

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 •

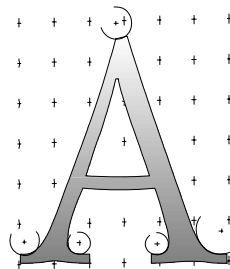
Информацию о Дне учителя информатики см. на с. 2

№ 14 (351) 8–15 апреля 2002

ПОДПИСКА: (095) 249-47-58

Еженедельная газета Издательского дома «ПЕРВОЕ СЕНТЯБРЯ»

# ИНФОРМАТИК



## ГЕОМЕТР

В 2002 году исполняется 210 лет со дня рождения великого русского математика Николая Ивановича Лобачевского (1792–1856), чье имя занимает одно из первых мест в истории мировой науки



**23 февраля 1826 г.** на заседании физико-математического отделения Императорского Казанского университета был прочитан доклад под названием “Сжатое изложение начал геометрии со строгим доказательством теоремы о параллельных линиях”. Указанная дата рассматривается как день рождения неевклидовой геометрии и является как бы гранью двух эпох. Предшествующие двадцать пять веков были эпохой создания классической геометрии. Принципы, лежащие в ее основе, а вместе с ними и вся геометрия считались неизбежными, не допускающими никаких изменений. Но в этот день родилась новая геометрия, расширившая человеческие представления о пространстве и форме, изменившая устоявшиеся взгляды на всю мате-

матику и окружающий мир [1]. Создателю новой геометрии, Николаю Ивановичу Лобачевскому, было тогда 33 года.

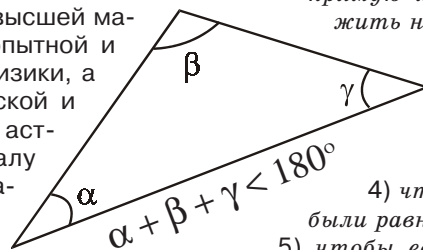
Вся жизнь Лобачевского была отдана науке и Казанскому университету, который он окончил в 1811 г. и ректором которого (с 1827 г.) долгое время являлся [1, 2]. Он вел активную преподавательскую работу, причем читал курсы не только элементарной и высшей математики, но и опытной и теоретической физики, а также теоретической и наблюдательной астрономии. К началу 1820-х годов Лобачевский написал два учебника (опубликованных много позже): “Алгебра. Исчисление конечных” и “Геометрия”, которые в рукописном виде распространялись среди студентов. И все эти годы он занимался теорией параллельных.

Проблема параллельных уходит своими корнями в античную Грецию. В знаменитых “Началах” Евклида пять постулатов (требований

геометрического характера, которые нужно принять, чтобы на их основе делать все дальнейшие выводы). Вот их оригинальные формулировки [1].

“Нужно потребовать:

- 1) чтобы от каждой точки ко всякой другой точке можно было провести прямую линию;
- 2) чтобы каждую ограниченную прямую можно было продолжить неограниченно;
- 3) чтобы из любого центра можно было описать окружность любым радиусом;
- 4) чтобы все прямые углы были равны между собой;
- 5) чтобы если при пересечении двух прямых третьей сумма внутренних односторонних углов оказалась меньше двух прямых, то эти прямые при достаточном их продолжении пересекались бы и притом с той стороны, с которой эта сумма меньше двух прямых”.



Окончание читайте на с. 32

## Читайте в номере

### Страницы повышения квалификации ..... 3–11

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

“На большинстве всероссийских и многих областных олимпиадах по информатике по крайней мере одна из задач связана с геометрическими понятиями”.

Приглашаем на лекцию, тема которой — “Геометрические задачи на олимпиадах по информатике”. Попробуйте решить с ребятами задачи: “Стена”, “Раздел царства”, “Рандеву”, “Куда идем мы с Пятачком?” — и другие.

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

“Каждый учитель должен понимать свой маневр” — таким, по мнению автора, мог бы быть один из основных тезисов пока не написанной книги “Наука обучать”.

Тема сегодняшней лекции — “Что за чем, а также зачем преподавать”.

### Экзамены..... 12–15, 18–20

#### Е.А. Еремин, А.П. Шестаков. Примерные ответы на примерные билеты

А теперь предложите вашим ученикам рассказать о назначении и основных возможностях текстового и графического редакторов.

Примерные билеты (и ответы на них) для проведения итоговой аттестации выпускников 9-х классов. В этом номере представлены девятый и десятый билеты.

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

#### DVD-диски

Когда появились цифровые универсальные, или многоцелевые, диски? Чем отличается DVD-Video от DVD-ROM? Что такое DVD-R и DVD-RAM? Каковы особенности стандарта записи звука Dolby Digital? Почему DVD-проигрыватель, купленный в России, может отказаться воспроизводить диски, привезенные из Америки или Японии? Все о DVD.

### Уроки ..... 21–27

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

Как известно, полученная информация не всегда пригодна к употреблению без дальнейшей обработки. Информацию, которую обрабатывают, называют **исходной**.

Продолжение второй книги, рассказывающей о способах хранения, передачи и обработки информации.

### Задачи..... 28–31

#### О.В. Туркин. Использование электронной таблицы Microsoft Excel и языка Visual Basic for Application для построения графиков состояния идеального газа

Пусть ребята вспомнят уравнение состояния идеального газа, а затем предложите им создать компьютерную программу, позволяющую строить графики газовых процессов и замкнутых циклов одновременно в трех системах координат, а также “иллюстрировать” энергетические превращения в этих процессах.



### День учителя информатики в Москве

15 апреля,  
Московский городской дом учителя  
(улица Пушечная, 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.

# Олимпиады по информатике.

## Пути к вершине

Лекции читает Е.В. Андреева

### Лекция 11. Геометрические задачи на олимпиадах по информатике

На большинстве всероссийских и многих областных олимпиадах по информатике по крайней мере одна из задач связана с геометрическими понятиями. Причем сформулированы они чаще всего в терминах вычислительной геометрии, и описание таких объектов, как прямая, отрезок, окружность, треугольник и т.д., производится путем задания координат точек, характеризующих эти объекты, в той или иной системе координат. Прежде чем мы перейдем к рассмотрению этого класса олимпиадных задач, перечислим элементарные подзадачи (иногда это просто формулы из курса математики), на решение которых обычно опираются решения задач вычислительной геометрии.

#### Основные формулы и алгоритмы

##### 1. Уравнения линий.

- 1.1. Уравнение прямой, проходящей через две различные точки, заданные своими координатами.
- 1.2. Уравнение прямой, перпендикулярной данной (заданной двумя точками или коэффициентами уравнения) и проходящей через заданную координатами точку.
- 1.3. Уравнение прямой, параллельной данной и находящейся от нее на заданном расстоянии.
- 1.4. Параметрическое уравнение отрезка, заданного координатами своих вершин, или луча, заданного координатами начальной точки, и одной из точек, принадлежащих лучу.
- 1.5. Уравнение прямой, частью которой является биссектриса угла, образованного между двумя векторами.
- 1.6. Уравнение окружности, заданной координатами центра и радиусом.
- 1.7. Уравнение окружности по координатам трех заданных точек, не лежащих на одной прямой.
- 1.8. Уравнение касательных к окружности, проходящих через заданную вне окружности точку, и нахождение координат точки касания.

##### 2. Взаимное расположение точек и фигур.

- 2.1. Проверка принадлежности точки, заданной своими координатами, прямой, лучу или отрезку.
- 2.2. Проверка того, что три точки, заданные своими координатами, не лежат на одной прямой (т.е. образуют треугольник).
- 2.3. Проверка существования треугольника со сторонами, длины которых известны.

2.4. Определение вида треугольника, заданного координатами вершин: остроугольный, прямоугольный или тупоугольный.

2.5. Взаимное расположение двух точек относительно заданной прямой.

2.6. Проверка принадлежности точки внутренней области многоугольника, заданного координатами вершин его границы, в порядке обхода (например, по часовой стрелке).

##### 3. Взаимное расположение фигур и нахождение точек их пересечения.

3.1. Определение взаимного расположения двух прямых и нахождение точки их пересечения, если таковая имеется.

3.2. Определение взаимного расположения двух отрезков или лучей и нахождение множества точек их пересечения, если оно не пусто.

3.3. Определение взаимного расположения двух окружностей и нахождение точек их пересечения, если таковые имеются.

3.4. Определение взаимного расположения окружности и прямой и нахождение точек их пересечения (или точки касания), если таковые имеются.

##### 4. Расстояние от точки до фигуры или между фигурами.

- 4.1. Определение расстояния между двумя точками.
- 4.2. Определение расстояния от точки до прямой (луча, отрезка).
- 4.3. Определение расстояния от точки до окружности.
- 4.4. Определение расстояния от точки до многоугольника.
- 4.5. Определение расстояния между двумя отрезками.
- 4.6. Определение длины наименьшей дуги окружности; центр окружности и концевые точки дуги заданы своими координатами.

##### 5. Особые точки многоугольников и множеств $N$ точек плоскости.

5.1. Нахождение координат центра окружности, описанной около треугольника или правильного  $N$ -угольника, заданного координатами вершин (см. 1.7).

5.2. Нахождение координат центра окружности минимального радиуса, внутри которой находятся все заданные  $N$  точек.

5.3. Нахождение радиуса и координат центра окружности, вписанной в треугольник, заданный координатами вершин.

5.4. Нахождение координат центра тяжести  $N$  точек без учета масс (для трех точек это точка пересечения медиан соответствующего треугольника) и с учетом масс точек.

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

5.5. Нахождение координат точки, сумма расстояний от которой до всех вершин треугольника минимальна (точки Штейнера).

5.6. Нахождение из  $N$  точек плоскости пары наиболее удаленных друг от друга или ближайших друг к другу точек.

## 6. Многоугольник и множество точек плоскости.

6.1. Проверка простоты многоугольника (отсутствие самокасааний и самопересечений). Здесь и далее многоугольник задается путем последовательного перечисления координат его вершин в порядке обхода, например, против часовой стрелки.

6.2. Проверка выпуклости многоугольника.

6.3. Вычисление площади треугольника, заданного координатами своих вершин, без использования формулы Герона и нахождения высоты треугольника (иногда по аналогичной формуле требуется найти значение так называемой “ориентированной” площади треугольника).

6.4. Вычисление площади произвольного многоугольника.

6.5. Построение выпуклой оболочки для множества из  $N$  точек плоскости. (Выпуклой оболочкой конечного множества точек называется наименьший выпуклый многоугольник, содержащий все точки из указанного множества, при этом часть точек оказывается внутри многоугольника.)

6.6. Построение замкнутой ломаной без самопересечений и самокасааний, проходящей через  $N$  заданных точек плоскости.

## 7. Векторы.

7.1. Координаты вектора, заданного путем перечисления координат его начала и конца.

7.2. Перевод координат вектора из декартовой в полярную систему координат и обратно.

7.3. Сумма и разность двух векторов в декартовой и полярной системах координат.

7.4. Скалярное произведение двух векторов в декартовой и полярной системах координат.

7.5. Косое произведение двух векторов в декартовой и полярной системах координат.

7.6. Определение величины угла между двумя векторами, значение угла при этом лежит в диапазоне  $[0, 2\pi)$  и зависит от порядка перечисления векторов.

Большинство из перечисленных задач либо не требуют пояснений, либо приведены в [1–4]. Напомним лишь наиболее важные из них. Причем основным инструментом для построения наиболее простых формул во многих задачах вычислительной геометрии является *косое произведение*. Поэтому рассмотрение начнем с вопросов, с ним связанных.

## Косое произведение в задачах вычислительной геометрии

Под *косым произведением* векторов  $p_1$  и  $p_2$  с декартовыми координатами  $(x_1, y_1)$  и  $(x_2, y_2)$  можно понимать ориентированную площадь параллелограмма, образованного точками  $(0, 0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_1 + x_2, y_1 + y_2)$ , которая равна  $p_1 \times p_2 = -p_2 \times p_1 = x_1 y_2 - x_2 y_1$  (задача 7.5). Косое произведение напрямую связано с понятием векторного произведения (но в отличие от послед-

него это скаляр). Поэтому в литературе по вычислительной геометрии иногда используется именно это понятие. По-другому косое произведение, как и векторное, обозначается  $[p_1, p_2]$ . Если два вектора провести из общей начальной точки, то их косое произведение больше нуля, если угол между первым и вторым вектором ориентирован так же, как угол между первым и вторым базисными векторами, и меньше нуля — в противном случае. Косое произведение ненулевых векторов равно нулю тогда и только тогда, когда они коллинеарны (сонаправлены или противоположно направлены).

В задаче 3.2 проверить наличие пересечения у двух отрезков (а зачастую нас интересует лишь сам факт пересечения) несложно именно с использованием косого произведения. Пусть первый отрезок задан точками  $p_1$  и  $p_2$ , а второй —  $p_3$  и  $p_4$  (так же обозначаются векторы с соответствующими координатами). Обозначим  $x_{max1}$  и  $x_{min1}$  — максимальную и минимальную из первых координат первого отрезка,  $x_{max2}$  и  $x_{min2}$  — то же для второго отрезка. Для второй координаты аналогично имеем  $y_{max1}$ ,  $y_{min1}$ ,  $y_{max2}$  и  $y_{min2}$ . Упомянутые отрезки пересекаются тогда, когда

а) пересекаются ограничивающие их прямоугольники, т.е.  $x_{max1} \geq x_{min2}$ ,  $x_{max2} \geq x_{min1}$ ,  $y_{max1} \geq y_{min2}$  и  $y_{max2} \geq y_{min1}$ ;

б) косые произведения

$$(p_3 - p_1) \times (p_2 - p_1) \text{ и } (p_4 - p_1) \times (p_2 - p_1)$$

имеют разный знак, точнее,

$$[(p_3 - p_1), (p_2 - p_1)] \cdot [(p_4 - p_1), (p_2 - p_1)] \leq 0;$$

$$\text{в) } [(p_1 - p_3), (p_4 - p_4)] \cdot [(p_2 - p_3), (p_4 - p_3)] \leq 0.$$

Последние два условия означают, что концы одного отрезка лежат по разные стороны от прямой, которой принадлежит другой отрезок. Первое же условие исключает из специального рассмотрения случай равенства нулю всех четырех косых произведений, при котором отрезки лежат на одной прямой и могут как пересекаться, так и нет.

Площадь треугольника (задача 6.3) равна половине модуля косого произведения двух векторов, образованных любыми двумя его сторонами. При необходимости вычислить ориентированную площадь модуль брать не нужно.

Тогда расстояние от точки  $C$  до прямой, заданной координатами точек  $A$  и  $B$  (задача 4.2), можно подсчитать как отношение модуля косого произведения векторов  $CA$  и  $CB$  к длине отрезка  $AB$  (данная формула следует из двух способов вычисления площади треугольника).

Площадь произвольного многоугольника с вершинами  $p_0, p_1, \dots, p_{n-1}$ , перечисленными в порядке его обхода против часовой стрелки (задача 6.4), можно вычислить как сумму ориентированных площадей треугольников, образованных векторами  $p_i$  и  $p_{i+1}$ ,  $i = 0, \dots, n - 1$ ;  $i + 1$  вычисляется по модулю  $n$ .

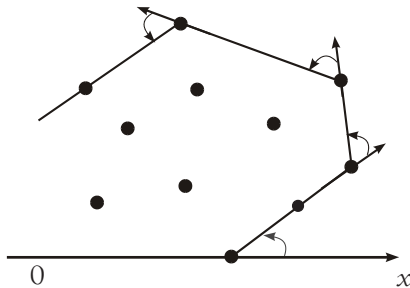
Выпуклость многоугольника (задача 6.2) с вершинами  $p_0, p_1, \dots, p_{n-1}$ , перечисленными в порядке его обхода, легко проверить с помощью сравнения знаков косого произведения пар векторов  $(p_{i+1} - p_i)$  и  $(p_{i+2} - p_{i+1})$ ,  $i = 0, \dots, n - 1$ ;  $i + 1$  и  $i + 2$  вычисляются по модулю  $n$ . В случае выпуклого многоугольника знаки у всех указан-

ных произведений совпадают (если мы знаем направление обхода, то знак косых произведений для выпуклого многоугольника определен: при обходе по часовой стрелке произведения отрицательны, а против часовой стрелки — положительны).

На этом способы полезного применения косого произведения отнюдь не исчерпаны.

## Выпуклая оболочка множества $N$ точек плоскости

Задача состоит в том, чтобы перечислить все точки, принадлежащие границе выпуклой оболочки заданного множества точек, в порядке ее обхода, например, против часовой стрелки (в некоторых задачах требуется перечислить только угловые точки). Для эффективного решения этой задачи существует несколько различных алгоритмов (см. [1] — [4]). Приведем наиболее простую реализацию одного из них — алгоритма Джарвиса.



Перечисление точек искомой границы выпуклого многоугольника начнем с правой нижней точки, которая заведомо принадлежит границе выпуклой оболочки. Обозначим ее координаты  $(x_0, y_0)$ . Следующей, при указанном порядке обхода, будет точка  $(x_1, y_1)$ , для которой угол между осью  $Ox$  и вектором  $(x_0, y_0) - (x_1, y_1)$  минимален. Если таких точек несколько, то угловой в многоугольнике станет точка, для которой длина вектора  $(x_0, y_0) - (x_1, y_1)$  максимальна, а следующей точкой, принадлежащей выпуклой оболочке, — та, длина вектора у которой минимальна (таким образом, наш выбор будет зависеть от конкретной постановки задачи). Для нахождения следующей точки значения углов между векторами вычислять необязательно. Мы опять можем воспользоваться понятием знака косого произведения. Пусть, далее, мы уже нашли  $i$ -ю вершину выпуклой оболочки  $(x_i, y_i)$ . Тогда  $(i + 1)$ -я точка есть такая точка, еще не вошедшая в выпуклую оболочку, для которой угол между вектором  $(x_{i-1}, y_{i-1}) - (x_i, y_i)$  и вектором  $(x_i, y_i) - (x_{i+1}, y_{i+1})$  минимален, при минимальной длине вектора  $(x_i, y_i) - (x_{i+1}, y_{i+1})$  среди всех векторов с таким углом. Заметим, что для всех остальных точек  $(x, y)$  вектор  $(x_i, y_i) - (x, y)$  будет лежать вне угла, образованного указанными векторами, левее него. Тогда векторное произведение

$$(x_{i+1} - x_i)(y - y_i) - (y_{i+1} - y_i)(x - x_i) \geq 0$$

для любой точки  $(x, y)$ , еще не вошедшей в границу выпуклой оболочки. Следовательно, мы можем сначала считать следующей,  $(i + 1)$ -й, любую еще не вошедшую в выпуклую оболочку точку, а затем вычисляем

указанное выражение для остальных “свободных” точек  $(x, y)$ . Если для одной из них

$$(x_{i+1} - x_i)(y - y_i) - (y_{i+1} - y_i)(x - x_i) < 0,$$

считаем следующей ее и продолжаем проверку остальных точек (аналогично алгоритму поиска минимального элемента в массиве). Если же значение выражения равно 0, то сравниваем квадраты длин векторов, а именно  $(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2$  и  $(x - x_i)^2 + (y - y_i)^2$ .

Таким образом, при решении данной задачи в случае изначально целочисленных координат мы полностью можем избежать применения вещественной арифметики, а следовательно, избавиться от потери точности вычислений. В противном случае в решение могут быть включены “лишние” точки, близкие к границе выпуклой оболочки, или не учтены некоторые из “нужных” точек. Сложность данного алгоритма составит  $O(kN)$ , где  $k$  — количество точек в выпуклой оболочке, в худшем случае равное  $N$ . Существуют алгоритмы решения этой задачи, основанные на предварительной сортировке точек исходного множества, например, по значению угла в полярной системе координат с центром в одной из точек выпуклой оболочки, с вычислительной сложностью  $O(N \log N)$  (алгоритм Грэхема). То есть наиболее трудоемкой задачей оказывается именно сортировка исходных точек.

Приведем программу решения данной задачи с помощью алгоритма Джарвиса:

```

var a, b : array[1..100] of record
    x, y : integer;
    f : boolean
end;

min, m, j, k, n : integer;
begin
    readln(n);
    for i := 1 to n do
        begin
            read(a[i].x, a[i].y);
            a[i].f := false
        end;
    {ищем нижнюю правую точку}
    m := 1;
    for i := 2 to n do
        if a[i].y < a[m].y then m := i else
        if (a[i].y = a[m].y) and
            (a[i].x > a[m].x) then m := i;
    b[1] := a[m];
    a[m] := a[1];
    a[1] := a[m];
    k := 1;
    repeat
        min := 2;
        {ищем первую еще свободную точку}
        while a[min].f do inc(min);
        {ищем очередную вершину выпуклой оболочки}
        for j := 1 to n do
            if (not a[j].f) and
                ((a[min].x - b[k].x) * (a[j].y - b[k].y) -
                 (a[j].x - b[k].x) * (a[min].y - b[k].y) < 0)
            then min := j else
            if (not a[j].f) and
                ((a[min].x - b[k].x) * (a[j].y - b[k].y) -
                 (a[j].x - b[k].x) * (a[min].y - b[k].y) = 0) and
                (sqr(a[min].x - b[k].x) + sqr(a[min].y - b[k].y) >
                 sqr(a[j].x - b[k].x) + sqr(a[j].y - b[k].y))

```

```

    then min := j;
    k := k + 1;
    a[min].f := true;
    b[k] := a[min] {записана очередная вершина}
until min = 1; {пока ломаная не замкнется}
for j := 1 to k - 1 do {печатать результата}
    writeln(b[j].x, ' ', b[j].y);
end.

```

Приведем примеры задач, при решении которых используется построение выпуклой оболочки.

### Задача 1. “Стена”

В англоязычном варианте задача предлагалась на студенческих командных соревнованиях по программированию в Санкт-Петербурге в 2001 г.

Однажды жадный король приказал своему архитектору построить стену вокруг дворца. Король был настолько жадный, что не стал слушать предложения архитектора о построении стены совершенной формы. Вместо этого король приказал потратить на строительство стены определенной высоты и произвольной формы (не обязательно в виде ломаной!) как можно меньше кирпичей, но потребовал, чтобы стена отстояла от дворца не меньше, чем на  $L$  футов. В случае невыполнения условия или перерасхода средств архитектор может лишиться головы.

Ваша задача — помочь бедному архитектору: написать программу, которая определит минимально возможную длину стены, которой можно окружить дворец, и при этом выполнить все требования короля.

Первая строка входного файла содержит 2 числа  $N$  ( $3 \leq N \leq 1000$ ) — количество углов в здании дворца и  $L$  ( $1 \leq L \leq 1000$ ) — минимальное расстояние, на которое стена может приближаться к дворцу. Следующие  $N$  строк описывают координаты на поверхности земли углов дворца, в порядке их обхода по часовой стрелке. Каждая строка содержит два целых числа —  $X_i$  и  $Y_i$ , разделенных пробелом ( $-10\,000 \leq X_i, Y_i \leq 10\,000$ ), которые описывают координаты углов дворца в футах. Все углы дворца различны, а стороны не имеют других общих точек, кроме угловых.

Запишите в выходной файл одно число, определяющее минимальную длину дворца в футах, удовлетворяющую условию задачи. Ответ должен быть записан в виде целого числа, так как с вещественными числами король не знаком, однако округлять его следует так, чтобы целое число футов отличалось от настоящего ответа менее чем на 8 дюймов (в 1 футе 12 дюймов).

Пример входного файла	Выходной файл
9 100	1628
200 400	
300 400	
300 300	
400 300	
400 400	
500 400	
500 200	
350 200	
200 200	

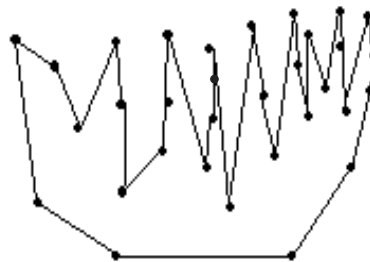
*Решение.* Ответом на данную задачу будет длина границы выпуклой оболочки, увеличенная на длину окружности с радиусом  $L$  и округленная до ближайшего целого.

### Задача 2.

На плоскости заданы  $N$  точек. Построить замкнутую ломаную без самопересечений и самокасааний, проходящую через все заданные точки.

(См., например, [5], аналогичная задача предлагалась на кировской областной олимпиаде в 2002 г.)

*Решение.* Следующий рисунок проиллюстрирует идею одного из возможных способов решения данной задачи:



### Численное решение геометрических задач

В ряде случаев при решении геометрических задач формулы из вычислительной геометрии могут оказаться слишком громоздкими и приводят к решению нелинейных уравнений. Тогда на помощь могут прийти численные (приближенные) методы, позволяющие решить задачу за требуемое время и с нужной точностью. Такой подход был продемонстрирован в [6] при рассмотрении задачи “Фонтан” (ее не следует путать с задачей 6, приведенной ниже).

Из численных методов наиболее часто употребляемым является метод дихотомии (деления пополам). Рассмотрим его применение на примерах.

### Задача 3. “Раздел царства”

XII Всероссийская олимпиада по информатике

Царство царя Гороха представляет собой выпуклый  $N$ -угольник, внутри которого расположены  $K$  селений. Царь решил завещать двум своим сыновьям по полцарства, одинаковые по площади и с равным количеством селений. Для этого он требует разделить царство одной прямолинейной границей.

Напишите программу, строящую границу согласно царской воле. Если граница проходит через селение, то оно может быть либо отнесено к одному из полуцарств, либо разделено на два селения, которые будут отнесены к разным полуцарствам (при нечетном  $K$  граница, естественно, должна разделить какое-то из селений).

Первая строка входного файла содержит количество вершин многоугольника  $N$  ( $3 \leq N \leq 50$ ). В следующих  $N$  строках заданы координаты вершин многоугольника, перечисленные в порядке обхода контура по часовой стрелке. В  $(N + 2)$ -й строке

указано количество селений  $K$  ( $0 \leq K \leq 100$ ), а в последующих  $K$  строках заданы координаты селений. Все координаты — целые числа, не превосходящие по модулю  $10^6$ . Размерами селений следует пренебречь.

В выходной файл нужно вывести координаты любых двух различных точек, через которые следует провести границу. Координаты должны быть выведены с 6 знаками после десятичной точки.

Пример входного файла	Пример выходного файла
4	30.000000 35.000000
9 10	30.000000 15.000000
20 40	
40 40	
51 10	
2	
21 30	
40 20	

*Решение.* Выберем произвольную точку на границе царства. Для поиска прямой, проходящей через эту точку и делящей царство на две равные пока только по площади части, зафиксируем две другие точки границы так, что прямая, проведенная через выбранную и первую из фиксированных точек, делит царство на две неравные части, причем левая (или нижняя для горизонтальной прямой) часть по площади меньше правой (верхней). Прямая же, проходящая через выбранную точку и вторую из фиксированных, делит царство в обратном соотношении. Тогда искомая точка находится между двумя фиксированными и ее можно искать методом деления пополам: рассматривается точка, лежащая на границе посередине между двумя фиксированными, в результате область поиска уменьшается в два раза. Теперь следует подсчитать количество селений в каждой из уже равных по площади частей. Если оно различно, то на границе нужно выбрать еще одну точку, при делении царства, с помощью которой количество селений в половинах будет соотноситься по-иному. Теперь можно применить метод деления пополам для правильного выбора опорной точки.

#### Задача 4. “Рандеву”

VII Всероссийская олимпиада по информатике.

Локаторы дальней космической связи замечают летящий в плоскости орбиты земли неизвестный астероид с координатами  $(x, y)$ . Астероид летит с постоянной скоростью, векторное значение которой равно  $(V_x, V_y)$ . С земли из точки с координатами  $(0, 0)$  немедленно стартует ракета с радиусом действия  $R$  ( $R > 0$ ). Ракета летит по прямой с постоянной скоростью в пределах от 0 до  $W$ .

Требуется определить, может ли ракета подлететь вплотную к астероиду в пределах радиуса ее действия, и найти вектор скорости ракеты, при котором время встречи ракеты с астероидом минимальное.

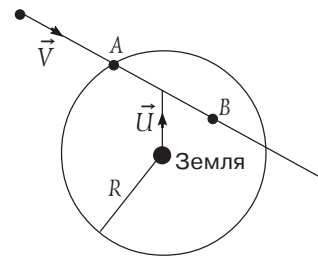
Результат решения задачи должен быть вычислен с погрешностью не более 0.01. Влиянием земли и всех космических объектов пренебречь. Ракета и астероид двигаются в одной плоскости.

В начале входного файла содержится число  $N$  — количество наборов исходных данных (тестов). Далее расположены  $N$  наборов исходных данных; каждый набор — шесть вещественных чисел:  $X, Y, V_x, V_y, W, R$ . Все числа в исходном файле разделяются пробелами и (или) символами перевода строки.

Для каждого набора исходных данных вывести с новой строки вектор скорости  $(U_x, U_y)$  и минимальное время до встречи либо сообщение “Встреча невозможна”.

Пример файла исходных данных	Пример выходного файла
2	Встреча невозможна
5.3 2.8 10.6 5.6 11.0 50.0	3.0 -4.0 0.5
3.0 -4.0 -3.0 4.0 5.0 10.0	

*Решение.* Для решения этой задачи прежде всего необходимо уметь определять взаимное расположение прямой, вдоль которой движется астероид, и окружности с центром на Земле и радиусом  $R$ . Если они не пересекаются, то встреча невозможна, в противном случае требуется отыскать точки их пересечения. Если в точке пересечения  $A$  (см. рисунок) встреча невозможна, то для поиска точки встречи с минимальным временем можно опять же применить дихотомию. В качестве второй начальной точки для дихотомии  $B$  можно выбрать любую точку хорды, в которой встреча возможна.



#### Различные задачи

##### Задача 5. “Куда идем мы с Пятачком?”

Кировское открытое командное первенство по программированию, 2001 г.

Пятачок и Винни-Пух каждое утро ходят пить чай в гости к Кролику. Естественно, самым коротким путем. К сожалению, однажды Винни-Пуху пришла в голову идея вырыть ловушку для Слонопотама. Самое обидное, что они с Пятачком ее даже вырыли. Поэтому теперь каждое утро, идя в гости к Кролику, они боятся в нее провалиться.

Напишите программу, которая посчитает длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика.

Ловушка для Слонопотама представляет собой яму абсолютно круглой формы. Путь является безопасным, если он не проходит по ловушке (но может проходить по ее границе).

Во входном файле записаны сначала координаты домика Винни-Пуха  $X_B, Y_B$ , затем — координаты домика Кролика  $X_K, Y_K$ , а затем — координаты центра и радиус ловушки  $X_L, Y_L, R_L$ . Все координаты — целые числа из диапазона от  $-32\ 000$  до  $32\ 000$ . Радиус ловушки — натуральное число, не превышающее  $32\ 000$ .

Домики Винни-Пуха и Кролика не могут находиться внутри ловушки, но могут находиться на ее границе.

Выведите в выходной файл одно число — длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика с тремя знаками после точки.

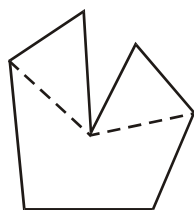
Примеры входного файла	Примеры выходного файла
0 0 0 1 10 10 1	1.000
5 0 0 5 0 0 5	7.854
-5 0 5 0 0 0 3	11.861

*Решение.* Для решения этой задачи необходимо уметь определять взаимное расположение окружности и отрезка (а не прямой!) и правильно вычислять длину дуги окружности, ограниченной двумя заданными точками.

### Задача 6. “Подсветка фонтана”

IX Всероссийская олимпиада по информатике

Плоское дно фонтана описывается замкнутой ломаной линией без самопересечений, причем никакие три вершины ломаной не лежат на одной прямой. Для организации подсветки фонтана между двумя заданными углами (вершинами) по дну проложен гибкий натянутый кабель (см. рисунок). Требуется написать программу, вычисляющую длину этого кабеля.



Исходные данные записаны в файле в следующей последовательности:

- в 1-й строке — число вершин  $N$  ( $N \leq 100$ );
- в каждой из последующих  $N$  строк — пара чисел через пробел, являющихся координатами вершин  $x_1 y_1 x_2 y_2 \dots x_N y_N$  в порядке обхода ломаной против часовой стрелки, где  $1, 2, \dots, N$  — номера вершин;
- в последней строке — номера соединяемых вершин  $k$  и  $l$  ( $1 \leq k < l \leq N$ ).

Координаты вершин являются вещественными числами.

Результат вывести в виде числа. Результат проверяется с точностью до шести значащих цифр. Результирующее число может быть как с фиксированной точкой, так и в нормализованном виде.

Примеры входного файла	Примеры выходного файла
7	7.5
2 0	
5 0	
6 3.5	
5 6	
4 2	
3 7	
0 5	
3 7	

*Решение.* Возможно несколько различных подходов к решению данной задачи. Один из них — поиск кратчайшего пути в графе (см. лекцию 8), в матрице смежности которого записаны расстояния между вершинами границы фонтана, если их можно напрямую соединить шлангом, и  $\infty$ , если этого сделать нельзя. Для построения такой матрицы необходимо уметь проверять наличие пересечения двух отрезков и в случае отсутствия пересечений — местоположение отрезка относительно границы фонтана (внутри или снаружи он находится). В последней подзадаче достаточно определить местонахождение одной из внутренних точек этого отрезка.

Знание различных геометрических формул было необходимо и при решении задачи XIII Всероссийской олимпиады по информатике “Пожар” (см. [7]).

### Литература

1. *Препарата Ф., Шеймос М.* Вычислительная геометрия: введение. М.: Мир, 1989.
2. *Окулов С.М.* Геометрические алгоритмы. “Информатика” № 15, 16, 17/2000.
3. *Окулов С.М.* 100 задач по информатике. Киров: Изд-во ВГПУ, 2000.
4. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы. Построение и анализ. М.: МЦНМО, 2000.
5. *Андреева Е., Фалина И.* Турбо Паскаль в школе. М.: Изд-во Н.Ф. Бочкаревой, 1998.
6. *Станкевич А.С.* Решение задач I Всероссийской командной олимпиады по программированию. “Информатика” № 12/2001.
7. *Андреева Е.В.* Решение задач XIII Всероссийской олимпиады по информатике. “Информатика” № 19/2001.



# Введение в профессию “учитель информатики”

Лекции читает А.Г. Гейн

## Лекция 5. Что за чем, а также зачем преподавать

Каждый солдат должен понимать свой маневр.  
А.В. Суворов. “Наука побеждать”

Если бы я взялся написать книгу “Наука обучать”, то один из ее основных тезисов звучал бы так: “Каждый учитель должен понимать свой маневр”. И разве неправомерно уподобить деятельность учителя с большой военной компанией, цель которой — сокрушить крепости незнания и неумения в умах школьников. Разница, однако, в том, что учитель в этой битве и солдат, и командир. Конечно, есть еще “генералитет”, разрабатывающий стратегию всей компании — концепции, стандарты, учебные программы. Есть “старшие офицеры”, создающие учебники и методические пособия. Но непосредственно “бои” ведут именно учителя. Им поэтому низкий поклон.

Начнем мы с рассмотрения стратегической линии. Обычно она излагается в концепции соответствующей образовательной области. Поэтому и мы процитируем концепцию (точнее, ее проект, опубликованный в “Информатике” № 17/2001, — надо смотреть в будущее, а не жить вчерашним днем) образовательной области “Информатика и информационные технологии”: “Именно поэтому перед системой образования встает сегодня новая проблема — подготовить подрастающее поколение к жизни и профессиональной деятельности в высокоразвитой информационной среде, научить их самостоятельно действовать в этой среде, эффективно использовать ее возможности и защищаться от негативных воздействий”.

Мысль о необходимости готовить подрастающее поколение к жизни и деятельности в высокоразвитой информационной среде на самом деле не так уж и нова. Она не только давно присутствует на страницах различных педагогических изданий, но нашла свое отражение в учебной литературе для школьников, т.е. разрабатывается уже на следующем уровне. В учебнике [4], к примеру, она звучит по крайней мере дважды: “В современном мире роль информатики, средств обработки, накопления и передачи информации неизмеримо возросла. Средства информатики и вычислительной техники сейчас во многом определяют научно-технический потенциал страны, уровень развития ее народного хозяйства, образ жизни и деятельности человека” (с. 5). И далее (с. 15): “Сегодня информатика и вычислительная техника, проникнув во многие сферы 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 образованному человеку в современном обществе”. В учебнике [2] этому вопросу посвящены многие страницы темы 1.1. “От индустриального общества к информационному” и практически вся тема 1.4. “Информатика в жизни общества”<sup>1</sup>.

Формулировка стратегической проблемы еще не указывает ни путей ее решения, ни даже целей, которые должны быть достигнуты, чтобы можно было считать проблему решенной. В чем именно видят авторы концепции подготовленность подрастающего поколения к жизни и профессиональной деятельности в высокоразвитой информационной среде? Вот какой ответ дан в самой концепции:

“Цели общеобразовательного изучения информатики кратко можно сформулировать как формирование *информационно-коммуникационной компетентности выпускников школы*, включая:

- 1) целостное миропонимание и научное мировоззрение, которые основаны на понимании единства основных информационных законов в природе и обществе;
- 2) представления об информационных объектах и их преобразовании с помощью средств информационных технологий, технических и программных средствах, реализующих эти технологии;
- 3) совокупность общеобразовательных и профессиональных знаний и умений, социальных и этических норм поведения людей в информационной среде XXI века”.

Сказать учителю, что, преподавая курс информатики, он должен сформировать у учеников целостное миропонимание и научное мировоззрение, это все равно что сказать солдату, что, взяв очередной рубеж, он должен обеспечить мир во всем мире<sup>2</sup>.

<sup>1</sup> Именно ввиду большого объема этого текста мы воздержимся от его цитирования, предоставляя читателю возможность самостоятельно проанализировать его с точки зрения отражения в нем целевых установок курса. Призыв к такому анализу содержится в приведенном ниже задании 1.

<sup>2</sup> Читатель не мог не обратить внимание на постоянно проводимую мной аналогию между стратегией и тактикой в военном деле и в образовании. Конечно, всякая аналогия условна. Но она позволяет (и заставляет) взглянуть на уже примелькавшиеся формулировки с новой стороны и увидеть, что все совсем не так просто и очевидно, как кажется на первый взгляд. Что касается аналогии, используемой здесь, то хочу напомнить еще не так уж давно употреблявшуюся риторику: борьба за знания, дать бой неуспеваемости и т.п. Может, она и вернется снова? А может, ее никто и не отменял?

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

Формирование целостного миропонимания и научно-мировоззрения — это задача образования (и не только школьного, но и вузовского) в целом, если, конечно, такая задача действительно ставится<sup>3</sup>. А для курса информатики цель в этой части должна формулироваться приземленней — обеспечить овладение основными сведениями об основных информационных закономерностях, присущих живой и неживой природе и обществу.

Напомним, что закономерность — это устойчивая повторяющаяся связь между явлениями и процессами природы или общества. Поэтому, чтобы говорить о закономерностях, надо прежде всего понимать, какие информационные процессы имеют место. Поэтому, хотя в процитированных выше целях нет и слова про информационные процессы, рассказ о них обязателен в любом курсе информатики.

Вот как определяется понятие информационного процесса в школьных учебниках.

В [3], с. 13: “Действия, выполняемые с информацией, называются **информационными процессами**. Есть три типа информационных процессов: **хранение, передача и обработка** информации”.

В [2], с. 7: “**Информационный процесс** — это процесс, в результате которого осуществляются **прием, передача (обмен), преобразование и использование** информации”.

Какому из этих определений должен отдать предпочтение учитель? Ведь выбор формулировки — все равно что выбор оружия: от этого зависит, насколько эффективной окажется атака.

Проанализируем эти определения. Сначала заглянем в толковый словарь русского языка [5]: “Процесс — ход, развитие какого-нибудь явления, последовательная смена состояний в развитии чего-либо”<sup>4</sup>. Можно еще заглянуть в толковый словарь по программированию [1]: “Процесс — 1) хронологически упорядоченная последовательность состояний некоторой системы; 2) исполнение алгоритма, программы”. Конечно, процесс может состоять из отдельных действий (тем самым, в частности, представлять собой исполнение алгоритма, что и отражено во втором толковании [1]), но вряд ли целесообразно отождествлять “действие” и “процесс”, как это сделано в [3]. Что касается определения из [2], то представляется странным считать прием, передачу, преобразование и использование информации *результатом* процесса, а не самим процессом. Может быть, просто заменить в этом определении слова “в результате” какими-нибудь другими? Вот как могло бы выглядеть определение в этом случае: **информационный процесс** — это процесс, в ходе ко-

торого осуществляются прием, передача (обмен), преобразование и использование информации”. Здесь, правда, исчезло (по сравнению с [3]) хранение информации, зато появилось ее использование. Можно, конечно, просто добавить хранение информации в перечисление типов информационных процессов, но где уверенность, что этим все будет исчерпано? Ряд специалистов считает, например, что наряду с традиционно рассматриваемыми процессами получения, хранения, передачи, обработки и использования информации нужно рассматривать процессы генерации и восприятия информации. Чтобы не оказаться в неловком положении, что завтра придется отказываться от определений, данных вчера, хочется иметь определение, не опирающееся на эту типологию. Вот такое определение: **информационный процесс** — это процесс, в ходе которого изменяется содержание информации или форма ее представления.

Уже после того, как дано такое определение, можно обсуждать с учащимися, как именно может меняться содержание или форма представления информации в информационном процессе, и опираться в этом на тот или иной учебник. Более того, при таком подходе можно не опасаться вопроса от какого-либо не в меру любознательного школьника, который скажет: “А вот в другом учебнике, не том, который вы нам рекомендовали, дано другое определение информационного процесса. Оно, что ли, неправильное?”

Конечно, учителю придется воевать тем оружием, которое ему дали. Но подогнать его себе по руке — это его забота и в общем-то обязанность.

На следующем шаге требуется провести анализ понятий получения, передачи, хранения, обработки, использования информации.

Попробуем найти в наших учебниках информатики определение процесса получения информации. Это оказывается не таким простым делом. В них говорится, что “получение и обработка информации являются необходимым условием жизнедеятельности любого организма”, а также, что “для получения (восприятия) информации у животных и человека имеются органы чувств” ([4], с. 5—6); что “с помощью органов чувств люди воспринимают информацию” ([2], с. 7) и т.п. Но о том, что такое получение информации, не сказано нигде. Между тем получение информации можно определить следующим образом.

**Получение информации** — это реализация способности к отражению различных свойств объектов, явлений и процессов в окружающем мире.

Можно было бы, наверное, обойтись и без определения этого понятия. В конечном счете кажется естественным, что все школьники интуитивно представляют себе, что такое получение информации. Но, к сожалению, авторы учебников не демонстрируют необходимое для этого единство в трактовке данного понятия. В [2], с. 7—8, читаем: “Почему осенью опадают листья и вся растительность засыпает на время холодов, а с приходом весны вновь появляются листья, трава, цветы? Почему определенные виды растений в одно

<sup>3</sup> Наша оговорка вовсе не случайна. Научное мировоззрение в отличие от научного знания всегда предполагает противопоставление вере и тому знанию, которое, как считает религия, может быть получено через откровение. Хочу подчеркнуть, что сама по себе информатика, как научное знание, нейтральна к этому вопросу.

<sup>4</sup> В словаре приводятся и два других значения слова “процесс”: развивающаяся болезнь и порядок разбирательства судебных и административных дел. Очевидно, что эти два смысла не имеют отношения к теме нашего рассуждения.

и то же время года? Это тоже результат информационных процессов. Клетка любого растения воспринимает изменения внешней среды (температуру, влажность, время суток) и соответствующим образом реагирует на них". По замыслу авторов получается, что листья, получив информацию, начинают опадать. А в [3] получение информации — это извлечение знаний из сообщений. Что касается информации, получаемой человеком с помощью органов чувств, то авторы этого учебника считают это не получением, а восприятием информации. (Напомним, что, как отмечалось выше, на выделении восприятия в самостоятельный вид информационных процессов настаивают и другие специалисты в области информатики.) Вот и хочется иметь определение, чтобы хоть как-то выбраться из этого разнобоя.

Тем не менее, как бы мы ни понимали слова "получить информацию", это будет информация о чем-то, т.е. объекте, процессе или явлении, и она будет заведомо неполной. Дело в том, что всякий приемник информации, будь то живое существо или автоматическое устройство, в силу конечности тех или иных своих характеристик (скажем, конечное количество клеток мозга или ячеек оперативной памяти) не способен отразить в себе *всю* информацию о рассматриваемом объекте, процессе или явлении ввиду априорной бесконечности этой информации. Ведь любой реальный объект, процесс, явление обладает бесконечным числом свойств и проявлений. Тем самым приемник информации всегда имеет дело не с самим объектом (процессом или явлением), а с соответствующей ему **моделью**. Вот и появилось следующее понятие, принципиально важное для курса информатики. На самом деле еще более важным для курса информатики будет частный случай этого понятия — **информационная модель**.

Неизбежно возникнут и другие понятия, за ними последуют еще, и т.д. В какой-то момент в этом понятийном облаке будут вскрыты связи. Фиксация обнаруженных связей в тех или иных формулировках и даст те самые информационные законы, о которых говорится в концепции. Но целью данной лекции не может быть развертывание всей системы понятий и связей между ними — это совсем иная тема, требующая отнюдь не одной лекции. Мы же, как и в предыдущих лекциях, лишь демонстрируем (притом в весьма ограниченных объемах), какую работу необходимо совершить учителю, чтобы точно реализовать цели обучения.

Помня, что нельзя объять необъятное, мы на этом остановимся в своих рассуждениях.

### Вопросы и задания

- В лекции указано, что в учебнике [4] обоснование для школьников целей изучения информатики приводится лишь в двух абзацах, в то время как в учебнике [2] тому же вопросу отведено не менее 10 страниц. Чем, по вашему мнению, можно объяснить такое различие во внимании к данному вопросу?
- В учебнике [4], с. 5, дано следующее определение: "Все процессы, связанные с определенными операциями над информацией, будем называть **информационными процессами**". Проанализируйте его в сравнении с другими определениями информационных процессов, которые даны в лекции.
- В учебниках, рассматривавшихся в данной лекции, нет определения процессу передачи информации. Вот что об этом процессе в них говорится. В [3], с. 12: "Человеку постоянно приходится участвовать в процессе передачи информации. Передача может происходить при непосредственном разговоре между людьми, через переписку, с помощью технических средств связи: телефона, радио, телевидения. Такие средства связи называются "каналами передачи информации". В [4], с. 6—8: "Первым средством обмена информации между людьми стала человеческая речь... Человечество научилось использовать для передачи и сохранения информации рисунки, чертежи, схемы, а впоследствии — фотографии, телевизионные изображения и т.д. По мере развития общества, появления технических средств связи все большее внимание науки стали привлекать процессы передачи информации. До XIX века основным носителем информации была письменность, а средством связи — почта. Положение принципиально изменилось с изобретением телеграфа, телефона, а затем радио, телевидения и других современных видов связи". Проанализируйте, что общего и в чем различается изложение вопроса о передаче информации в этих учебниках. Какое определение понятию "передача информации" дали бы вы? Нужно ли давать определение этому понятию? С какими новыми понятиями должны, на ваш взгляд, познакомиться учащиеся, осваивая понятие передачи информации?
- Какие, на ваш взгляд, вопросы должен поставить перед собой учитель, прежде чем начать изучать с учащимися понятие "обработка информации"? Чем и как он может воспользоваться, чтобы получить ответы на свои вопросы?

### Литература

- Заморин А.П., Марков А.С. Толковый словарь по вычислительной технике и программированию. Основные термины. М.: Русский язык, 1988, 221 с.
- Информатика: Учебник для 10—11-х классов средней школы / Под ред. Н.В. Макаровой. СПб.: Питер-Ком, 1999, 304 с.
- Информатика. Базовый курс для 7—9-х классов / Под ред. Е.К. Хеннера. М.: Лаборатория базовых знаний, 2000, 384 с.
- Кузнецов А.А., Апатова Н.В. Основы информатики. 8—9-е классы: Учебник для общеобразовательных учебных заведений. М.: Дрофа, 1999, 176 с.
- Ожегов С.И. Словарь русского языка. М.: Русский язык, 1983, 816 с.

# Примерные ответы на примерные билеты

Е.А. Еремин, А.П. Шестаков,  
г. Пермь

Продолжение. См. № 9, 10, 11, 13/2002

## Билет № 9

**1. Текстовый редактор. Назначение и основные возможности.**

**2. Разработка алгоритма или программы, формирующей на экране рисунок.**

\* \* \*

**1. Текстовый редактор. Назначение и основные возможности.**

Текстовые редакторы (процессоры) относятся к программному обеспечению общего назначения, они предназначены для создания, редактирования, форматирования, сохранения во внешней памяти и печати текстовых документов. Обычно текстовыми редакторами принято называть программы, выполняющие простейшие операции по редактированию текста, а процессорами — программы, обладающие расширенными по сравнению с редакторами средствами для компьютерной обработки текста. Современные текстовые процессоры по своим функциональным возможностям приближаются к *издательским системам* — пакетам программ, предназначенным для верстки газет, журналов, книг.

Основные функции текстовых процессоров:

- создание документов;
- редактирование документов (перемещение по тексту, вставка и замена символов, выделение, удаление, перемещение и копирование фрагментов, поиск и замена фрагментов текста, отмена команд; вставка фрагментов других документов или целых документов и т.д.);
- сохранение документов во внешней памяти (на дисках) и чтение из внешней памяти в оперативную;
- форматирование документов, т.е. выполнение преобразований, изменяющих форму документа (оформление отдельных символов и абзацев, страниц, документа в целом — изменение длины строки, межстрочного расстояния, выравнивания текста, изменение шрифта, его размера, применение различного начертания шрифтов и т.д.);
- печать документов (или их некоторой части);
- составление оглавлений и указателей в документе;
- создание и форматирование таблиц;
- внедрение в документ рисунков, формул и др.;
- проверка пунктуации и орфографии.

Основными элементами текстового документа являются:

- *символ* — минимальная единица текстовой информации. Каждый символ имеет свой код (как и любая информация, символы представляются в памяти ЭВМ в виде двоичного кода). Соответствие между кодом и символом устанавливается специальными коди-

ровочными таблицами (code page — CP), причем в разных кодировочных таблицах одному и тому же коду ставятся в соответствие разные символы. В MS-DOS для кодирования русского алфавита используется CP-866, в Windows — CP-1251. Текстовые процессоры (например, MS Word) чаще всего позволяют перейти от одной кодировки к другой;

- *слово* — произвольная последовательность букв и цифр, ограниченная с двух сторон служебными символами. В качестве служебных символов могут выступать пробел, запятая, скобки и т.д.;

- *строка* — произвольная последовательность символов между левой и правой границами абзаца;

- *предложение* — произвольная последовательность слов, завершающаяся точкой;

- *абзац* — в литературе часть текста, представляющая собой смысловое единство, окончание которой служит естественной паузой для перехода к новой мысли, и обычно выделяемая отступом в первой строке. В документе, оформляемом с помощью компьютера, абзац обычно завершается специальным символом конца абзаца (в DOS и Windows это “возврат каретки” — символ с кодом 13). Допускаются пустые абзацы;

- *страницу* составляют строки и абзацы, таблицы и внедренные в документ объекты;

- наиболее крупной единицей является собственно *документ*, где все составляющие его абзацы определенным образом структурированы, снабжены при необходимости заголовками, выстроена иерархия структурных разделов.

Обычно текстовые процессоры предусматривают *две основные операции изменения формата документа*:

- форматирование произвольной последовательности символов (от одного до любого количества, чаще всего эта последовательность предварительно выделяется);

- форматирование абзацев.

При форматировании символов можно изменить:

- шрифт;
- начертание шрифта (полужирный, курсив, подчеркнутый);
- размер шрифта;
- межсимвольный интервал;
- применить к символам эффекты (нижний, верхний индекс, малые строчные буквы и т.д.).

При форматировании абзацев можно изменить:

- способ выравнивания строк абзаца (влево, вправо, по центру, по ширине);
- отступ в красной строке абзаца;
- ширину и положение абзаца на странице;

- межстрочное расстояние (интерлиньяж) и расстояние между соседними абзацами;
- создать специальные абзацы (маркированные или нумерованные списки и т.д.).

Наиболее распространенные форматы текстовых файлов, поддерживаемые современными текстовыми процессорами (типа MS Word):

- **текстовый (TXT)**. Сохраняет текст без форматирования, в текст вставляются только управляющие символы конца абзаца. Обычно этот формат применяют для хранения документов, которые должны быть прочитаны приложениями, работающими в разных операционных системах (Windows, MacOS, Unix и т.д.);
- **Rich Text Format (RTF)**. Универсальный формат, который сохраняет все форматирование. Преобразует управляющие коды в текстовые команды, которые могут быть прочитаны и интерпретированы многими приложениями (в т.ч. и в разных ОС). В связи с этим информационный объем файла при этом достаточно велик по сравнению с другими форматами;
- **текст DOS (TXT)**. Текст в кодировке ASCII, отформатированный в ОС DOS;
- **документ Word (DOC)**. Оригинальный формат документов MS Word;
- **документ HTML (HTML)**. Формат, используемый для хранения web-страниц. Содержит коды разметки языка гипертекста.

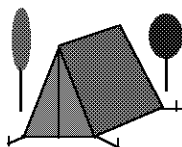
## Литература

1. Гейн А.Г., Сенокосов А.И., Шолохович В.Ф. Информатика: 7—9-е классы. Учебник для общеобразовательных учебных заведений. М.: Дрофа, 1998.
2. Каймин В.А., Щеголев А.Г., Ерохина Е.А., Федюшин Д.П. Основы информатики и вычислительной техники: Пробный учебник для 10—11-х классов средней школы. М.: Просвещение, 1989.
3. Кушницренко А.Г., Лебедев Г.В., Сворень Р.А. Основы информатики и вычислительной техники: Учебник для средних учебных заведений. М.: Просвещение, 1993.
4. Семакин И., Залогова Л., Русаков С., Шестакова Л. Информатика: учебник по базовому курсу. М.: Лаборатория базовых знаний, 1998.
5. Угринович Н. Информатика и информационные технологии. Учебное пособие для общеобразовательных учреждений. М.: БИНОМ, 2001, 464 с. (Глава 11. Технология обработки текстовой информации, с. 293—310.)
6. Информатика. 7—8-е классы /Под ред. Н.В. Макаровой. СПб.: ПитерКом, 1999, 368 с.
7. Шафрин Ю.А. Информационные технологии. М.: Лаборатория базовых знаний, 1998, 704 с.

\* \* \*

## 2. Разработка алгоритма или программы, формирующей на экране рисунок.

Задание: разработать программу, формирующую на экране ЭВМ приведенное ниже изображение:



Построение рисунка рекомендуется начать с проектирования на бумаге размещения этого изображения в экранной системе координат. На экране изображение можно формировать в *абсолютных* и *относительных* координатах. Абсолютные координаты предполагают задание некоторых числовых значений, характеризующих местоположение изображения, а относительные позволяют вести отсчет координат относительно некоторой заранее выбранной точки (в примерах ниже это будут координаты правого нижнего угла экрана). Абсолютные координаты привязывают изображение к вполне определенному месту на экране, в то время как изменение точки отсчета в относительных координатах позволяет вывести изображение в любом месте экрана.

При отображении рисунка на экране компьютера проходят следующие этапы:

- 1) подключение графической библиотеки (модуля);
- 2) инициализация графического режима;
- 3) вывод изображения;
- 4) задержка, необходимая для просмотра изображения;
- 5) закрытие графического режима и завершение работы программы.

Программа на языке Паскаль:

```

Program Risunok_Malchiski;
Uses Graph, Crt;
Var x, y : Integer;
    GraphDriver, GraphMode : Integer;
Begin
  {для автоопределения графического режима}
  GraphDriver := Detect;
  GraphMode := 0;
  {инициализация графики}
  InitGraph(GraphDriver, GraphMode, '');
  x := GetMaxX - 3 * (GetMaxX div 4);
  y := GetMaxY - 3 * (GetMaxY div 4);
  SetBkColor(White); SetColor(DarkGray);
  {дерево слева от палатки}
  ellipse(x + 40, y + 20, 0, 360, 10, 30);
  setfillstyle(1, DarkGray);
  Bar(x + 40, y + 50, x + 42, y + 90);
  {дерево справа от палатки}
  ellipse(x + 380, y + 0, 360, 15, 20);
  Bar(x + 380, y + 20, x + 382, y + 60);
  {закраска крон деревьев}
  setfillstyle(1, Green);
  floodfill(x + 40, y + 20, DarkGray);
  floodfill(x + 380, y, DarkGray);
  {палатка}
  Line(x + 120, y + 20, x + 120, y + 120);
  Line(x + 40, y + 120, x + 200, y + 120);
  Line(x + 120, y + 20, x + 40, y + 120);
  Line(x + 120, y + 20, x + 200, y + 120);
  Line(x + 120, y + 20, x + 250, y - 20);
  Line(x + 200, y + 120, x + 330, y + 80);
  Line(x + 250, y - 20, x + 330, y + 80);
  Line(x + 40, y + 120, x + 20, y + 130);
  Line(x + 20, y + 120, x + 20, y + 135);
  Line(x + 200, y + 120, x + 220, y + 130);
  Line(x + 220, y + 120, x + 220, y + 135);
  Line(x + 330, y + 80, x + 345, y + 85);
  Line(x + 345, y + 75, x + 345, y + 90);

```

```
{закраска палатки}
setfillstyle(1, LightCyan);
floodfill(x + 121, y + 30, DarkGray);
floodfill(x + 119, y + 30, DarkGray);
floodfill(x + 200, y + 30, DarkGray);
Repeat Until KeyPressed; {задержка}
CloseGraph {закрытие графического режима}
End.
```

Программа на языке Си:

```
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
int x, y, GraphDriver, GraphMode;
void main()
/*для автоопределения графического режима*/
{ GraphDriver = DETECT;
  GraphMode = 0;
  /*инициализация графики*/
  initgraph(&GraphDriver, &GraphMode, "");
  x = getmaxx() - 3 * (getmaxx() / 4);
  y = getmaxy() - 3 * (getmaxy() / 4);
  setbkcolor(WHITE); setcolor(DARKGRAY);
  /*дерево слева от палатки*/
  ellipse(x + 40, y + 20, 0, 360, 10, 30);
  setfillstyle(1, DARKGRAY);
  bar(x + 40, y + 50, x + 42, y + 90);
```

```
/*дерево справа от палатки*/
ellipse(x + 380, y, 0, 360, 15, 20);
bar(x + 380, y + 20, x + 382, y + 60);
/*закраска крон деревьев*/
setfillstyle(1, GREEN);
floodfill(x + 40, y + 20, DARKGRAY);
floodfill(x + 380, y, DARKGRAY);
/*палатка*/
line(x + 120, y + 20, x + 120, y + 120);
line(x + 40, y + 120, x + 200, y + 120);
line(x + 120, y + 20, x + 40, y + 120);
line(x + 120, y + 20, x + 200, y + 120);
line(x + 120, y + 20, x + 250, y - 20);
line(x + 200, y + 120, x + 330, y + 80);
line(x + 250, y - 20, x + 330, y + 80);
line(x + 40, y + 120, x + 20, y + 130);
line(x + 20, y + 120, x + 20, y + 135);
line(x + 200, y + 120, x + 220, y + 130);
line(x + 220, y + 120, x + 220, y + 135);
line(x + 330, y + 80, x + 345, y + 85);
line(x + 345, y + 75, x + 345, y + 90);
/*закраска палатки*/
setfillstyle(1, LIGHTCYAN);
floodfill(x + 121, y + 30, DARKGRAY);
floodfill(x + 119, y + 30, DARKGRAY);
floodfill(x + 200, y + 30, DARKGRAY);
getch();
/* закрытие графического режима */
closegraph();
}
```

## Билет № 10

**1. Графический редактор. Назначение и основные возможности.**

**2. Разработка алгоритма или программы, содержащей команду повторения (оператор цикла).**

\*\*\*

**1. Графический редактор. Назначение и основные возможности.**

Каждому известно, что графическое представление информации является для человека наиболее наглядным. Индикатор в виде столбика или кругового сектора во многих случаях гораздо удобнее показаний самого точного цифрового прибора. График существенно понятнее столбика цифр, карта позволяет емко и точно описать любую местность, а электрическую схему телевизора или ЭВМ никто даже не пытается изобразить как-нибудь по-другому. Рисунок или фотография дают самое компактное представление о внешнем виде любого объекта, поэтому они широко используются в книгах, особенно справочниках и энциклопедиях. Число подобных примеров можно продолжать неограниченно.

Современный компьютер, естественно, способен вводить, хранить, обрабатывать и выводить графическую информацию. Для этой цели он использует целый ряд технических устройств. Дисплей и печатающее устройство, возникшие исторически для вывода текстов, давно уже стали графическими и могут работать с цветом. Устройства для ввода изображений, такие, как сканер, сейчас также широко распространены. Наконец, рост быстродействия процессоров, наличие больших объемов ОЗУ и

появление все новых носителей информации большой емкости, например дисков CD-ROM, также способствуют существенному прогрессу в компьютерной графике. Принципы работы компьютерного оборудования с графической информацией довольно интересны сами по себе, но в рамках данного вопроса мы их рассматривать не будем.

Мы знаем, что для реализации обработки любой информации, помимо аппаратных средств, обязательно требуется *программное обеспечение*. Графическая информация, разумеется, не исключение. Программа для создания и изменения графических изображений, а также их сохранения на внешних носителях и печати на бумагу называется **графическим редактором**. В силу важности машинной графики для современного пользователя несложный графический редактор является сейчас обязательной сервисной программой в большинстве операционных систем.

Существует огромное количество графических редакторов для пользователей самого разнообразного уровня — от неопытных новичков до профессиональных художников. Возможности их, конечно, существенно отличаются, однако имеются некоторые общие свойства и приемы работы; именно они и представляют интерес для нашего обсуждения.

При выборе конкретного графического редактора следует прежде всего иметь в виду, что в современных компьютерах существует два принципиально различных способа хранения изображений: **растровый** и **векторный**. При растровом методе сохраняется информация о цвете каждой точки рисунка или фотографии; он очень точно

передает сложные изображения, но зато требует много памяти и плохо масштабируется (при попытке изменить размеры рисунка его контуры и цветопередача заметно искажаются). Векторная графика, строящая изображения из простых базовых элементов типа прямых, эллипсов и т.д., напротив, создает хотя и не очень сложные, но компактные рисунки, прекрасно поддающиеся изменению размера. Не случайно библиотека картинок для оформления текстов (так называемый *clipart*), входящая в состав редактора MS Word, хранится именно в векторном формате. Более подробно о достоинствах и недостатках растровой и векторной графики можно почитать, например, в статье [1] из списка дополнительной литературы.

В соответствии с описанным разделением имеется два типа редакторов. Для векторной графики наиболее известны Corel Draw! и Adobe Illustrator, а ассортимент более простых в обращении редакторов растровых изображений значительно шире — от простейшего, входящего в состав Windows, редактора Paint до профессионального Adobe PhotoShop.

Мы остановимся на работе с растровым графическим редактором, поэтому подробнее поговорим о принципах построения изображений именно для этой категории редакторов. Термин **растр** возник довольно давно (в конце XIX века) и означает разложение изображения на отдельные точки с помощью специальной сетки. Метод создания изображения из отдельных точек появился задолго до изобретения компьютеров. Вспомним, например, мозаику или вышивание; формирование растра на экране лежит также в основе работы используемого нами каждодневно телевизора.

Рисунок с точки зрения рассматриваемого редактора состоит из отдельных точек (элементов) — **пикселей**. Сразу подчеркнем, что пиксель — это *логическое* понятие, оно не совпадает с физической точкой монитора. Для тех, кому данный факт кажется неочевидным, напомним, что на одном и том же мониторе можно установить несколько различных размеров экрана, например, 640 × 480 или 1024 × 768 пикселей. Следовательно, чаще всего пиксель есть объединение нескольких физических точек экрана, и только в частном случае каждый элемент изображения совпадает с единственной точкой на мониторе. Все пиксели характеризуются своим положением в изображении (иными словами, двумя координатами) и цветом. Поскольку растровая сетка однозначно определяет последовательность обхода точек рисунка, специально сохранять координаты нет необходимости, достаточно запомнить последовательность цветов всех точек.

Теперь естественно поговорить о представлении цветов в графическом изображении. В мониторах изображение формируется по так называемому “RGB-методу” (Red — красный, Green — зеленый, Blue — голубой). Все остальные цвета формируются путем смешения трех перечисленных базовых, например, смешение красного и зеленого дает желтый, а все три в одинаковой пропорции порождают белый.

**Примечание.** Следует знать, что при печати графических изображений на бумагу цветовую модель RGB, строго говоря,

использовать нельзя. Дело в том, что монитор *излучает* изображение, а на бумаге мы рассматриваем его *в отраженном свете*. Поэтому все базовые цвета при печати надо заменять на дополнительные: Cyan — голубой, Magenta — пурпурный и Yellow — желтый. В такой системе цветов белый получается при нулевой интенсивности компонент (в самом деле, если совсем не наносить краску на чистый лист бумаги, то какой будет цвет?), а смешение всех базовых цветов теоретически должно давать черный. На практике чисто черный цвет в полиграфии “смешать” очень трудно, а он в бумажных документах является основным! Поэтому модель CMY приходится дополнять чистым черным (black) и, взяв от него последнюю букву, чтобы не путать с голубым, назвать полученную модель CMYK. Подробнее об этих и некоторых других используемых на практике цветовых системах советуем почитать в статье [2] из списка дополнительной литературы\*.

Важной характеристикой растрового изображения является **количество цветов**. Чем больше количество цветов, тем лучше цветопередача, но тем больше занимает места полученный рисунок.

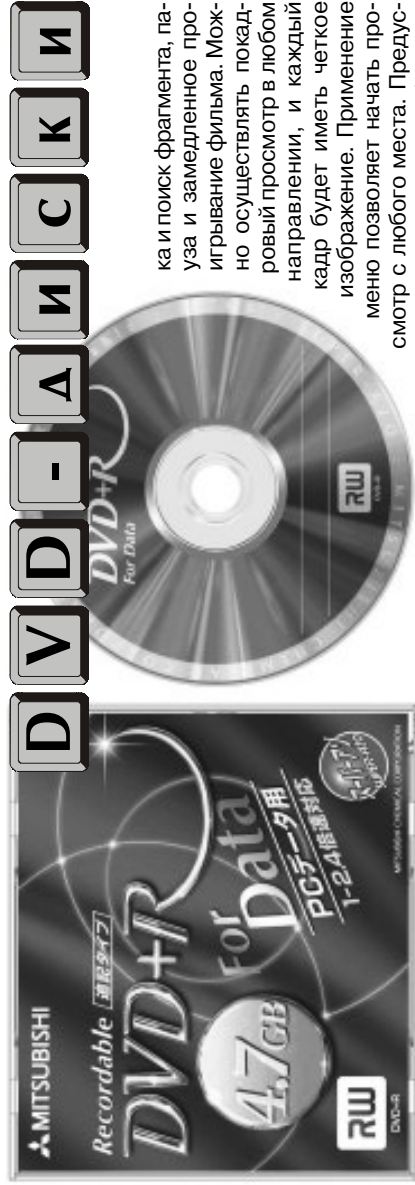
**Примечание.** Для преодоления этого противоречия в современной машинной графике часто используют **палитру** — специальную таблицу, в каждой строке которой записывается содержание трех базовых компонентов для данного цвета. При сохранении информации о пикселе берется не сам цвет, а его номер в палитре, что при небольшом числе цветов заметно короче и позволяет сократить объем графической информации в несколько раз. Вы спросите: а зачем мне знать все эти “внутренние” детали, если я их все равно никогда не увижу? Не торопитесь с выводами, лучше попробуйте совместить два рисунка из разных библиотек (не две фотографии, а именно два рисунка с небольшим числом цветов!) в один и сразу увидите: палитры их скорее всего не совпадут, и цвета исказятся. К сожалению, продемонстрировать это в черно-белом варианте слишком затруднительно, поэтому, как и ранее, соответствующие файлы “сопровождения” можно взять на сайте <http://info-bilet.narod.ru>.

Вооруженные пониманием принципов создания растровых изображений, мы можем наконец загрузить графический редактор и начать рисовать. Для определенности возьмем графический редактор Paint, входящий в состав операционной системы Windows. Данный редактор прост в использовании и позволяет создавать различные схемы и диаграммы, не слишком сложные черно-белые и цветные рисунки, а также редактировать уже существующие графические файлы, например, наносить подписи на фотографии или ретушировать изображение после сканера.

**Примечание.** В практической деятельности следует обязательно учитывать, что Paint не является профессиональным редактором, а значит, многие операции делает “не лучшим образом”. Например, если сравнить один и тот же рисунок для web-странички, сохраненный из Paint и из Adobe PhotoShop, то мы с удивлением обнаружим, что объем полученных файлов может отличаться вдвое. А для названной цели размер является очень существенным параметром!

Итак, редактор запущен. Обратим наше внимание на экран (поскольку Paint есть в *каждом* компьютере с ОС Windows, приводить здесь большой рисунок с общим

\*Короткое и понятное изложение этого вопроса вы также можете найти в “Информатике” № 24/99, с. 11—19. — **Прим. ред.**



изображение, чем лазерный диск (LD), и более насыщенный звук, чем аудио- и компакт-диски.

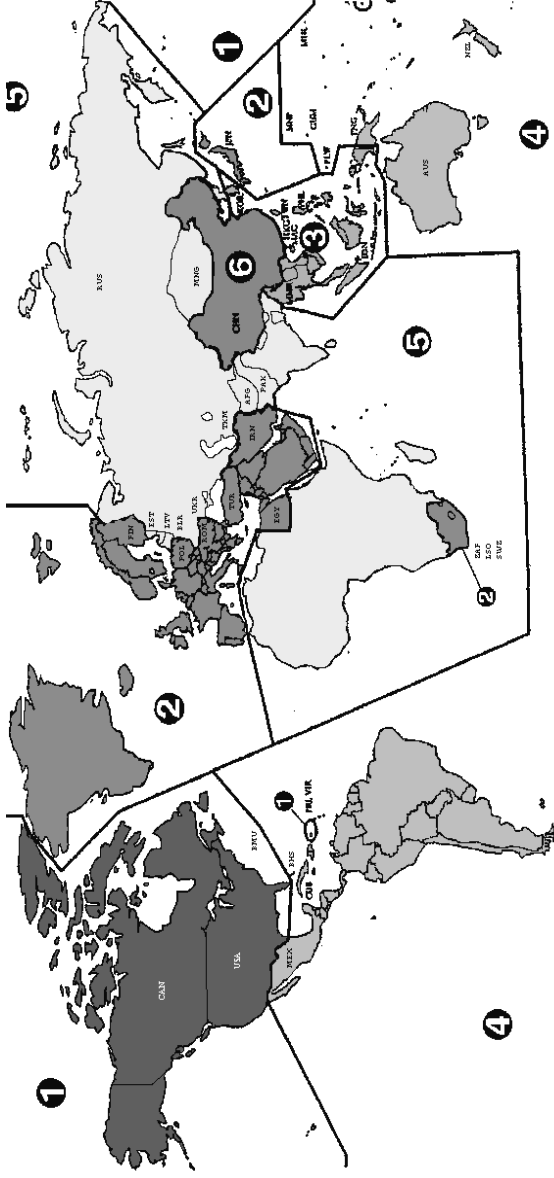
Существуют устройства DVD-R и DVD-RAM, которые позволяют хранить данные на специальных записываемых или перезаписываемых дисках DVD. Устройства DVD-RAM предназначены для многократной записи. Устройства DVD-R позволяют сделать лишь однократную запись. К сожалению, пока стандартные DVD-R и DVD-RAM-диски могут хранить лишь 3,95 Гб и 2,58 Гб данных на каждой стороне соответственно, взамен 4,7 Гб предлагаемых на одной стороне обычных дисков DVD.

Наученные горьким опытом, разработчики DVD заранее заложили в стандарт возможность защиты авторских прав. Как говорилось в одном журнале, "большинство людей в России не испытывали и никогда не будут испытывать угрызений совести при использовании пиратской продукции".

Поэтому всю территорию Земли разработчики разделили на несколько обширных регионов, каждому из которых соответствует свой уникальный код. Этот код закладывается в DVD-проигрыватель согласно тому региону, в котором он должен быть продан, после чего проигрыватель может воспроизводить лишь те DVD-диски, которые "разрешены" к использованию в данном регионе. Так, проигрыватель, купленный в России, может отказаться воспроизводить диски, привезенные из Америки или Японии.

Регионы распределены следующим образом:

- Территория США и Канады;
- Европа, Япония, Южная Африка и Ближний Восток;
- Юго-Восточная и Восточная Азия;
- Австралия, Новая Зеландия, Океания, Латинская Америка, страны Карибского бассейна;
- Африка, Индия, Россия и страны СНГ;
- Китай;
- Специальный экстерриториальный регион (для использования на международном транспорте — самолетах и океанских лайнерах).



ка и поиск фрагмента, пауза и замедленное проигрывание фильма. Можно осуществлять кадровый просмотр в любом направлении, и каждый кадр будет иметь четкое изображение. Применение меню позволяет начать просмотр с любого места. Предусмотрена и специальная функция

контроля, позволяющая родителям закрывать доступ детям к некоторым специфичным фрагментам видеофильма, например, к сценам с насилием. Но самым фантастическим является то, что большая емкость DVD-дисков позволяет применять многопоточные технологии передачи данных. Так, программы на диске могут иметь аудиоперевод на 8 языков и/или 32-язычные субтитры. Например, вы можете наслаждаться просмотром программ на различных языках или даже продублированных на разных языках, причем перевод может быть и в виде субтитров.

Поставщики фильмов на DVD предлагают типичные потрассующий звук, записанный по стандарту Dolby Digital (AC-3 Digital Sound), который обеспечивает пять + один звуковых каналов.

Если вы были в современном кинотеатре или видели рекламу о звуке домашнего кинотеатра, то можете представить, о чем идет речь. В большинстве современных фильмов звуковой ряд записан в стандарте AC-3 5.1 channels, т.е. обеспечиваются пять раздельных (дискретных) каналов и один общий низкочастотный канал. В отличие от стандарта Dolby ProLogic, многоканальный стандарт записи звука AC-3 рассчитан на то, что перед зри-

телем расположены три колонки (левая, правая и центральная), за его спиной расположено еще два твитера (левый и правый), а в произвольном месте расположен сабвуфер (низкочастотная колонка, которую, впрочем, обычно тоже располагают впереди под центральной колонкой).

Запись звука по стандарту AC-3 дает возможность создателям звукового ряда для фильмов пользоваться различными различными студийными спецэффектами добиваться того, что у зрителя создается полное ощущение реальности происходящего на экране действия.

Дискретная сущность звука Dolby Digital обеспечивает дополнительную четкость звучания (особенно важную при диалогах) и эффект объемного пространства. Звук, записанный по стандарту AC-3, может распространяться по помещению в произвольных направлениях, в результате чего и достигается эффект присутствия. Как дополнительная возможность, на DVD можно записывать 16-, 20- или 24-битный стереозвук с качеством CD и частотой 48 КГц или 96 КГц по стандарту Dolby ProLogic.

В перспективе появляется возможность выбора различных сюжетных линий одного и того же фильма и различных вариантов финалов. Кроме того, поставщики фильмов на DVD могут создавать программы с параллельными видеопотоками, т.е. одни и те же сцены можно просматривать с разных углов зрения, создавая для режиссера невиданные ранее возможности для творчества. При просмотре специально созданных фильмов можно, например, выбрать любое из 9 возможных положений камеры, чтобы с разных сторон посмотреть на одну и ту же сцену.

Даже простой футбольный матч в режиме параллельных видеопотоков записывается с различных точек, позволяя наслаждаться одним и тем же моментом со стороны трибун, со стороны вратаря или с использованием крупного плана. Естественно, все это должно быть заложено в специально сделанном интерактивном DVD-диске.

DVD разработывался как единый цифровой формат, нацеленный на последующее вытеснение аудио- и компакт-дисков, видеокассет, лазерных дисков, CD-ROM и даже картриджей для видеоигр. DVD обеспечивает более четкое и качественное

прогресс в компьютерной индустрии не просто не стоит на месте. Он набрал совершенно головокружительные темпы, и пока не так давно их замедляла. Если бы последние 30 лет так же развивалась авиация, то "Боинг-727" стоил бы сейчас меньше ста долларов, а полет вокруг земного шара обошелся бы менее 10 у.е.

Но на фоне сверхбыстродействующих процессоров, многогигабайтных винчестеров, скоростных CD-ROM все еще остаются два "компьютерных dinosaurs", не меняющихся вот уже третий десяток лет.

Про первого из них мы уже рассказывали — это накопители на гибких дисках. Второй — компакт-диски, стандарт которых был предложен и реализован фирмой Sony в 1980 году. Предпринималось довольно много попыток улучшить и даже заменить технологию производства компакт-дисков, но все они упирались в отсутствие согласия между производителями, старавшимися выработать единый стандарт для нового носителя цифровой информации. И вот наконец-то в 1995 году состоялась встреча миру новой технологии — Объединенного формата DVD, — ознаменовавшей собой взятие очередной вершины компьютерной индустрии.

Это — результат совместной работы компаний Toshiba, Matsushita, Philips, Sony и других. Первоначально существовали два стандарта второго поколения DVD. Формат MMCD предлагался Sony, Philips и другими. Конкурирующий формат SD был предложен фирмами Toshiba, Time Warner и некоторыми другими. Группа компьютерных компаний во главе с IBM настаивала, чтобы все компании, предлагающие стандарты DVD, согласились с общим единым стандартом. И здравый смысл взял-таки верх.

DVD призвано перевернуть представления неискушенного человека о хранении информации и ее распространении. Аббревиатура DVD (*Digital Versatile Disk*) расшифровывается как "Цифровой Универсальный (или Многоцелевой) Диск".

DVD-диски используются для хранения компьютерных программ и данных, а также оцифрованных полнометражных фильмов и высококачественного звука. Так же, как обычные компакт-диски подразделяются на Audio-CD и CD-ROM, DVD-диски могут быть записаны в форматах DVD-Video (обычно их и называют просто DVD) и DVD-ROM; для просмотра DVD-Video выгружаются специальные проигрыватели, а диски DVD-ROM читаются специальными компьютерными DVD-приводами, подобными привычным нам CD-ROM.

Важно понять различие между DVD-Video и DVD-ROM. DVD-Video (их обычно и называют DVD) содержит видеопрограммы и проигрывается в DVD-плеере, подключенном к телевизору. DVD-ROM содержит компьютерные данные и читается дисководом DVD-ROM, подключаемым к компьютеру. Их различие подобно различию между аудио-CD и CD-ROM. DVD-ROM также включает такие разновидности, как однократно записываемые (DVD-R) и диски многократной записи (DVD-RAM). Компьютеры с дисковыми DVD-ROM смогут также воспроизводить DVD-диски. Имеется также DVD-Audio звуковой формат.

Внешне диски DVD выглядят, как обычные диски CD-ROM. Однако возможностей у DVD гораздо больше. Диски DVD могут хранить значительно больше данных, по сравнению с обычным CD-ROM. Имея физические размеры и внешний вид, как у обычного компакт-диска или CD-ROM, стандартный однослойный, односторонний диск DVD может хранить 4,7 гигабайта информации. Но это не предел — DVD могут изготавливаться по двухслойному стандарту, который позволяет увеличить емкость хранения на одной стороне данных до 8,5 Гб. Кроме этого, диски DVD могут быть двусторонними, что увеличивает емкость одного диска до 17 Гб.

Громадные объемы данных, умещающиеся на DVD-дисках, в очередной раз подтвердили истину о переходе количества в качество. Теперь становятся реальными самые смелые мечты о домашнем просмотре фильмов с качеством изображения онлайн-студийной аппаратуры.

Диск обеспечивает около 2 часов высококачественного цифрового видео (более 8 — на двустороннем двухслойном диске) и поддержку широкоэкранных фильмов и телефильмов как на широкоэкранных, так и на стандартных телевизорах (иными словами, имеется возможность выбора либо стандартного масштаба 4:3, либо широкоэкранный — 16:9).

Кроме того, DVD-проигрыватели имеют дополнительные возможности, такие, как сканирование дис-

Разумеется, если сам DVD-диск не несет на себе регионального кода, он с легкостью работает на любом DVD-проигрывателе. С другой стороны, сами DVD-проигрыватели можно перенастроить таким образом, чтобы они либо обнуляли свой региональный код, что, правда, не дает стопроцентной гарантии использования закодированного диска, либо получили возможность оперативного переключения своего кода в зависимости от диска. В один проигрыватель эта возможность вкладывается самим производителем (с ограничением на количество таких переключений), в другие — нет, что уже привело к появлению пиратского рынка по их перенастройке.

В случае же DVD-ROM региональные коды не применяются. Компьютерный DVD-привод проигрывает код лишь в том случае, если с него проигрывается видео DVD-диск. Для аудио DVD-дисков региональные коды не применяются вообще. И так, как уже было сказано выше, технология DVD допускает 4 типа дисков:

- односторонний, однослойный — 4,7 Гб;
- односторонний, двухслойный — 8,5 Гб;
- двусторонний, однослойный — 9,4 Гб;
- двусторонний, двухслойный — 17 Гб.

**Совместимость DVD и CD:**

Все DVD-ROM-дисководы могут читать CD-ROM. Само собой, DVD-ROM-диски не читаются дисковыми CD-ROM.

Все DVD-плееры и дисководы могут читать звуковые компакт-диски. Этого не требует спецификация DVD, но все изготовители резонно решают, что их DVD-устройство вряд ли будут пользоваться популярностью без этой функции.

**Недостатки DVD:**

- Имеют встроенную защиту от копирования и региональную блокировку, что в условиях нашей страны требует специальной доработки DVD-устройства.
- Некоторые DVD-плееры и дисководы не могут читать CD-R- и тем более CD-RW-диски.
- Относительно высокая стоимость.

**Использованные источники информации**

- http://people.kstu.edu.ru/CSN/DVD/pages/
- http://www.stravt.ru/dvd
- http://orion7.dp.ua/dvd\_faqs/index.htm





Продолжение. См. с. 12–15

видом его окна нам кажется нецелесообразным). Прежде всего бросается в глаза белый прямоугольник в центральной части окна — это место под будущий рисунок, своеобразная электронная версия чистого листа бумаги. Размеры листа легко регулируются как с помощью мыши (за выделенные угловые точки или середины его сторон), так и с помощью основного меню, о котором речь пойдет чуть позже.

Взаимодействие пользователя с редактором происходит с помощью стандартного графического интерфейса, принятого в среде Windows. Конечно, для графического редактора интерфейс несколько расширен, но еще раз подчеркнем, что рассматриваемое программное обеспечение не есть нечто абсолютно уникальное, живущее по исключительным законам.



В верхней части окна расположено **главное меню**, содержащее традиционные пункты “Файл”, “Правка”, “Вид” и “Справка”, которые дополнены двумя уникальными для графического редактора пунктами “Рисунок” и “Палитра”. Пункт меню “Файл” стандартно обеспечивает чтение и сохранение файлов, а также операции, связанные с печатью результатов работы (в нашем случае рисунка) на бумагу. Обязательно присутствующий в любой “уважающей себя” программе пункт “Правка” служит для сохранения информации (у нас графической) в буфере обмена и извлечения ее в требуемую область окна. Пункт “Вид” включает и выключает отображение на экране служебной информации. Уникальный для графического редактора пункт “Рисунок” вызывает преобразование графического изображения: отражение, поворот, растяжение — и некоторые другие. Кстати, именно здесь обеспечиваются упоминавшиеся ранее изменения размеров листа бумаги. Назначение двух последних пунктов “Палитра” и “Справка” понятно из их названия.

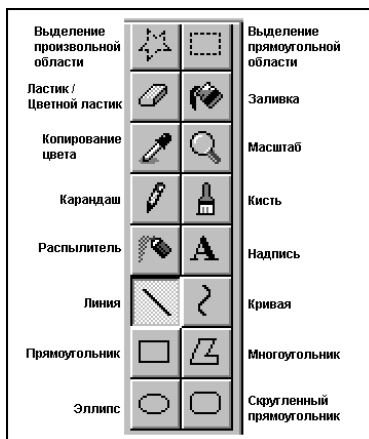
Другой орган управления графическим редактором называется **панель инструментов**. Она находится в левой части окна и состоит из набора **операционных кнопок**. Нажимая мышью на ту или иную кнопку, пользователь выбирает необходимый ему графиче-

ский инструмент. Paint располагает широким спектром инструментов. Часть из них имеют аналоги в обычном процессе рисования, например, карандаш, кисть, ластик, распылитель и заливка краской. Специальные операционные кнопки обеспечивают выделение области для дальнейшей обработки (например, копирования или сдвига), а также наложение на изображение набранного с клавиатуры текста (традиционно на этой операционной кнопке нарисована заглавная буква “А”). Очень полезна возможность изменения масштаба изображения, что позволяет, например, подретушировать неровную границу или замаскировать цветное пятно, возникшее в результате некачественного сканирования. Наконец, последняя группа инструментов, часто называемая **графическими примитивами**, позволяет рисовать аккуратные линии, прямоугольники и эллипсы, в частности, окружности. Они необычайно полезны при создании всевозможных схем и диаграмм. В заключение заметим, что Paint имеет специальную управляющую область для настройки выбранного инструмента, где, например, можно задать толщину линии или установить режим закраски внутренней области рисуемой фигуры.

Отдельно стоит рассмотреть этап сохранения вновь созданного в графическом редакторе рисунка. Опыт показывает, что начинающие пользователи если даже и знают о существовании различных графических форматов, то обычно не придают своим теоретическим знаниям значения. В результате созданный в редакторе Paint несложный рисунок с 4—5 цветами может занимать у них около 1 Мб. Создание неоправданно больших графических файлов является одним из ярких признаков “компьютерной безграмотности”.

Отчего зависит размер графического файла и как его можно существенно уменьшить?

Как уже отмечалось ранее, размер файла зависит от **количества цветов**. Даже если удастся уменьшить количество цветов не очень сильно, это все равно может дать ощутимый эффект, поскольку он сказывается на объеме информации о *каждой* точке рисунка. Приведем пример: при уменьшении с 256 цветов до 16 размер файла сокращается вдвое. Очевидно, что при большом количестве точек абсолютное уменьшение размера файла происходит значительно сильнее. К сожалению, Paint не позволяет выбирать количество цветов абсолютно произвольно, а имеет набор фиксированных значений: 2, 16, 256 и т.п. Если ваш рисунок содержит всего 3 цвета, то приходится задавать значение с запасом, т.е. 16. В то же время, более мощный редактор Adobe PhotoShop дает возможность сохранить минимально необходимое число цветов, причем определяется оно автоматически. Итак, рекомендуется устанавливать количество цветов, не превышающее (или незначительно превышающее) реальное число цветов на вашем рисунке.



Другим фактором, влияющим при прочих равных условиях на объем графического файла, является размер рисунка, или, выражаясь более точно, **количество точек** в нем. Каждому ясно, что чем больше точек, тем больше объем информации о них. Конечно, это не означает, что надо делать только небольшие рисунки. Просто не стоит сохранять их так, чтобы размер используемой части листа был незначительным. В качестве классического примера приведем упоминавшуюся в известной книге о Карлсоне картину “Очень одинокий петух”, где реальное изображение занимало только один из углов листа. Отсюда еще одна практическая рекомендация — всегда делайте окончательную площадь рисунка минимально возможной и не оставляйте без особой необходимости неиспользуемых полей (если потребуется, то дополнительные поля можно будет сделать при вставке картинки в документ).

Наконец, еще одним способом уменьшения объема графического файла является правильный выбор его **формата**. Стандартным графическим форматом в операционной системе Windows является **ВМР**. Это очень простой формат, который может использовать любое программное обеспечение. Тем не менее он имеет существенный недостаток — файлы **ВМР** очень велики. Для смягчения этого недостатка многие графические форматы используют внутреннее сжатие, базирующееся на объединении информации о нескольких идущих подряд точках одинакового цвета. В сети Интернет очень распространен один из таких форматов — **GIF**. Если сохранить один и тот же рисунок в форматах **ВМР** и **GIF**, то последний окажется существенно меньше при абсолютно таком же качестве изображения. К сожалению, рисунок в формате **GIF** не может иметь более 256 цветов, что несколько ограничивает его применение. Наконец, для хранения многоцветных художественных иллюстраций и фотографий существует специальный формат со сжатием — **JPEG** (часто расширение файлов уменьшают до стандартной длины и пишут **JPG**). Традиционные методы сжатия на многоцветные изображения практически не действуют (в них почти нет соседних точек с одинаковым цветом: например, все точки листа зеленые, но их оттенки различны). Поэтому большая группа экспертов по цветопередаче (*Joint Photographic Experts Group*), обобщив полувековой опыт исследований в области компьютеров, телевидения, полиграфии и других родственных областях, а также по особенностям человеческого зрения, представила детальные рекомендации по сжатию графической информации. Результат получился впечатляющим, объем графических файлов сократился более чем на порядок. Не следует только забывать, что преобразование в формат **JPEG** является необратимым — восстанавливаемый после сжатия рисунок не абсолютно идентичен исходному, хотя на глаз это практически незаметно. Файлы **JPG** являются вторым базовым стандартом изображений в сети Интернет.

Завершая обсуждение достоинств и недостатков различных файловых форматов, отметим, что, начиная с Windows версии 1998 года, редактор Paint способен, помимо стандартного формата **ВМР**, работать со всеми тремя обсуждаемыми выше форматами файлов.

Вопросы, связанные с графическими форматами файлов, представляют большой самостоятельный интерес. Наиболее любознательные читатели могут обратиться за интересными подробностями к указанным источникам дополнительной литературы [3, 4].

#### **Методическое замечание по содержанию ответа (предназначено для учителей информатики)**

Возможно, подбор материала для ответа на вопрос показался читателю несколько нетрадиционным, поэтому постараемся объяснить нашу логику. Наиболее существенные отличия от описаний в существующей литературе следующие.

**Больше внимания уделяется принципам обработки графической информации, а не конкретным инструментам редактора.** Справедливо принято считать, что на экзамен выносятся наиболее важные вопросы — те, которые являются главными в изучаемом курсе. Поэтому прежде всего нас удивило то, что в билетах есть теоретический(!) вопрос по возможностям графического редактора, но нет вопроса, связанного с представлением графической информации: векторная и растровая графика, принципы кодирования цвета, форматы графических файлов, сжатие графической, как, впрочем, и остальных видов информации и некоторые другие фундаментальные вещи, без которых повседневная работа на современном компьютере даже на уровне пользователя мне кажется невозможной. С другой стороны, в билете № 5 дополнительно имеется практическое задание по графическому редактору, где вполне можно поговорить о его возможностях. Исходя из всего этого, нам захотелось “усилить” часть материала, связанную с графической информацией; тем более что работа с конкретным редактором многократно описана во многих учебниках (см., например, список литературы).

**Практически не рассматриваются принципы работы видеооборудования компьютера.** К тому, что устройство компьютеров современных “юзеров” не интересует *принципиально*, мы как-то потихоньку уже стали привыкать, ибо трудно, практически невозможно бороться с их *воинствующим* невежеством (“не знаю и знать не желаю!”). Поэтому с этой точки зрения любая попытка “попутного” изучения устройства компьютера должна всячески приветствоваться. Но в данной конкретной ситуации — ответе на вопрос билета № 10 — мы не стали затрагивать этот интересный для нас круг проблем. В самом деле, пусть школьники отвечают на тот вопрос, который сформулирован в билете. А если все-таки требуется понимать, что такое видеопамять и какой ее объем требуется иметь в компьютере, что дает графический акселератор и всем ли он нужен, какую устанавливать частоту регенерации изображения и т.д. — так давайте все же не будем играть в прятки и спросим об этом напрямую, а не будем прикрываться общими разговорами о важности информационных технологий обработки графических изображений. По изложенным соображениям материалы по работе компьютерных видеоприборов в данном описании, наоборот, были опущены.

Напомним, что вопросы к экзамену являются примерными и учитель вправе изменять их самостоятельно.

### Литература

1. Шафрин Ю.А. Основы компьютерной технологии. Учебное пособие для 7—11-х классов по курсу “Информатика и вычислительная техника”. М.: АБФ, 1996, 560 с.
2. Шафрин Ю.А. Информационные технологии (в 2 ч.). М.: Лаборатория базовых знаний, 2000.
3. Семакин И., Залогова Л., Русаков С., Шестакова Л. Информатика. Учебник по базовому курсу (7—9-е классы). М.: Лаборатория базовых знаний, 1998, 464 с.
4. Примерные билеты и ответы по информатике для подготовки к устной итоговой аттестации выпускников 9-х классов общеобразовательных учреждений в 2000/2001 учебном году / Авт.-сост.: А.А. Кузнецов, Н.Д. Угринович, Л.Е. Самовольнова. М.: Дрофа, 2001, 64 с.
5. Радченко Н.П., Козлов О.А. Школьная информатика: экзаменационные вопросы и ответы. М.: Финансы и статистика, 1998, 160 с.

### Дополнительная литература

1. Залогова Л.А. Форматы графических изображений. “Информатика и образование” № 6, 1999, с. 76—81.
2. Залогова Л.А. Цвет в компьютерной графике. “Информатика и образование” № 7, 1999, с. 85—90.
3. Залогова Л.А. Форматы графических файлов. “Информатика и образование” № 8, 1999, с. 85—94.
4. Залогова Л.А. Практикум по компьютерной графике. М.: Лаборатория базовых знаний, 2001, 320 с.
5. Кенцл Т. Форматы файлов Internet. СПб.: Питер, 1997, 320 с.

\*\*\*

## 2. Разработка алгоритма или программы, содержащей команду повторения (оператор цикла).

### Задача

Найти и вывести на экран сумму первых  $N$  натуральных чисел.

Говорят, что задание найти сумму первых 100 чисел в далекую “докомпьютерную” эру дал один учитель арифметики своим ученикам, намереваясь немного отдохнуть, пока они считают. К несчастью для него, в классе якобы оказался юный Гаусс. Он моментально догадался разбить числа на 50 пар (первое с последним, второе с предпоследним и т.д.), сумма каждой из которых равнялась 101, и получил ответ.

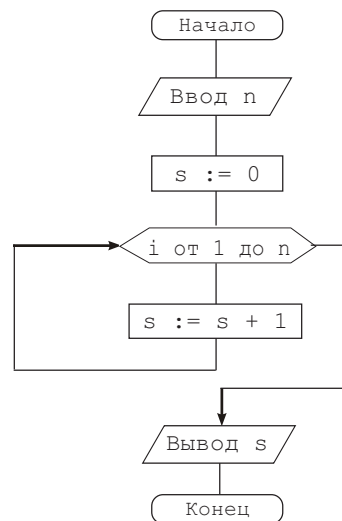
Наш алгоритм будет честно суммировать все числа без всяких рационализаций.

Во всех приведенных решениях используются одинаковые обозначения переменных:

- $n$  — значение последнего суммируемого числа, аргумент алгоритма;
- $s$  — переменная для накопления суммы; результат алгоритма;
- $i$  — промежуточная (рабочая) переменная, “пробегающая” от 1 до  $n$ .

### Решение

#### Блок-схема алгоритма



### Решение на различных языках

#### Школьный алгоритмический язык

```

алг сумма
нач цел n, i, s
  вывод "n=?"; ввод n
  s := 0;
  нц для i от 1 до n
    | s := s + i
  кц
  вывод "s=", s
кон
  
```

#### Язык Паскаль

```

program summa(input, output);
var n, i, s: integer;
begin write('N=? '); readln(n);
      s := 0;
      for i := 1 to n do s := s + i;
      writeln('S=', s)
end.
  
```

#### Язык Бейсик

```

10 input "n=";n
20 s = 0
30 for i = 1 to n
40 s = s + i
50 next i
60 print "s=";s
  
```

#### Язык Си

```

#include <stdio.h>
void main()
{int n, i, s;
 printf("N=? "); scanf("%d", &n);
 for (s = 0, i = 0; i <= n; s += i, i++);
 printf("S= %d\n", s);
}
  
```

# Азы информатики

А.А. ДУВАНОВ

Материалы  
Роботландского  
университета

Книга 1. См. № 1, 2/2002

Книга 2. См. № 5, 6, 7, 8, 9, 11, 13/2002

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*

### 8. ОБРАБОТКА ИНФОРМАЦИИ



#### Читальный зал

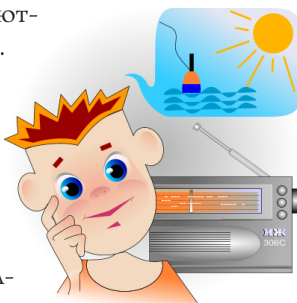
#### Как получить новую информацию

Новые сведения поставляются источниками информации.

Вася Кук внимательно послушал прогноз погоды на завтра и остался доволен: запланированный поход на рыбалку будет поддержан ясным, безветренным днем и согрет ласковым летним солнышком. Кук получил информацию о погоде из радиопередачи.

Полученная от источника информация не всегда пригодна к употреблению без дальнейшей обработки.

Известно, что, наблюдая за природой, можно предсказать погоду на завтра. После удачного рыболовного дня Вася вышел из палатки, внимательно смотрел и слушал. Вот какую информацию преподнесла ему природа:



квакали лягушки, похолодало, выпала роса, ветки деревьев не тревожил ветер. Но полученная информация не помогла Васе узнать, какой же завтра будет день.

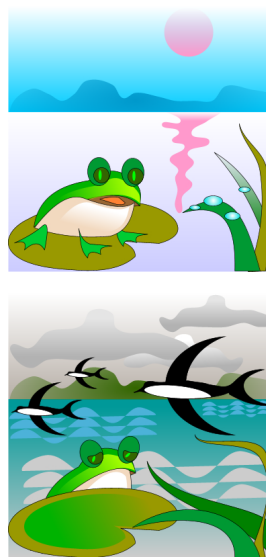
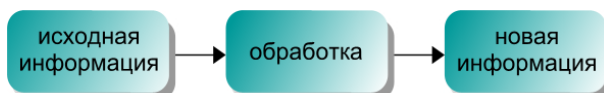
Дело в том, что для предсказания погоды нужно было обработать полученную информацию по известным с древних времен правилам. Васе эти правила были неизвестны, а между тем они очень просты.

Если вечером громко квакают лягушки, выпала роса, похолодало, на закате небо было чистым, нет ветра и в воздухе чувствуются запахи, можно смело собираться в поход — на следующий день погода будет хорошей.

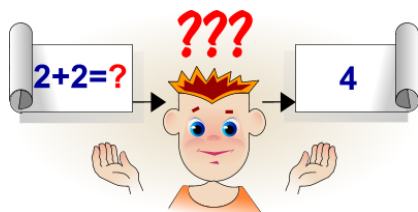
Если комары очень больно кусаются, порывистый ветер, на закате тучи заволокли небо, низко летают ласточки, то погода будет плохой.

Собирая информацию, которую дает природа, и применяя перечисленные выше правила, можно получить информацию о том, какая завтра будет погода.

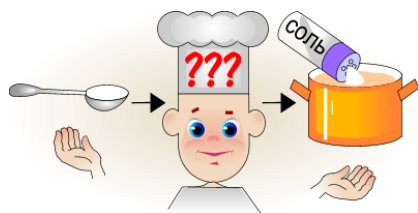
Информацию, которую обрабатывают, называют **исходной**. После обработки исходной информации получается **новая** информация.



Информация обрабатывается людьми, животными и техническими устройствами.



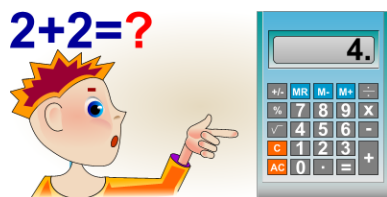
Школьник получает условия задачи (исходную информацию), думает (обрабатывает) и сообщает ответ (новую информацию).



Повар пробует суп. Вкус супа — исходная информация. Затем он думает (обрабатывает) и получает новую информацию: чего в супе не хватает.



Служебная собака по запаху легко находит человека. Здесь запах — исходная информация, а направление, куда пошел человек, — новая.



На калькуляторе можно выполнить сложные вычисления. Введенный в него пример — исходная информация, а полученный ответ — новая.

## Обработка информации на компьютере

Компьютер — специальный прибор, созданный человеком для обработки информации.

Если калькулятор можно сравнить с лопатой, то компьютер — с мощным трактором: компьютер способен “перепахивать” огромные пласты информации за удивительно короткое время. Современные компьютеры способны за одну секунду выполнять более миллиарда арифметических и логических операций.



Лопата и трактор — это средства механизации труда человека.

Компьютер же не просто механизует умственный труд человека, он автоматизирует его. Это означает, что компьютер способен работать автоматически по заранее написанной программе.



Существуют, например, целые автоматические заводы, в которых за станками не стоят рабочие. Компьютер, следуя заданной программе, автоматически управляет станками, и выпуск продукции идет круглые сутки без перерыва на обед и сон.

Для выпуска другой детали просто меняют компьютерную программу. И если сегодня станки производили гайки, то завтра они будут производить болты.

Обычный компьютер, который стоит на вашем столе, тоже настроен на автоматическую работу. После его включения начинает выполняться специальная программа, которая называется **операционной системой**, или сокращенно **ОС**. На самом деле в состав ОС входит не одна, а целый комплекс программ.

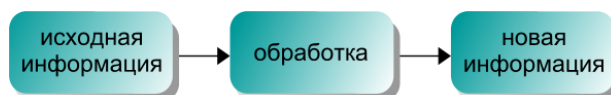
Вероятно, на вашем компьютере установлена операционная система Windows. В ее задачи входит:

- Проверка компьютера и подготовка его к работе.
- Предоставление пользователю удобных средств для управления работой компьютера.
- Запуск программ (приложений) на выполнение.
- Подготовка компьютера к отключению от электрической сети.

## Конспект

Информация обрабатывается людьми, животными и техническими устройствами.

Схема обработки информации:



Компьютер — это специальный прибор, созданный человеком для обработки информации.

Компьютер способен работать автоматически по заранее написанной программе.

**Операционная система** — это специальная программа, которая начинает работу после включения компьютера.

В задачи ОС входит:

- Проверка компьютера и подготовка его к работе.
- Предоставление пользователю удобных средств для управления работой компьютера.

- Запуск программ (приложений) на выполнение.
- Подготовка компьютера к отключению от электрической сети.



## Вопросы

1. Прочитайте рассказ. Где в рассказе описаны хранение, передача и обработка информации?

### Котенок Васька завтракает в лесу

Пошел рыжий котенок Васька в лес позавтракать. Увидел ягоду бруснику. Попробовал — кислая. Учюял гриб-боровик, попробовал его — горький. Вдруг слышит шуршание и видит: лежит в густой траве комочек взъерошенный.



“Ну, — подумал котенок, — тобой и позавтракаю”.

Хотел он лапой подкатить комочек чуть-чуть поближе да как закричит: “Ой! Ай!”

Комочек ежом оказался.



Сел Васька на пенек и задумался. Думал он, думал и решил, что лучше с ежом не связываться, грибным завтраком не рисковать, а поесть кислых ягод.



2. Приведите примеры обработки информации человеком, животными, техническими устройствами. В каждом примере укажите исходную информацию и новую, которая получилась в результате обработки.

3. Нарисуйте схему обработки информации.

4. Для чего предназначен компьютер?
5. Что такое механизация и автоматизация? Чем они отличаются друг от друга?

6. Является ли телевизионный пульт управления устройством автоматизации? Если да, то что он автоматизирует, умственный или физический труд?

7. Верно ли утверждение о том, что лопата автоматизирует труд человека?

8. Почему компьютер является прибором для автоматизации труда человека?

9. Может ли быть автоматом не компьютер? Если да, то как программируются такие устройства?

10. Может ли компьютер автоматизировать полив огорода? Если да, то как?

11. Может ли компьютер участвовать в пошиве платья, строительстве моста?

12. Расскажите, как работает завод-автомат.

13. Что такое операционная система?

14. Какие задачи выполняет ОС?



## Задания на дом



### Вариант 1

1. Ученик решил примеры и записал их в тетрадь:

$$5 \cdot 4 - 2 = 18$$

$$5 - 4 : 2 = 3$$

Укажите, что здесь является исходной информацией, а что — новой?

2. Собираясь за город, веселые человечки напекли пирожков и разложили их по рюкзакам. Знайка записал: “У Синеглазки 5 пирожков с повидлом и 5 — с ягодами. Пончик забрал 10 пирожков с мясом. Винтик и Шпунтик поделили поровну 8 пирожков с капустой и 12 — с яблоками. Пилюлькин взял витаминные пирожки: 5 — с морковкой, 2 — с капустой, 3 — с яблоками и 5 — с ягодами”.

Помогите Знайке обработать эту информацию и представить в виде таблицы. Что при этом будет исходной информацией, а что — новой?

3. Чем надо дополнить таблицу, чтобы Знайка мог увидеть в ней ответы на вопросы:

- Сколько пирожков разных видов взяли коротышки?
- Сколько пирожков взял каждый из них?
- Сколько всего пирожков взяли в поход?



### Вариант 2

1. Поработайте с программой *Калькулятор* из группы *Стандартные*.

Выберите в меню “Вид” позицию “Инженерный”. Решите пример:

$$7 \cdot 31 + 4 \cdot 30 + 29 =$$

Что здесь было исходной информацией, а что — новой?

2. Выполнив предыдущее задание, обратите внимание на то, в какой позиции стоит отметка в блоке радиокнопок: “Hex”, “Dec”, “Oct” или “Bin”? Не стирая ответа, перемещайте отметку в другие позиции и записывайте результат. Какое действие производит компьютер с информацией?

Попытайтесь найти объяснение с помощью раздела меню “Справка”.

3. Как вы думаете, что вычисляет алгоритм, приведенный в примере задания 1?



### Вариант 3

1. Восхитившись видами природы, поэт написал:

“Мороз и солнце — день чудесный!”

Что произошло с информацией? Что здесь было исходной информацией, а что — новой?

2. Узнайте прогноз погоды на завтра и попытайтесь произвести его творческую обработку, написав короткую зарисовку в прозе или в стихах или нарисовав картинку — завтрашний вид из вашего окна. На следующий день сравните результат вашей творческой обработки с действительностью.



## Практикум

## Бухгалтер



Исполнитель *Бухгалтер* может выполнять четыре арифметических действия: сложение, вычитание, умножение и деление. Для выполнения определенного действия нужно:

- Задать операцию на панели управления.
- Записать в полях ввода операнды ( $a$  и  $b$ ).
- Нажать кнопку “Пуск” для запуска вычисления.

Примеры заданий *Бухгалтера*:

- 1) Вычислить:  $11\ 623 - 3544$ .
- 2) Вычислить:  $10\ 022 + 2544$ .
- 3) Вычислить:  $24\ 523 : 137$ .
- 4) Вычислить:  $9106 + 8197$ .
- 5) Вычислить:  $528 \cdot 933$ .

## Переводчик



Исполнитель *Переводчик* выполняет перевод слов с русского языка на английский и обратно. Для работы с ним нужно:

- Задать направление перевода на панели управления.
- Записать в поле ввода слово.
- Нажать кнопку “Пуск” для запуска перевода.

Примеры заданий *Переводчика*:

- 1) Перевести: user.
- 2) Перевести: сохранить.
- 3) Перевести: open.
- 4) Перевести: клавиатура.
- 5) Перевести: word.

## Вопросы

1. Что является исходной и новой информацией в исполнителях?
2. Что означает для *Бухгалтера* нажатие на кнопку с новой операцией?

3. Что означает для *Переводчика* нажатие на кнопку с новым направлением перевода?

4. Расскажите алгоритм работы *Бухгалтера*.

5. Расскажите алгоритм работы *Переводчика*.

6. Как работает клавиша **Enter** в исполнителях?



## Зачетный класс

1. Отметьте устройства, предназначенные для механизации труда человека:

- лопата;
- бурильная установка;
- стиральная машина;
- мясорубка;
- счеты.

2. Отметьте устройства, предназначенные для автоматизации труда человека:

- бурильная установка;
- автоматическая стиральная машина;
- калькулятор;
- компьютер;
- станок с программным управлением.

3. Какие свойства компьютера позволяют считать его устройством для автоматизации умственного труда?

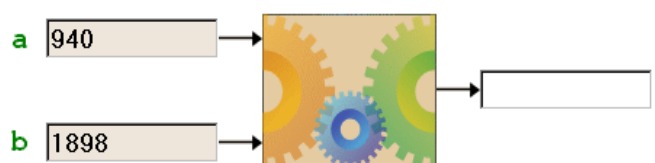
- Очень быстро работает;
- может выполнять арифметические операции;
- может выполнять логические операции;
- выполняет программы;
- помогает писать и рисовать.

4. Отметьте те случаи, когда человек занимается программированием:

- Вася играет на компьютере;
- Игорь пишет письмо на компьютере;
- Иван составляет план вычисления площади прямоугольника;
- бухгалтер считает на компьютере зарплату сотрудников;
- Света ищет в Интернете красивую картинку.

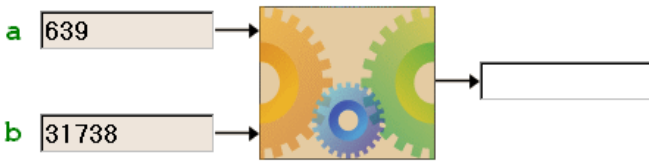
5. Исполнитель умножает числа на два. На вход поступило число 25. На устройстве отображения зажглось число 50. Какая информация является исходной, а какая новой?

6. Исполнитель работает по следующему алгоритму. Новое число получается приписыванием по порядку к  $a$  справа тех цифр  $b$ , которые меньше 5. Запишите, что получится в результате такой обработки:



7. Исполнитель работает по следующему алгоритму. Новое число получается приписыванием по порядку к  $a$  справа тех цифр  $b$ , которых нет в  $a$ . Запишите, что получится в результате обработки:





8. Исполнитель работает по следующему алгоритму:

- 1) Найти наибольшую цифру в  $a$ .
- 2) Найти наименьшую цифру в  $b$ .
- 3) Записать новое число из двух цифр: первая — результат выполнения пункта 1, вторая — результат выполнения пункта 2.

Запишите, что получится в результате такой обработки:



9. Исполнитель работает по следующему алгоритму:

- 1) Найти разность между наибольшей и наименьшей цифрой в  $a$ .
- 2) Найти разность между наибольшей и наименьшей цифрой в  $b$ .
- 3) Записать разницу между значениями, найденными в пунктах 1 и 2.

Запишите, что получится в результате такой обработки:

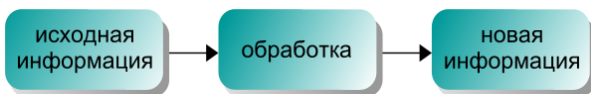


## Книга для учителя

### 8. ОБРАБОТКА ИНФОРМАЦИИ

#### Схема обработки информации

На приведенной в “Читальном зале” схеме:



использованы термины “исходная информация” и “новая информация”.

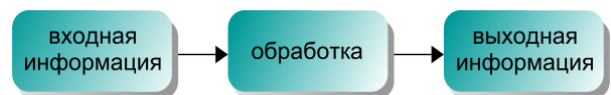
Так как информатика не рассматривает субъективные стороны содержания информационных процессов, то под “новой” понимается информация, полученная в результате обработки, вне зависимости от ее новизны для обработчика или сторонних субъектов.

Например, Васю попросили сложить два и три. Является ли ответ “пять” новостью для Васи или его товарищей? Конечно, нет. Число пять — это просто результат обработки, и “новой” эта информация называется для обозначения этого факта.

На самом деле ничто не мешает “новой” информации (то есть информации, полученной в результате обработки) совпадать с исходной. Пусть, например, обработчик делит число на два, если оно четное, и не меняет его в противном случае. Обработка числа “3” приведет тогда к точно такой же “3” на выходе.

На практике исходную информацию часто называют **входной**, а новую информацию — **выходной**, в соответствии с тем, что входная информация поступает на вход обработчика, а выходная считывается с его выхода. Эти термины не содержат в себе субъективизма, поэтому в дальнейших темах курса заменят термины “исходная” и “новая”, введенные в этом уроке ради большего соответствия интуитивным понятиям младшего школьника.

Информационная схема обработки при этом будет иметь вид:



Отметим, наконец, что входная или выходная информация (или обе вместе) могут быть пустыми. Это обычная ситуация в практической информатике.

#### Первый пример

Обработчик перемещает по экрану объект и следит за клавиатурой. Пока клавиши не нажаты (пустая информация на входе), объект движется в прежнем направлении. Нажатие клавиши меняет направление движения.

#### Второй пример

Обработчик ожидает нажатия специальной управляющей клавиши (чтобы закрыть, например, информационную панель). На неверные клавиши он не реагирует (пустая информация на выходе), хотя все нажатия принимает и анализирует (обрабатывает).

#### Третий пример

Пока пользователь думает у клавиатуры, текстовый редактор ничего не выводит на экран (пустая информация на входе и на выходе). Но обработка идет! Обработчик все время опрашивает клавиатуру — не нажал ли заснувший пользователь какую-нибудь клавишу.

Таким образом, в информатике иногда бывает, как у пожарных: отсутствие всяких новостей — хорошая новость!

#### Котенок Васька

Котенок Васька анализирует информацию, которая хранилась в ягоде, в грибе, в еже. Для получения этой информации ему пришлось пустить в ход все свои информационные каналы — органы чувств: зрение (увидел

ягоду), обоняние (учуял гриб), вкус (ощутил кислый привкус ягоды, горечь гриба), слух (услышал шуршание ежа в траве), осязание (потрогал колючие иголки ежа).

Правда, об обработке информации в этой истории сказано весьма коротко: “Сел Васька и задумался”. Зато очевиден результат обработки: не очень вкусные ягоды котенок Васька предпочитает колючему ежу.

История про любопытного котенка Ваську, несмотря на всю ее простоту, может дать богатый материал для работы в классе. С этой целью можно использовать традиционное сказочное повторение сюжета: в первый раз объектом исследования котенка служит ягода (увидел — попробовал), во второй раз — гриб (учуял — попробовал), в третий раз — еж (услышал — потрогал).

Анализ встречи с первым объектом — ягодой — учитель может взять на себя, а описание двух других встреч, пользуясь сказочным подобием ситуаций, поручить детям.

Новая информация, появившаяся в результате первой обработки, содержится в мысли: “ягоду есть не стоит”. Итог второй обработки: “гриб тоже невкусен”. Результатом третьей обработки является вывод о предпочтении невкусных, но безопасных ягод.

## Бухгалтер и Переводчик

С одной стороны, эти исполнители подкрепляют тему обработки информации, с другой — продолжают тренировать навыки текстового ввода. В этой двойке функции их и надо рассматривать.

*Бухгалтер* имеет четыре встроенных алгоритма, каждый из которых настроен на выполнение одной математической операции. Алгоритмы находятся в “постоянной” памяти робота, они недоступны пользователю ни для просмотра, ни для изменения, но пользователь может переключать алгоритмы на панели управления.

*Переводчик* имеет два “встроенных” алгоритма: один переводит слова с английского на русский, другой — в обратном направлении. Кроме алгоритмов, в “постоянную” память робота записан словарь (таблица, каждая строка которой состоит из слова и его перевода). Переключение алгоритмов, так же как и в *Бухгалтере*, выполняется кнопками на панели управления.

На вход алгоритмов *Бухгалтера* поступают два числа:  $a$  и  $b$  — первый и второй операнды. Выходная информация — число, результат операции.

На вход *Переводчика* поступает слово, а на выходе отображается результат перевода.

Обратите внимание детей на аварийные сообщения исполнителей. *Бухгалтер* показывает панель с надписью “Не понимаю”, когда на один из его входов поступает строка, отличная от числа. Сообщение “Не могу” появляется при попытке деления на ноль.

*Переводчик* выдает аварийное сообщение тогда, когда заданного для перевода слова нет в его словаре. При этом, если задан алгоритм  $Ru \rightarrow En$ , поиск слова производится в “русском” столбце таблицы-словарика, а если задан алгоритм  $En \rightarrow Ru$  — в английском.

## Задания зачетного класса

Задания зачетного класса, в которых необходимо выполнить алгоритм обработчика и записать полученный ответ (выходную информацию), не слишком сложные по выполнению, но коварные по формулировкам. Юному школьнику придется очень внимательно прочитать описание алгоритма, чтобы понять его суть.

Цель предлагаемых заданий:

- Работа с конкретными алгоритмами, повышающими алгоритмическую культуру школьника.

- Практика выполнения алгоритма вручную (прокрутка).

- Работа с терминами и понятиями этого и предыдущих уроков (исполнитель, алгоритм, вход, выход, исходная информация, новая информация, обработка).

- Развитие абстрактного мышления.

На зачете школьники почувствуют больший комфорт, если заранее порешать подобные задачи в классе.

Задания зачета генерируются случайным образом, списывание ответов друг у друга в классе практически исключается.

## Ответы на вопросы

**Ответ на вопрос 1.** Разбор этой удивительной истории приводится выше в разделе “Котенок Васька”.

**Ответы на вопрос 2.**

История: Вася купил мороженое не за 10 рублей, а за 5.

Исходная информация: цены на мороженое и величина суммы в кошельке.

Обработчик: Вася.

Обработка: Вася сравнил цены с содержимым своего кошелька.

Новая информация: решение купить мороженое за пять рублей.

История: кошка увидела собаку и забралась на дерево.

Исходная информация: отображение собаки на сетчатке кошачьих глаз, опыт прежних встреч с собаками, хранимый в памяти кошки.

Обработчик: кошка.

Обработка: образ объекта был признан собакой, в памяти была найдена информация о прежних встречах с этим животным.

Новая информация: решение забраться на дерево.

История: Вася опустил жетон в телефон-автомат, набрал номер и пригласил Катю на прогулку.

Исходная информация: сигнал об опущенном жетоне, набранный номер.

Обработчик: телефон-автомат.

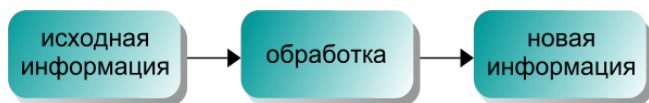
Обработка: сначала сравнение опущенного предмета с признаками стандартного жетона.

Новая информация: предмет является жетоном для оплаты разговора.

Новая обработка: проверка возможности соединения по набранному номеру.

Новая информация: соединение произошло.

Ответ на вопрос 3.



Ответ на вопрос 4. Компьютер — это специальный прибор, созданный человеком для обработки информации. Конечно, компьютер не только обрабатывает информацию, но и получает ее (через клавиатуру, мышь, сканер, микрофон, диск, Интернет), передает (на монитор, диск, принтер, в Интернет) и хранит (в ОЗУ, на винчестере).

Ответ на вопрос 5. Механизация — это использование механизмов для выполнения работы. Автоматизация — это выполнение механизмами работы без участия человека. Автомат — это программируемый механизм.

Автомат — это механизм, но не любой механизм — автомат, так как не все механизмы можно программировать.

Ответ на вопрос 6. Пульт управления — это не автомат, он работает под управлением человека и механизмирует его физический труд (не надо бежать к телевизору, чтобы переключить программу).

Ответ на вопрос 7. Неверно, ведь лопату нельзя программировать.

Ответ на вопрос 8. Компьютер работает по программе.

Ответ на вопрос 9. Да. Например, механический будильник. Запрограммировать его на подачу сигнала очень просто: нужно, вращая колесико, установить специальную стрелку в желаемое положение.

Ответ к заданию 10. Конечно. Он может управлять включением системы полива, когда датчики сообщат ему, что почва суха, и, наоборот, выключать подачу воды, когда почва достаточно увлажнится.

Ответ на вопрос 11. Компьютер может по специальным программам рассчитать выкройку платья для конкретного клиента, получив от него необходимые размеры, или рассчитать прочность конструкции спроектированного моста.

Ответ на вопрос 12. В компьютер вводят программу изготовления детали. Выполняя эту программу, компьютер подает управляющие сигналы на исполнительные устройства станков, и станки производят деталь. Смена программы приводит к смене изделий, изготавливаемых автоматическим заводом.

Ответ на вопрос 13. Операционная система — это специальная программа, которая начинает работу после включения компьютера.

Ответ к заданию 14. В задачи ОС входит:

- Подготовка компьютера к работе.
- Управление компьютером.
- Запуск приложений.
- Подготовка компьютера к отключению.

## Решения зачетного класса

Ответ к заданию 1. Все указанные устройства.

Ответ к заданию 2. Автоматическая стиральная машина, компьютер, станок с программным управлением.

Ответ к заданию 3. Выполняет программы.

Ответ к заданию 4. Иван составляет план вычисления площади прямоугольника.

Ответ к заданию 5. Исходная информация — 25, новая — 50.

Ответ к заданию 6. 9401.

Ответ к заданию 7. 639178.

Ответ к заданию 8. 80.

Ответ к заданию 9. 4.

## Калейдоскоп

### Слушается дело

Интересное дело слушалось недавно в суде Соединенных Штатов. Одна из почитательниц американской поп-звезды Чарли Прайда (Charley Pride) подала в суд на звукозаписывающую компанию за то, что та не поставила ее в известность о том, что альбом ее любимца снабжен защитой от копирования. В силу этой причины его нельзя не только скопировать, но и прослушать на DVD-, MP3- и компьютерных CD-ROM-плеерах.

Возмущенная покупательница заявила, что ее ожидания оказались обманутыми и звукозаписывающая компания ответит за это.

Самое интересное, что дело было выиграно. Компании, ответственные за выпуск альбома и снабжение его защитой от копирования, были обязаны поставить в известность всех будущих покупателей о том, что их диск не совместим с большинством распространенных аудиоплееров.

Другой интересный судебный процесс проходил в Германии и продолжался

около двух лет. Все дело в том, что в 1995 году компания Medianet, штаб-квартира которой располагается во Франкфурте, зарегистрировала для себя web-адрес [www.deutschland.de](http://www.deutschland.de), где и разместила онлайн-консультацию. Позднее права на тот же адрес решило получить немецкое правительство, однако закон есть закон: кто раньше зарегистрировал, тот и владеет.

Два года лучшие адвокаты Германии доказывали, что никто не вправе обладать именем страны, кроме правительства, эту страну представляющего.

Наконец суд принял решение в пользу правительства. Уже в этом месяце по вышеозначенному адресу откроется официальный портал с полной информацией по стране. Что до Medianet, то представители правительства утверждают, что они решили дать компании откупного, дабы не выглядеть грабителями.

### Сухой язык цифр

Согласно исследованиям, проведенным в Великобритании, англичане тратят около трех с половиной милли-

онов часов, ожидая возможности воспользоваться домашним компьютером! Таким образом, соперничество вокруг машины разворачивается даже более ожесточенно, нежели вокруг телевизора. Согласно отчету, составленному в Microsoft, в настоящее время домашними компьютерами владеют 60% из 15,2 миллиона семей.

Другое исследование касается Интернета в Америке. Граждане Штатов стали относиться ко Всемирной сети уже без былого восторга, они уже не проводят в Интернете большую часть своего времени, а когда все же выходят в сеть — предпочитают заниматься чем-нибудь полезным. Скажем, искать работу, резервировать билеты или покупать акции. В 2001 году пользователи за одно соединение проводили в on-line примерно восемьдесят три минуты — на семь минут меньше, чем в позапрошлом году. Очевидно, среднестатистическое время пребывания в сети будет сокращаться и дальше.

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

# Использование электронной таблицы Microsoft Excel и языка Visual Basic for Application для построения графиков состояния идеального газа

О.В. Туркин,  
Москва

Предлагаемая программа предназначена для наглядной иллюстрации первого закона термодинамики, поэтому сначала уделим немного внимания материалу из курса физики 10-го класса. Как известно, уравнение состояния идеального газа имеет вид

$$\frac{PV}{T} = \frac{m}{\mu} R, \quad (1)$$

где  $P$  — давление газа,  
 $V$  — объем сосуда, который занимает газ,  
 $T$  — температура газа по шкале Кельвина,  
 $m$  — масса газа,  
 $\mu$  — молярная масса газа,  
 $R$  — универсальная газовая постоянная.

Если рассматривается один и тот же газ неизменной массы, то правая часть уравнения является постоянной.

$$\frac{PV}{T} = R_0 \quad (2)$$

Если мы хотим исследовать какой-либо изопроцесс, необходимо, приняв один из параметров с левой стороны уравнения за постоянную, менять два оставшихся параметра. Так при постоянном давлении можно получить уравнение изобарного процесса

$$\frac{V}{T} = \frac{R_0}{P}; \quad (3)$$

и если речь идет о переходе газа из состояния с параметрами  $P, V_1, T_1$  в состояние  $P, V_2, T_2$ , то связь между этими состояниями будет иметь вид

$$\frac{V_1}{T_1} = \frac{V_2}{T_2}. \quad (4)$$

Аналогично можно получить формулы для изохорного и изотермического процессов

$$\frac{P}{T} = \frac{R_0}{V}; \quad \frac{P_1}{T_1} = \frac{P_2}{T_2}; \quad (5)$$

$$PV = R_0 T; \quad P_1 V_1 = P_2 V_2. \quad (6)$$

Известно, что эти изопроцессы могут быть изображены на графиках в координатах  $P-V$ ;  $P-T$ ;  $V-T$  и существует ряд задач на подобные построения, а также на переход из одной системы координат в другие. Кроме того, в школьном курсе есть задачи на построение замкнутых термодинамических циклов, состоящих из подобных процессов.

Поставим перед собой следующую цель — создать программу, которая позволяла бы строить графики газовых процессов и замкнутых циклов одновременно в трех системах координат, а также иллюстрировала бы энергетические превращения в этих процессах.

В качестве оболочки для создания такой программы используем электронную таблицу Microsoft Excel, имею-

щую развитые средства создания графических изображений и возможность подключения модулей на языке VBA для автоматизации процесса построения графиков.

Обсудим общий вид нашей программы и определимся с числовыми параметрами.

Сначала определяем начальное состояние газа. Как известно, при нормальных условиях газ имеет давление 100 кПа и температуру 273 К. В термодинамических расчетах обычно рассматриваются давления, не превышающие 100 атмосфер и температуры до 1000 К. Поэтому в качестве ориентировочного исходного состояния принимаем давление равным 500 кПа, температуру — 500 К и объем — 5 м<sup>3</sup>. Тогда, полагая отношение  $m/\mu = 1$ , можно значение постоянной  $R_0$  принять равным 5. Тем не менее предусмотрим в программе возможность устанавливать исходное состояние по своему выбору, сохраняя значение постоянной.

Из параметров газа мы будем менять температуру и давление. Действительно, если говорить о реальных физических процессах, то в них, как правило, происходит либо изменение температуры газа, либо давления. В большинстве случаев изменение объема, который занимает газ, обусловлено изменением внешнего давления и, следовательно, давления газа.

Далее следует определить шаг, с которым будут изменяться выбранные параметры. Практически оказалось удобным изменять давление с шагом 20 кПа и температуру с шагом 50 градусов.

Также необходимо определиться в длительности происходящих процессов. Если, скажем, речь идет о построении замкнутого цикла, то, видимо, придется предусмотреть возможность создания различных изопроцессов в рамках одного этапа построения. Пусть у нас будет 25 таких изопроцессов. Ясно, что это число выбрано произвольно.

Переходим к созданию программы. Сначала поработаем в электронной таблице. Определяем ячейки для исходного состояния. Во второй строке блока A1:C2 будут размещены начальные значения давления, объема и температуры (этим ячейкам даны имена PPP, VVV и TTT соответственно), как это представлено в табл. 1.

Таблица 1

	A	B	C
1	=PPP * 100	=VVV	=TTT * 100
2	5	5	5

В данном случае эти цифры выбраны в качестве ориентира. Перед нами же стоит задача обеспечить произвольный выбор начальных параметров. Конечно, можно было бы заполнять содержимое ячеек A2:C2 вручную, но гораздо удобнее использовать для этого такой элемент управления, как полоса прокрутки — ScrollBar. Этот элемент дает богатые возможности для автоматизации работы даже без применения программного кода.

Итак, внедряем на наш рабочий лист элемент управления `ScrollBar` (Полоса прокрутки). Для этого через пункты меню “Меню | Вид | Панели” инструментов вызываем панель элементов управления, входим в режим конструктора и перетаскиваем с этой панели на рабочий лист полосу прокрутки, причем располагаем ее в вертикальном положении.

При выделенной полосе прокрутки там же, в панели элементов управления, вызываем окно свойств и определяем:

1. Минимальное и максимальное значения числа (`Value`), соответствующие двум крайним значениям ползунка на полосе прокрутки. Так как первая полоса будет отвечать за значение давления, а числа могут принимать только целые положительные значения, то задаем свойство `Max = 1000`, а свойство `Min = 0`.

2. Используем и такое свойство, как `LinkedCell` (Связанная ячейка). Оно устанавливает ячейку, в которой будет отображаться значение `Value`. В качестве таковой выбираем `J3` и, предварительно занеся в нее произвольное целое число (до 1000), в строке окна свойств `LinkedCell` вписываем `J3`.

Теперь, если при отключенном конструкторе перемещать ползунок полосы прокрутки, то значения в этой ячейке будут изменяться от 0 до 1000. Мы же определились, что в `A2` значение будет меняться от 0 до 10 ( $\times 100$  кПа), поэтому необходимо произвести перерасчет. В ячейку `J4` заносим формулу `=J3/100`, а в ячейку `A2` — формулу `=J4`.

В действительности можно было бы обойтись без `J4` и сразу в `A2` получить значение из `J3/100`, но в данном случае это будет нелишним в целях наглядности и дополнительного контроля.

Таким же образом располагаем вторую полосу прокрутки (горизонтальную), устанавливаем минимальное и максимальное значения 0 и 1000 соответственно, указываем ячейку связи `K3`, проводим в ячейке `K4` деление на 100 и в ячейку `C3` (температура, деленная на 100) заносим формулу `=K4`. Теперь при перемещении ползунка на горизонтальной полосе прокрутки начальное значение температуры будет изменяться.

Необходимо отметить, что физически нулевые значения температуры и давления нереализуемы, поэтому предпочтительнее установить минимальное значение не 0, а, например, 100. Но в данном случае нам важно продемонстрировать использование полосы прокрутки.

Итак, начальные давления и температура определены. Вычисляем объем по формуле (2). Для этого в таблице должно находиться значение  $R_0 = 5$ . Занесем в ячейку `H2` число 5 (это значение газовой постоянной, которое при необходимости можно менять), а в ячейку `B2` внесем формулу `=H2*C2/A2`. В ячейки `D2` и `E2` помещаем значения шага для изменения давления и температуры в соответствующем масштабе — 0,2 и 0,05.

Теперь займемся формированием графической области. Предполагается, что программные коды будут составлены таким образом, чтобы, изменяя с определенным шагом давление или температуру, автоматически заполнялись все столбцы, характеризующие состояние газа. Под значения давления нами отведен столбец `A`, под объем — столбец `B`, а под температуру — `C`. Первая строка, в которую будут занесены вычисленные значения, — строка 3. Следовательно, для построения графиков мы должны использовать эти три столбца.

Полагаем, что мы будем проводить вычисления до строки 250, это значение выбрано, исходя из возможности построения до 25 различных процессов с длительностью каждого в 10 шагов. Используя стандартные средства электронной таблицы, создадим три диаграммы типа “Точечная”, которые имеют следующие исходные данные:

- диаграмма  $P-V$ : значения  $X$  — `B2:B250`, значения  $Y$  — `A2:A250`;
- диаграмма  $V-T$ : значения  $X$  — `C2:C250`, значения  $Y$  — `B2:B250`;
- и диаграмма  $P-T$ :  $X$  — `C2:C250`,  $Y$  — `A2:A250`.

Если параметры диаграмм заданы правильно, то на каждой из них можно будет увидеть точку, перемещающуюся при изменении положения ползунков на полосах прокрутки.

Теперь приступаем к созданию элементов управления. Для внешнего оформления располагаем на рабочем листе четыре элемента управления `TextBox` или `Label`, в которых просто поместим текстовую информацию: соответственно “Изменение положения начальной точки”, “Изобара”, “Изотерма”, “Изохора” (см. рис. 1). Внешнее оформление этих элементов — дело вкуса.

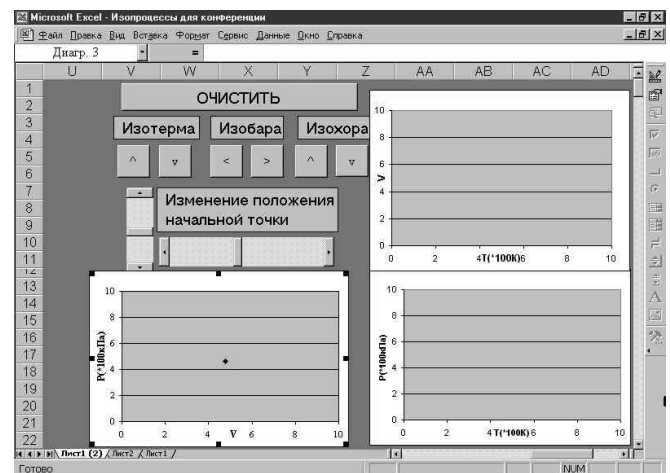


Рис. 1. Компоновка элементов управления на листе Microsoft Excel

Затем там же размещаем семь элементов `CommandButton` (Кнопка), при щелчке на которые и будут запускаться соответствующие подпрограммы. Не будем тратить время на переименование этих элементов и согласимся с тем, что по умолчанию предлагает VBA, запомнив лишь, что кнопки 1 и 2 (`CommandButton1` и `CommandButton2`) будут относиться к изотерме, 3 и 4 — к изобаре, 5 и 6 — к изохоре. Свойство `Caption` у этих кнопок нужно изменить таким образом, чтобы указать направление изменения соответствующего параметра. Для изотермы будем либо увеличивать (символ “^”), либо уменьшать давление, для изобары и изохоры соответственно меняем температуру. Необходимо отметить, что это соответствует реальным способам создания этих процессов.

Не забудем и еще об одном элементе управления с надписью “Очистка”, назначение которого — инициализация переменных начальными значениями. Еще раз обратите внимание на рис. 1, где изображена компоновка элементов.

Теперь пришла пора писать коды подпрограмм.

Позаботимся о создании начальных условий, необходимых для начала построения графиков: на диаграмме изображена только одна точка, а все ячейки блока `A2:C250` — пустые. Перейдем в окно редактора VBA (например, двойным щелчком по кнопке `CommandButton7` или другим способом) и

создадим в модуле для 1-го листа (на котором размещены все наши элементы управления) процедуру

```
Private Sub CommandButton7_Click()
Range("a3:c250").Clear
stroka = 2
End Sub
```

В этой процедуре, во-первых, очищаются ячейки указанного блока, а во-вторых, определяется переменная *stroka*, указывающая текущий номер строки, для которой производятся вычисления. Она должна быть доступной для всех процедур, поэтому ее необходимо объявить как глобальную переменную. Для этого в блоке *General*, в разделе *Declarations*, набираем

```
Dim stroka As Integer
```

Следующий шаг — расчет какого-либо процесса и заполнение ячеек столбцов А—С с последующим построением соответствующих диаграмм. Для примера рассмотрим вариант расчета изотермического процесса с возрастанием давления. В этом случае из последней заполненной строки (определяется переменной *stroka*) в переменную *p* пересылаем значение из ячейки столбца давлений (А) и одновременно увеличиваем эту переменную на величину шага, которая хранится в D2 (2-я строка, 4-й столбец). Получаем следующую строку кода

```
p = Cells(stroka, 1) + Cells(2, 4)
```

По формуле (6) для изотермического процесса вычисляем новый объем. Для этого, помимо “нового” давления, необходимы значение температуры (та же строка, столбец 3) и газовая постоянная (ячейка H2 — строка 2, столбец 8). “Новый” объем помещаем в следующую строку столбца 2. Строка кода выглядит так:

```
Cells(stroka + 1, 2) =
Cells(2, 8) * Cells(stroka, 3) / p
```

Наступила пора увеличить переменную *stroka* на единицу —

```
stroka = stroka + 1
```

и в “новую” строку перенести значения температуры и давления —

```
Cells(stroka, 3) = Cells(stroka - 1, 3)
Cells(stroka, 1) = p
```

Чтобы отобразить новые данные в диаграммах, необходимо вставить команду “Делай событие”

```
DoEvents
```

Получившийся набор команд полезно оформить в виде отдельной процедуры. Дадим ей имя *termab* и, используя в редакторе VBA команду меню *Insert/Procedure*, создаем следующую процедуру:

```
Public Sub termab()
p = Cells(stroka, 1) + Cells(2, 4)
Cells(stroka + 1, 2) =
Cells(2, 8) * Cells(stroka, 3) / p
stroka = stroka + 1
Cells(stroka, 3) = Cells(stroka - 1, 3)
Cells(stroka, 1) = p
DoEvents
End Sub
```

Почти так же будет выглядеть процедура для изотермического процесса, если давление нужно уменьшать, только здесь необходимо позаботиться о том, чтобы она выполнялась, если давление больше 0,5 (×100 кПа). Имя процедуры — *termam*, и выглядит она так:

```
Public Sub termam()
p = Cells(stroka, 1)
If p > 0.5 Then
p = p - Cells(2, 4)
Cells(stroka + 1, 2) =
Cells(2, 8) * Cells(stroka, 3) / p
stroka = stroka + 1
Cells(stroka, 3) = Cells(stroka - 1, 3)
Cells(stroka, 1) = p
DoEvents
End If
End Sub
```

Процедуры для реализации изобарического процесса (температура вверх — *izobb* и температура вниз — *izobm*) отличаются тем, что происходят считывание, оценка и изменение температуры (переменная *t*), расчет объема по формуле (3) и перенос без изменения в следующую строку давления.

```
Public Sub izobb()
t = Cells(stroka, 3) + Cells(2, 5)
Cells(stroka + 1, 2) =
Cells(2, 8) / Cells(stroka, 1) * t
stroka = stroka + 1
Cells(stroka, 1) = Cells(stroka - 1, 1)
Cells(stroka, 3) = t
DoEvents
End Sub
```

```
Public Sub izobm()
t = Cells(stroka, 3)
If t > 0.5 Then
t = t - Cells(2, 5)
Cells(stroka + 1, 2) =
Cells(2, 8) / Cells(stroka, 1) * t
stroka = stroka + 1
Cells(stroka, 1) = Cells(stroka - 1, 1)
Cells(stroka, 3) = t
DoEvents
End If
End Sub
```

Для изохорного процесса (*izoxb* и *izoxm*) по сравнению с двумя предыдущими процедурами изменена расчетная формула. Здесь вычисляется новое давление по формуле (5), а объем остается без изменения.

```
Public Sub izoxb()
t = Cells(stroka, 3) + Cells(2, 5)
Cells(stroka + 1, 1) =
Cells(2, 8) / Cells(stroka, 2) * t
stroka = stroka + 1
Cells(stroka, 2) = Cells(stroka - 1, 2)
Cells(stroka, 3) = t
DoEvents
End Sub
```

```
Public Sub izoxm()
t = Cells(stroka, 3)
If t > 0.5 Then
t = t - Cells(2, 5)
Cells(stroka + 1, 1) =
Cells(2, 8) / Cells(stroka, 2) * t
stroka = stroka + 1
Cells(stroka, 2) = Cells(stroka - 1, 2)
Cells(stroka, 3) = t
DoEvents
End If
End Sub
```

Теперь позаботимся о том, чтобы для каждого изопротесса процедура выполнялась десять раз подряд, то есть заполнялись 10 новых строчек. Вот здесь нам и понадобятся

6 кнопок CommandButton. Для первой кнопки под событие “Щелчок по кнопке” мы создаем следующую процедуру

```
Private Sub CommandButton1_Click()
    For i = 1 To 10
        termab
    Next
End Sub
```

Остальные процедуры будут запускаться оставшимися пятью кнопками.

Теперь, выключив режим конструктора и щелкнув предварительно по кнопке “Очистить”, выбираем с помощью полос прокрутки начальное положение точки где-то в середине графика P—V и строим различные цепочки изопрцессов и термодинамические циклы.

Поговорим более подробно о том, как в этом же проекте можно проиллюстрировать первое начало термодинамики. Вернемся к физике. Как известно, первое начало имеет вид

$$Q = A + \Delta U,$$

где Q — количество теплоты, подведенное к газу, A — работа, совершенная газом в процессе, ΔU — изменение внутренней энергии газа.

Для вычисления работы в различных процессах используются разные формулы, но основная формула имеет вид

$$A = P \cdot \Delta V,$$

где ΔV — изменение объема.

Именно эту формулу мы будем использовать, полагая, что в исследуемых нами процессах шаг расчетов настолько мал, что давление при получении изменения объема можно считать постоянным. Вообще-то это допущение необходимо только для изотермического процесса, потому что там одновременно меняется и объем, и давление. Изменение внутренней энергии будем вычислять по формуле  $\Delta U = C_m \Delta T$ , или, не вдаваясь в подробности преобразования,

$$\Delta U = kR_0 \Delta T = kR_0 T_2 - kR_0 T_1.$$

Коэффициент k определяется природой газа, строением его молекул, и в наших расчетах полагаем, что этот коэффициент равен 2 (ячейка P1). Кроме того, запишем следующую формулу:

$$\Delta U = kR_0 T_2 - U_1,$$

где U<sub>1</sub> — внутренняя энергия газа в начальном состоянии.

Таким образом, расчеты можно проводить по следующей схеме. Определяем U<sub>1</sub> по начальным параметрам газа, затем, используя значение температуры для произвольного состояния, можем найти ΔU. Так как значение работы зависит от того, какой процесс происходит с газом, то здесь необходимо производить суммирование работ, совершенных газом на каждом шаге, поэтому в одном из столбцов таблицы в каждой строке будем производить сложение предыдущей строки с работой, выполненной на этом шаге. Рядом будет располагаться столбец, в котором складывается работа газа и изменение его внутренней энергии, значение которого берем еще из одного столбца. Теперь, согласно этой схеме, можно внести формулы в электронную таблицу. Фрагмент заполнения показан в табл. 2.

Необходимо напомнить, что в этих формулах rrr — имя ячейки H2, в которой хранится R<sub>0</sub>.

	A	B	□	...	R	S	T
1	=PPP * 100	=VVV	=TTT * 100		A	DU	Q
2						= \$P\$1 * rrr * C2	
3					=A3 * (B3 - B2) + R2	= \$P\$1 * rrr * C3 - \$\$S2	= R3 + S3
...					...	...	...
7					=A7 * (B7 - B6) + R6	= \$P\$1 * rrr * C7 - \$\$S2	= R7 + S7
8					=A8 * (B8 - B7) + R7	= \$P\$1 * rrr * C8 - \$\$S2	= R8 + S8
...					...	...	...

В качестве диаграммы, на которой можно наблюдать соотношение между этими термодинамическими параметрами, используем обычную гистограмму с тремя рядами данных. Первый ряд будет отображать изменение внутренней энергии газа для последней расчетной точки, второй — значение работы, а третий — количество теплоты. Для этих рядов назначаем по одной ячейке — M5, N5 и O5 соответственно. Теперь, чтобы диаграмма отображала данные текущей точки, мы должны позаботиться о том, чтобы в эти ячейки при каждом пересчете переносились значения ΔU, A, Q из столбцов S, R и T, взятые из последней рассчитанной строки. С этой целью дополняем нашу программу процедурой termod

```
Public Sub termod()
    Cells(5, 14) = Cells(stroka, 18)
    Cells(5, 13) = Cells(stroka, 19)
    Cells(5, 15) = Cells(stroka, 20)
End Sub.
```

Вызов этой процедуры будем осуществлять во всех шести процедурах для командных кнопок (CommandButton1\_Click() — CommandButton2\_Click()) перед строкой Next.

Теперь при запуске программы на диаграмме можно наблюдать изменение размеров столбиков в зависимости от значения термодинамических параметров. В качестве контроля правильности работы программы можно использовать тот общеизвестный факт, что для замкнутых циклов, соответствующих изображенным на рис. 2, изменение внутренней энергии равно нулю при полном завершении цикла.

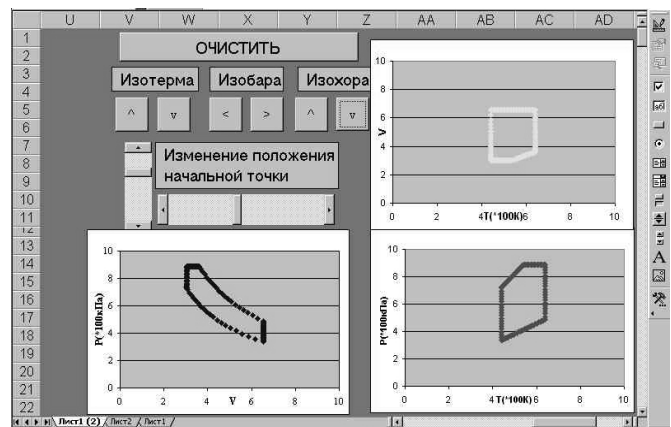


Рис. 2. Примеры работы программы

Безусловно, этот проект может быть усовершенствован рядом дополнений, например, можно добавить счетчик цепочек процессов, ввести дополнительные ограничения на максимальные значения температуры, объема, давления и т.д.

Но важнее другое. Удачное, на наш взгляд, сочетание возможностей ЭТ и программирования открывает большие перспективы для наглядного моделирования разнообразных природных процессов.

Таблица 2

# ГЕОМЕТР



Окончание.  
Начало на с. 1

Необходимость последнего, пятого, постулата для построения геометрии не кажется очевидной: сам Евклид доказывает целый ряд теорем, не опираясь на него, и непонятно, почему в этот ряд не могут быть включены все теоремы евклидовой геометрии. Особая роль пятого постулата, его сложность и недостаточная наглядность привели к тому, что математики более поздних времен стали пытаться доказать его как теорему.

Лобачевский занялся исследованием той совокупности теорем, которая может быть выведена из системы аксиом, получаемой, если заменить пятый постулат (аксиому параллельных) евклидовой геометрии противоположным утверждением: *в плоскости через точку А, не принадлежащую прямой l, можно провести более одной прямой, не пересекающейся с l*. Созданная, таким образом, геометрическая система носит теперь имя неевклидовой геометрии, или геометрии Лобачевского.

Независимо от Лобачевского существование такой системы установили Карл Фридрих Гаусс (1777—1855) и венгерский математик Янош Бойяи (1802—1860). Гаусс не публиковал сообщений о своем открытии, а Бойяи опубликовал результаты своих исследований в 1832 году в виде приложения к большому геометрическому трактату своего отца, учителя математики. Эта работа относится к числу выдающихся математических произведений, однако в ней представлены лишь первые понятия новой геометрии.

Лобачевский был не только великим геометром. Ему принадлежат ценные работы в области математического анализа, причем он одним из первых дал общее определение функциональной зависимости. В алгебре известен его метод приближенного вычисления корней уравнений любой степени. В своих высказываниях о соотношении геометрии и физики Лобачевский предвосхитил идеи науки двадцатого столетия [2].

Новую геометрию ученый назвал "воображаемой", говоря, что если она не существует в природе, то, во всяком случае, существует в математическом анализе.

Неевклидова геометрия (это название предложено Гауссом) в течение нескольких десятилетий оставалась заброшенной областью науки. Первым видным математиком, целиком понявшим ее значение, был Бернхард Риман (1826—1866); его общая теория многообразий (1854 г.) допускала не только существовавшие виды неевклидовой геометрии, но и многие другие, так называемые "римановы геометрии". Однако полное признание этих теорий состоялось уже после смерти Лобачевского.

После того, как в 1868 году был решен вопрос, касающийся непротиворечивости новой геометрии, она становится неотъемлемой частью математики, причем ее идеи и методы находят применение в теории чисел, теории функций, теории дифференциальных уравнений, в математической физике [1, 2].

На основе идей Лобачевского сформировалась вся современная геометрия, играющая важнейшую роль как в математике, так и вообще в точном естествознании.

## Литература

1. Соловьев Ю.П. Николай Иванович Лобачевский // Рассказы о математике и математиках. М.: МЦНМО, 2000.
2. Стройк Д.Я. Краткий очерк истории математики: Пер. с нем. Изд. 4-е. М.: Наука, 1984.

Гл. редактор  
С.Л. Островский  
Зам. гл. редактора  
А.И. Сенокосов  
Редакция:  
Е.В. Андреева  
Н.Л. Беленькая  
Л.Н. Картвелишвили  
Н.П. Медведева  
Дизайн и верстка:  
Н.И. Пронская  
Корректоры:  
Е.Л. Володина,  
С.М. Подберезина

©ИНФОРМАТИКА 2002  
выходит четыре раза в месяц  
При перепечатке ссылка  
на ИНФОРМАТИКУ обязательна,  
рукописи не возвращаются

Адрес редакции  
и издателя:  
121165, Киевская, 24  
тел. 249-48-96  
Отдел рекламы  
тел. 249-98-70

Учредитель: ООО "Чистые пруды"

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 27.03.2002.  
Номер подписан 27.03.2002.  
Заказ №  
Цена свободная

ИНДЕКС ПОДПИСКИ  
для индивидуальных подписчиков 32291  
комплекта изданий 32744

Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84

Internet: [inf@1september.ru](mailto:inf@1september.ru)  
WWW: <http://www.1september.ru>

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

Газеты ИЗДАТЕЛЬСКОГО ДОМА: Первое сентября — гл. ред. Е.Бирюкова, Английский язык — гл. ред. А.Громушкина, Библиотечка 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.Бузоева, 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.