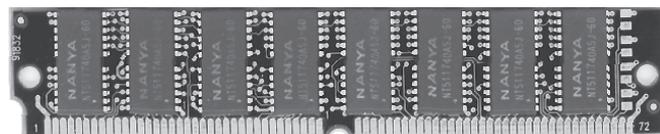
Так как чтение информации с лазерного диска происходит при помощи света (лазерным лучом), их часто называют «оптическими дисками».

### Хранение информации в компьютере

Компьютер — это прибор, который специально создан для работы с информацией.



Для временного (оперативного) хранения компьютер использует модули памяти ОЗУ — Оперативного Запоминающего Устройства. Информация из этой памяти пропадает, когда отключается электрическое питание.



ОЗУ расположено внутри системного блока. Это самая быстрая память компьютера. Компьютер может записать в ОЗУ информацию, а потом прочесть ее быстрее, чем при работе с другими информационными носителями.

Кроме ОЗУ, информация хранится в компьютере на магнитных дисках: съемных и постоянных.

Съемные магнитные диски (дискеты, или гибкие диски) вставляют в компьютер через специальную щель системного блока.



Постоянный магнитный диск компьютера (чаще его называют **винчестер**) скрыт внутри корпуса системного блока.

Винчестер является основным хранилищем информации на компьютере. На нем

хранятся все программы, которые постоянно нужны для работы. Кроме программ, винчестер содержит документы: тексты, рисунки, звук, музыку, видео — все, что нужно пользователю.

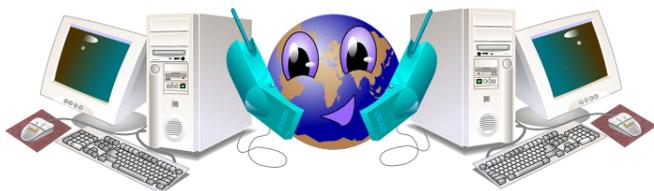
Магнитные дискеты, а также лазерные диски обычно используют для переноса информации от одного компьютера к другому. Правда, теперь для этой цели все чаще используют всемирную компьютерную сеть Интернет или локальные компьютерные сети, объединяющие несколько компьютеров, например, в школьном классе.

## Хранение информации в Интернете



**Интернет** — это объединение компьютеров по всему миру в единую информационную сеть. По-другому Интернет называют **мировой компьютерной сетью**.

Для соединения компьютеров можно использовать обычные телефонные линии и прибор **модем**. Модем преобразует информацию к виду, пригодному для передачи по телефону.



Таким образом, информация, хранящаяся по всему миру, становится доступной каждому, кто имеет компьютер и модем.

В Интернете можно найти ответ практически на любой вопрос. Прочитать свежую газету, заглянуть в библиотеку, заказать билеты на самолет, купить товары, завести друзей по переписке.



### Конспект

Информация хранится в предметах окружающего нас мира.

Человек изучает информацию, которую дает ему природа, и придумывает способы хранения информации, созданной им в процессе деятельности.

Объекты, на которых хранится информация, называются **информационными носителями**.

Самым важным информационным носителем, придуманным человеком, является книга.

Компьютер может хранить, передавать и обрабатывать информацию.

Информация в компьютере хранится в **ОЗУ (Оперативное Запоминающее Устройство)**, на магнитных и лазерных дисках.

В современном обществе возрастает значение сети Интернет, как средства доступа к информации, хранящейся на компьютерах по всему миру.

**Интернет** — это объединение компьютеров по всему миру в единую информационную сеть. По-другому Интернет называют **мировой компьютерной сетью**.



### Вопросы

1. Хранят ли информацию предметы вокруг нас? Например, хранит ли информация стол, на котором стоит компьютер?

2. Почему из семечка кабачка не вырастает елка?

3. Картошка хранится в подвале. Хранится ли вместе с картошкой информация?

4. Как люди могут узнать о жизни своих предков, живших много лет назад?

5. Что такое информационный носитель?

6. Информационные носители существуют в природе. Зачем человек придумывает новые информационные носители?

7. Какой самый важный информационный носитель придумал человек?

8. Учитель записал на доске условия контрольной работы; ученики выполняют эту работу. Где и как хранится информация об условиях задачи в ходе выполнения контрольной работы?

9. Вася сказал другу, что запомнил номер его телефона. Где Вася хранит эту информацию?

10. Егор решил записать песню, которая ему понравилась. Какой информационный носитель он может использовать?

11. В следующих примерах укажите информационный носитель и тип информации, которая хранится на носителе:

- письмо от друга;
- магнитная лента с песнями;
- газета;
- кинофильм;
- видеофильм;
- табличка с номером дома;
- билет на самолет;
- компьютерная игра.

12. Как книги хранят человеческую речь?

13. Как сохраняют информацию о музыкальном произведении?

14. Как хранится информация на фотопленке?

15. Как хранят изображения? Назовите информационные носители.

16. Старинные бухгалтерские счета хранят информацию или обрабатывают ее?

17. Какие устройства использует компьютер для хранения информации?

18. Объясните принципы работы магнитного диска.

19. Объясните принципы работы лазерного диска.

20. Где хранится информация в Интернете?

21. Хранятся ли в Интернете товары, например, обычные бумажные книги?



### Задания на дом

Найдите информацию и подготовьте сообщение по одной из следующих тем:

- История письменности.
- Изобретение бумаги.

- Из чего и как делают бумагу.
- Изобретения книгопечатания.
- Как работает печатный станок.
- Как работает магнитофон.
- Как работает фотоаппарат.
- Как изготавливают лазерные диски.



## Практикум

### Редактор строки. Латинские буквы

Привычные буквы русского языка понятны не только русским людям.

Похожий алфавит имеют некоторые близкие к русскому славянские языки — украинский, белорусский, болгарский.

Этот алфавит называют кириллицей в честь одного из двух братьев — великих болгарских просветителей Кирилла и Мефодия. Более тысячи лет тому назад они изобрели славянскую письменность.

Во многих других странах используется иной алфавит — латинский. Это алфавит латинского языка, на котором говорили в Древней Италии. От него произошли многие европейские языки. Алфавит латинского языка лежит в основе таких широко распространенных языков, как французский, английский, немецкий, испанский, итальянский, и многих других.

В латинском алфавите 26 букв:

Aa, Bb, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk, Ll, Mm,  
Nn, Oo, Pp, Qq, Rr, Ss, Tt, Uu, Vv, Ww, Xx, Yy, Zz.

Как видите, многие латинские буквы совсем не похожи на буквы кириллицы — алфавита русского языка.

Если вы хотите дружить с детьми всего мира, вы не сможете обойтись без знания иностранных языков, использующих латинский алфавит. Посмотрите, как записывается слово **мир** на некоторых языках:

- на французском: raix (по-русски читается “пэ”);
- на английском: pease (“пис”);
- на немецком: frieden (“фриден”);
- на польском: pokoj (“покой”);
- на чешском: mir (“мир”).

Но даже людям, не знающим английского, французского или немецкого языков, часто приходится иметь дело с иностранными словами и буквами латинского алфавита.

Врач, выписывая лекарство для больного, обязательно напишет название лекарства латинскими буквами. Ботаник, собирающий коллекцию растений, пишет латинское имя растения около каждого экспоната своей коллекции.

Латинскими буквами шахматисты всего мира описывают ходы в шахматной игре.

Программист записывает алгоритмы на языках программирования, основу которых составляют, как правило, слова английского языка и их сокращения.

Поэтому-то на клавиатуре компьютера рядом с кириллицей есть буквы латинского алфавита.

Чтобы компьютер воспринимал нажатия клавиш как ввод латинских букв, надо установить для клавиатуры латинский алфавит.

Какой из двух алфавитов (кириллица или латинский) включен в текущий момент времени, легко определить, взглянув на алфавитный индикатор “Панели задач”:



Включена кириллица



Включен латинский алфавит

Для переключения алфавита нужно щелкнуть по алфавитному индикатору и в появившемся меню выбрать нужную строку:



Алфавит можно переключать и с помощью клавиатурных аккордов.

Аккорд — это одновременное нажатие нескольких клавиш. Алгоритм исполнения аккорда на клавиатуре приводится ниже:

1. Нажать первую клавишу аккорда.
2. Не отпуская первую клавишу, нажать вторую.
3. Отпустить обе клавиши.

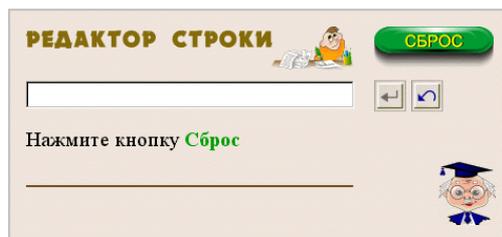
Аккорд обычно записывают, используя “+”, например, аккорд **[Shift] + [R]** приводит к вводу заглавной “R”.

Переключение алфавита обычно выполняется аккордом **[Ctrl] + [Shift]** или аккордом **[Alt]** слева + **[Shift]**.

Вам надо проверить, какой аккорд действует на вашем компьютере.

### Вопросы

1. Какие языки используют латинский алфавит? Назовите 3—4 примера.
  2. Можете ли вы назвать пример языка, алфавит которого отличается и от латинского, и от кириллицы?
  3. Почему алфавит русского языка назван кириллицей?
  4. Где могут встречаться латинские буквы людям, не изучающим иностранные языки?
  5. Как переключить алфавит мышкой?
  6. Как переключить алфавит с клавиатуры?
- Поработайте с латинскими буквами в Редакторе строки.



Задания исполнителя: наберите указанные символы.

- 1) a b c d e f g h i j k l m;
- 2) n o p q r s t u v w x y z;
- 3) A B C D E F G H I J K L M;
- 4) N O P Q R S T U V W X Y Z;
- 5) raix;
- 6) pease;

- 7) mir;
- 8) computer;
- 9) Internet;
- 10) Monitor — это монитор;
- 11) Mouse — это мышь;
- 12) Memory — это память;
- 13) Keyboard — это клавиатура.

## Работа с информацией

Продолжаем работу с исполнителем **Обработчик**. Попробуйте отгадать четыре новых алгоритма.

### 1. Первый алгоритм Обработчика.

Вход	Выход
1	1
2	1
10	1
11	2
12	3
25	7
48	?

### 2. Второй алгоритм Обработчика.

Вход	Выход
1	1
215	1
345	3
790	0
798	7
654	4
525	2
956	?

### 3. Третий алгоритм Обработчика.

Вход	Выход
1	1
215	5
345	5
790	9
798	9
654	6
525	5
956	?

### 4. Четвертый алгоритм Обработчика.

Вход	Выход
1	0
215	4
345	2
790	9
798	2
654	2
525	3
956	?



## Зачетный класс

1. Информационные носители — это:
  - объекты, которые записывают информацию;
  - объекты, которые передают информацию;
  - объекты, которые обрабатывают информацию;
  - объекты, на которых хранится информация;
  - объекты, которые читают информацию.
2. Отметьте верные высказывания:
  - семена хранят информацию о растении;
  - дневник хранит всю информацию о школьнике;
  - дверной косяк всегда хранит информацию о росте;
  - дверной косяк может хранить информацию о росте;

- еда хранит информацию о собаке;
- собака хранит информацию о еде.

### 3. Отметьте верные высказывания:

- слова хранятся в азбуке;
- азбука нужна для сохранения слов;
- нотная азбука хранит мелодии;
- ноты нужны для сохранения мелодий;
- речь можно сохранить на магнитной ленте;
- магнитная лента всегда хранит речь.

### 4. Отметьте верные высказывания:

- используя 0 и 1, можно сохранить любой знак;
- используя магнитные свойства материала, можно сохранить любой знак;
- используя углубления на диске, можно сохранить любой знак;
- используя 1 и 2, можно сохранить любой знак.

### 5. Отметьте верные свойства ОЗУ:

- сохраняет информацию после отключения питания;
- расположено внутри системного блока;
- самая быстрая память компьютера;
- расположено внутри монитора;
- не сохраняет информацию после отключения питания.

### 6. Отметьте верные свойства винчестера:

- это оптический диск;
- это магнитный диск;
- это съемный носитель;
- это постоянный носитель;
- не сохраняет информацию после отключения питания.

### 7. Отметьте верные свойства дискеты:

- это оптический диск;
- это магнитный диск;
- это съемный носитель;
- это постоянный носитель;
- сохраняет информацию после отключения питания.

### 8. Отметьте верные свойства лазерного диска:

- это оптический носитель;
- это магнитный носитель;
- это съемный носитель;
- это постоянный носитель;
- это самый быстрый носитель.

### 9. Интернет — это:

- объединение компьютеров в классе;
- соединение компьютеров по телефону;
- соединение компьютеров по проводам;
- объединение компьютеров по всему миру;
- компьютеры за границей.

### 10. Отметьте верные свойства Интернета:

- для связи могут использоваться телефонные линии;
- для передачи по телефону нужен модем;
- компьютер используется для входа в Интернет;
- через Интернет можно совершать покупки;
- это самый большой “носитель” информации.

### 11. Где Интернет хранит информацию:

- в телефонных линиях;
- в модеме;
- на винчестерах компьютеров;
- в головах людей;
- в бумажных книгах.

### 12. Напишите слово “Интернет” латинскими буквами.

## Книга для учителя

## 4. ХРАНЕНИЕ ИНФОРМАЦИИ

## Информационные компьютерные носители

В качестве компьютерных информационных носителей используются:

- оперативная память;
- магнитные диски (постоянные и сменные);
- магнитные ленты;
- оптические диски (CD и DVD).

Ниже приводятся объемы этих носителей и скорость чтения информации.

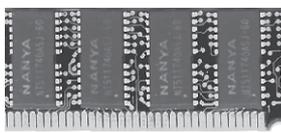
Числа показаны довольно условные, но достаточно реальные для того, чтобы сравнить эти информационные носители между собой.

Носитель	Емкость (Мбайт)	Скорость (Мбайт/с)	Примечание
ОЗУ	256	300	Информация пропадает при отключении питания
Винчестер	40000	30	Постоянный носитель
Флоппи-диск	1,44	0,4	Сменный носитель
Лента	8000	1	Сменный носитель
Диски CD	700	4	Сменный носитель
Диски DVD	20000	2	Сменный носитель

Магнитным носителям присущи такие недостатки, как невысокая стойкость к внешним электромагнитным воздействиям, а также старение и изнашивание поверхности с нанесенным магнитным слоем, называемое “осыпанием”.

На оптические диски (CD и DVD) не влияет электромагнитное излучение. К тому же они более долговечны, так как их информационная поверхность покрыта толстым слоем прочного прозрачного пластика. Емкость одного носителя CD составляет около 700 мегабайт, а для DVD-диска она вчетверо больше.

Так выглядит модуль компьютерной памяти.



Это постоянный магнитный диск компьютера — винчестер в сборе с устройством чтения/записи.



Это устройство чтения/записи сменных флоппи-дисков.



Это флоппи-диск.



Здесь показаны два устройства для работы с магнитной лентой — стримеры. Рядом кассета с лентой.



Это устройство чтения лазерных дисков. Сам лазерный диск виден в устройстве подачи дисков.

На заре компьютерной информатики в большом ходу были бумажные носители информации — перфокарты и перфоленты.

Идея разработки перфокарты, принадлежащая Герману Холлериту (США, 29.02.1860—17.11.1929), возникла еще до появления первого компьютера.

В 1884 году Герман Холлерит получил патент на “машину для переписи населения”.



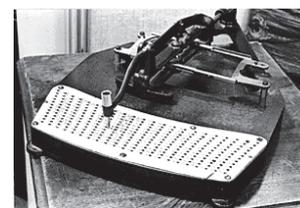
Рисунок с сайта [www.peoples.ru](http://www.peoples.ru)



Эта машина включала в себя клавишный перфоратор, позволяющий перфорировать около 100 отверстий в минуту на бумажном прямоугольнике — перфокарте.

Машина Холлерита была использована для переписи населения в 1890 году.

В 1896 году Холлерит основал фирму по продаже своих машин, а в 1911 году продал ее другой фирме, которая вскоре стала именоваться IBM (*International Business Machines Corp.*).

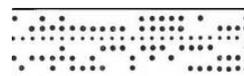


Фотография с сайта [old.submarine.ru](http://old.submarine.ru)

Здесь показана перфокарта IBM в том виде, в котором она долгое время служила в качестве компьютерного носителя информации.

На перфокарте размещены 80 колонок и 12 строк. Прямоугольные отверстия могут быть отперфорированы в любой из  $80 \cdot 12 = 960$  позиций на карте.

Каждая колонка кодирует один символ.



Следом за перфокартами появились перфоленты. Они пришли в компьютерную информатику из телеграфной техники. Перфоленты уступают по прочности перфокартам, но устройства ввода с перфолент дешевле и имеют большее быстродействие.

## Ответы на вопросы

1. Хранят ли информацию предметы вокруг нас? Например, хранит ли информацию стол, на котором стоит компьютер?

Конечно. Стол, как информационный носитель, может “рассказать”:

— о фабрике, которая его изготовила (по фабричной метке);

— о том, хорошие ли на этой фабрике специалисты (по качеству стола);

— о современной моде на столы (по внешнему виду изделия);

— о породе дерева, которое пошло на изготовление;

— о материалах (грунтовках, красках, лаках), которыми была обработана поверхность стола;

— о том, как владелец стола относится к своей собственности (по слою пыли и царапинам).

2. Почему из семечка кабачка не вырастает елка?

Потому что семена растений содержат всю информацию о растении, которое их породило.

3. Картошка хранится в подвале. Хранится ли вместе с картошкой информация?

Если весной посадить картофелину, то из нее вырастет не апельсиновое дерево, а картофельный куст. Это означает, что каждый отдельный клубень содержит полную информацию о том, каким должен быть картофельный куст и как надо расти и развиваться, чтобы этот куст действительно вырос.

4. Как люди могут узнать о жизни своих предков, живших много лет назад?

По тем предметам, которые люди оставили после себя. Наука, которая занимается изучением таких предметов, называется “археологией”.

5. Что такое информационный носитель?

Объекты, на которых хранится информация, называются “информационными носителями”.

6. Информационные носители существуют в природе. Зачем человек придумывает новые информационные носители?

Действительно, семена растений, реки, озера, любые природные объекты вокруг нас являются информационными носителями.

Человек в процессе своей деятельности приобретает новые знания, которые надо уметь сохранить, чтобы передать другим людям. Это нужно для того, чтобы все человечество пользовалось тем, что придумывается каждым отдельным человеком.

Леонардо да Винчи (1452—1519) разработал теоретические основы механической счетной машины арифмометра. А Блез Паскаль в 1642 году сделал первый арифмометр. Люди узнали об этом не из природных источников информации, а из рукотворных носителей (журналы, книги) и наладили производство этих полезных приборов.

7. Какой самый важный информационный носитель придумал человек?

Это, конечно, бумага и печатная книга.

До появления бумаги люди писали на коже животных, коре деревьев, листьях, камнях, вырубали рисунки и надписи на скалах. Эти древние информационные носители были очень неудобными в ис-

пользовании и дорогими. Скалу с рисунками не отправишь как письмо другу в соседний район, а пергамент (носитель из кожи) очень дорог.

Первые книги были рукописными. И поэтому существовали в очень небольшом количестве экземпляров. Настоящая революция в области бумажных информационных носителей произошла с изобретением книгопечатания: книги (а значит, и информация в них) стали доступны широким массам.

Первые опыты книгопечатания были предприняты китайцем Би Шэном в 1041—1048 годах. Возникновение книгопечатания в Европе связывают с немцем Иоганном Гутенбергом и 40-ми годами XV века. Основателем книгопечатания в России был Иван Федоров. В 1564 году он выпустил в Москве первую печатную книгу “Апостол”.

Иван Федоров печатал свои книги на ручном печатном станке.

Немец Фридрих Кёниг придумал более совершенную машину. Первый экземпляр этой машины был установлен в лондонской типографии газеты “Таймс” в 1814 году. Машина печатала 800 листов в час (против 150 на ручном станке).

8. Учитель записал на доске условия контрольной работы; ученики выполняют эту работу. Где и как хранится информация об условиях задачи в ходе выполнения контрольной работы?

Сначала информация хранилась у учителя в голове (или на листочке).

Затем она стала храниться на классной доске. Ребята переписали задание к себе, и информация стала храниться у них в тетрадях. А Игорь не стал записывать условие в тетрадь, он его запомнил и сохранил в своей голове.

9. Вася сказал другу, что запомнил номер его телефона. Где Вася хранит эту информацию?

В голове. А быть может, Вася записал телефон в блокнот или на листочек. Вася мог записать телефон и на компьютерный диск или отпечатать его на принтере. Мог бы записать и на магнитофон.

Вася мог бы нацарапать телефон ножом на столе, но он воспитанный мальчик и, конечно, так не сделал. Вася мог бы зубилом и молотком вырубить телефон на камне, но это не пришло ему в голову.

10. Егор решил записать песню, которая ему понравилась. Какой информационный носитель он может использовать?

Магнитная лента, вероятно, самый доступный носитель. Но можно, используя компьютер, записать песню на магнитный или лазерный диск.

11. В следующих примерах укажите информационный носитель и тип информации, которая хранится на носителе:

— письмо от друга;

— магнитная лента с песнями;

— газета;

— кинофильм;

— видеофильм;

— табличка с номером дома;

— билет на самолет;

— компьютерная игра.

— Письмо от друга. Бумага, визуальная информация (текстовая, а возможно, смешанная: текст, числа, рисунки).

- Магнитная лента с песнями. Магнитный носитель (лента), звуковая информация.
- Газета. Бумага, визуальная информация (текст, числа, графика).
- Кинофильм. Кинолента, визуальная информация.
- Видеофильм. Магнитный носитель (лента), визуальная информация.
- Табличка с номером дома. Металлическая пластинка, визуальная информация (числовая).
- Билет на самолет. Бумага, визуальная информация.
- Компьютерная игра. Магнитный или лазерный диск, визуальная информация.

12. Как книги хранят человеческую речь?

При помощи алфавита, знаками которого записываются слова и предложения.

13. Как сохраняют информацию о музыкальном произведении?

Музыканты записывают мелодию при помощи специальных знаков — нот.

14. Как хранится информация на фотопленке?

В виде менее и более прозрачных участков.

15. Как хранят изображения? Назовите информационные носители.

Картины рисуют на бумаге, холстах, коже, металле, дереве, асфальте, песке, экране компьютера. Картины с экрана компьютера можно записать на магнитный или лазерный диск. Картину на холсте или бумаге можно сфотографировать и хранить на фотопленке.

Сейчас в большом ходу цифровые фотоаппараты, которые вместо пленки используют магнитные носители и носители, подобные ОЗУ компьютера на микросхемах. Такие изображения легко перенести на компьютер.

Изображение можно отсканировать, то есть перенести в компьютерную память при помощи специального устройства — сканера.

16. Старинные бухгалтерские счета хранят информацию или обрабатывают ее?

Счета запоминают числа, а математические операции производит человек.

Правда, счета помогают ему в этом, механизмируя вычислительный труд.

17. Какие устройства использует компьютер для хранения информации?

- оперативная память;
- магнитные диски (постоянные и сменные);
- магнитные ленты;
- оптические диски.

18. Объясните принципы работы магнитного диска.

Для записи информации на магнитный диск используют магнитные свойства материала, из которого сделан диск: намагниченный участок поверхности диска означает число 1, а ненамагниченный — число 0.

19. Объясните принципы работы лазерного диска.

Информация на лазерный диск записывается в виде микроскопических углублений на его поверхности. Яркость отражения лазерного луча уменьшается, когда он попадает в углубления. Таким образом, удастся хранить и считывать информацию: углубление соответствует числу 0, а отсутствие углубления — числу 1.

20. Где хранится информация в Интернете?

Информация хранится на компьютерах, объединение которых и образует Интернет. Как правило, в Интернете доступна информация, которая хранится на винчестере компьютера.

21. Хранятся ли в Интернете товары, например, обычные бумажные книги?

Конечно, нет. Когда через Интернет покупается, например, книга, она просто выбирается в каталоге, отображаемом на экране компьютера. Когда книга выбрана, в магазин через Интернет посылается заказ, а книга приходит бандеролью по обычной почте.

## Ответы на вопросы практикума

Алгоритмы **Обработчика** (работает только с числовой информацией):

1. Сумма цифр числа.
2. Наименьшая цифра числа.
3. Наибольшая цифра числа.
4. Разница между наибольшей и наименьшей цифрами числа.

## Решения зачетного класса

1. Информационные носители — это:

- объекты, которые записывают информацию;
- объекты, которые передают информацию;
- объекты, которые обрабатывают информацию;
- объекты, на которых хранится информация;
- объекты, которые читают информацию.

Информационные носители — это объекты, на которых хранится информация.

2. Отметьте верные высказывания:

- семена хранят информацию о растении;
- дневник хранит всю информацию о школьнике;
- дверной косяк всегда хранит информацию о росте;
- дверной косяк может хранить информацию о росте;
- еда хранит информацию о собаке;
- собака хранит информацию о еде.

Семена действительно хранят информацию о растении.

Дневник не хранит **всю** информацию о школьнике. Например, он не хранит сведения о том, какой длины нос у школьника.

Дверной косяк хранит информацию о росте только тогда, когда на нем сделаны соответствующие пометки. Он **может** хранить информацию о росте, но не всегда.

Еда не хранит информацию о собаке, а вот собака, конечно, имеет понятие о вкусной мозговой косточке!

3. Отметьте верные высказывания:

- слова хранятся в азбуке;
- азбука нужна для сохранения слов;
- нотная азбука хранит мелодии;
- ноты нужны для сохранения мелодий;
- речь можно сохранить на магнитной ленте;
- магнитная лента всегда хранит речь.

Азбука нужна для сохранения слов, но никак не наоборот! В азбуке хранятся символы, знаки, но не слова. То же относится и к нотной азбуке.

Речь можно сохранить на магнитной ленте, но магнитная лента не всегда хранит речь. Может хранить и картинку, переписанную с компьютера.

4. Отметьте верные высказывания:

- используя 0 и 1, можно сохранить любой знак;
- используя магнитные свойства материала, можно сохранить любой знак;
- используя углубления на диске, можно сохранить любой знак;
- используя 1 и 2, можно сохранить любой знак.

Используя 0 и 1, можно сохранить любой знак. Докажем это. Пусть всего имеется 20 знаков. Тогда каждый знак можно записать в виде набора из 20 нулей, в котором на нужном месте ноль заменен единицей:

10000000000000000000 — первый знак;

01000000000000000000 — второй знак;

00100000000000000000 — третий знак;

...

00000000000000000001 — последний знак.

Доказательство не изменится при любом числе знаков, просто запись каждого знака будет длиннее. Существуют, конечно, гораздо более компактные способы записи нулями и единицами.

Используя магнитные свойства материала, можно сохранить любой знак, так как его можно закодировать единицами и нулями (намагниченными и ненамагниченными участками).

Используя углубления на диске, можно сохранить любой знак, так как его можно закодировать единицами и нулями (ямками и ровными участками). Любой знак можно записать двумя цифрами. Все равно, какие цифры использовать, 0 и 1 или 1 и 2.

5. Отметьте верные свойства ОЗУ (правильные ответы помечены галочкой):

- сохраняет информацию после отключения питания;
- расположено внутри системного блока; ✓
- самая быстрая память компьютера; ✓
- расположено внутри монитора;
- не сохраняет информацию после отключения питания. ✓

6. Отметьте верные свойства винчестера (правильные ответы помечены галочкой):

- это оптический диск;
- это магнитный диск; ✓

- это съемный носитель;
- это постоянный носитель; ✓
- не сохраняет информацию после отключения питания.

7. Отметьте верные свойства дискеты (правильные ответы помечены галочкой):

- это оптический диск;
- это магнитный диск; ✓
- это съемный носитель; ✓
- это постоянный носитель;
- сохраняет информацию после отключения питания. ✓

8. Отметьте верные свойства лазерного диска (правильные ответы помечены галочкой):

- это оптический носитель; ✓
- это магнитный носитель;
- это съемный носитель; ✓
- это постоянный носитель;
- это самый быстрый носитель.

9. Интернет — это (правильный ответ помечен галочкой):

- объединение компьютеров в классе;
- соединение компьютеров по телефону;
- соединение компьютеров по проводам;
- объединение компьютеров по всему миру; ✓
- компьютеры за границей.

10. Отметьте верные свойства Интернета (правильные ответы помечены галочкой):

- для связи могут использоваться телефонные линии; ✓
- для передачи по телефону нужен модем; ✓
- компьютер используется для входа в Интернет; ✓
- через Интернет можно совершать покупки; ✓
- это самый большой “носитель” информации. ✓

11. Где Интернет хранит информацию (правильный ответ помечен галочкой):

- в телефонных линиях;
- в модеме;
- на винчестерах компьютеров; ✓
- в головах людей;
- в бумажных книгах.

На винчестерах компьютеров, объединение которых и представляет собой Интернет.

12. Напишите слово “Интернет” латинскими буквами.

INTERNET

*Продолжение следует*

<p><b>Гл. редактор</b> С.Л. Островский <b>Зам. гл. редактора</b> А.И. Сенокосов <b>Редакция:</b> Е.В. Андреева Н.Л. Беленькая Л.Н. Картвелишвили Н.П. Медведева <b>Дизайн и верстка:</b> Н.И. Пронская <b>Корректоры:</b> Е.Л. Володина, С.М. Подберезина</p>	<p><b>©ИНФОРМАТИКА 2002</b> выходит четыре раза в месяц При перепечатке ссылка на ИНФОРМАТИКУ обязательна, рукописи не возвращаются</p>	<p><b>Адрес редакции и издателя:</b> 121165, Киевская, 24 тел. 249-48-96 Отдел рекламы тел. 249-98-70</p>	<p><b>Учредитель: ООО “Чистые пруды”</b> Зарегистрировано в Министерстве РФ по делам печати. ПИ № 77-7230 от 12.04.2001. Отпечатано в ОИД “Медиа-Пресса”, 125993, ГСП-3, Москва, А-40, ул. “Правды”, 24. Тираж 7000 экз. Срок подписания в печать по графику 13.02.2002. Номер подписан 13.02.2002. Заказ № Цена свободная</p>
<p><b>ИНДЕКС ПОДПИСКИ</b> для индивидуальных подписчиков <b>32291</b> комплекта изданий <b>32744</b></p>			
<p><b>Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84</b></p>		<p><b>Internet: inf@1september.ru</b> <b>WWW: http://www.1september.ru</b></p>	

**ИЗДАТЕЛЬСКИЙ  
ДОМ «ПЕРВОЕ  
СЕНТЯБРЯ»,  
ГЛАВНЫЙ  
РЕДАКТОР —  
А. СОЛОВЕЙЧИК**

Газеты ИЗДАТЕЛЬСКОГО ДОМА: **Первое сентября** — гл. ред. Е.Бирюкова, **Английский язык** — гл. ред. А.Громушкина, **Библиотека в школе** — гл. ред. О.Громова, **Биология** — гл. ред. Н.Иванова, **Воскресная школа** — гл. ред. монах Киприан (Яценко), **География** — гл. ред. О.Коротова, **Дошкольное образование** — гл. ред. М.Аромштам, **Здоровье детей** — гл. ред. А.Лекманов, **Информатика** — гл. ред. С.Островский, **Искусство** — гл. ред. Н.Исмаилова, **История** — гл. ред. А.Головатенко, **Литература** — гл. ред. Г.Красухин, **Математика** — гл. ред. И.Соловейчик, **Начальная школа** — гл. ред. М.Соловейчик, **Немецкий язык** — гл. ред. М.Бузова, **Русский язык** — гл. ред. Л.Гончар, **Спорт в школе** — гл. ред. Н.Школьникова, **Управление школой** — гл. ред. А.Адамский, **Физика** — гл. ред. Н.Козлова, **Французский язык** — гл. ред. Г.Чесновицкая, **Химия** — гл. ред. О.Блохина, **Чудесная газета** — гл. ред. М.Аромштам, **Школьный психолог** — гл. ред. М.Сартан.