

ИНФОРМАТИК А



IBM 5100. Первую и неудачную попытку выпустить персональный компьютер фирма IBM сделала еще в 1975 г.

Двадцатилетие

20 лет назад, в 1981 году, фирма IBM выпустила свой первый персональный компьютер (IBM PC)

В конце лета 1981 года в средствах массовой информации промелькнуло сообщение, последствия которого тогда вряд ли кто-то смог оценить по достоинству. Было объявлено о выпуске корпорацией IBM (особенно знаменитой в ту пору большими и мощными машинами семейства IBM 370) "своей самой компактной и недорогой компьютерной системы — IBM Personal Computer". Отмечалось, что "сконструированная специально для применения в бизнесе, в школе и дома,

Окончание читайте на с. 32

Читайте в номере

Информация 2, 8, 30

Заканчивается учебный год. Впереди — весенние каникулы, за ними — мимолетная четвертая четверть и — лето. Снова пора готовиться к 1 сентября. В летних номерах мы предложим вам тематические выпуски, которые существенно облегчат подготовку к новому учебному году.

Уроки 3–8

С.М. Окулов. Основы программирования

"Функции, как и процедуры, используются для структурирования программ", причем в ряде языков "вообще нет разделения программ на процедуры и функции".
Постоянный автор нашей газеты приглашает на занятие, посвященное функциям, в начале которого он предлагает вспомнить несколько фактов из комбинаторики.

Начальная школа 9–24

Ю.А. Первин. Зимние вечера. Информатика для начинающих. Выпуск 8

"От того, что ученик смотрит решения другого, — двойная польза..." Предложите малышам проверить решения задач, полученные другими детьми. Далее познакомьте ребят с текстовым редактором "Микрон", а затем расскажите о файлах и попросите немного с ними поработать.

Внеклассная работа 25–27

Д.М. Златопольский. Внеклассная работа по информатике

Можно ли размер каталога измерить в кубических сантиметрах, размер текстового файла — в словах, а размер монитора — в символах?
Пусть ребята найдут ошибки, а потом предложите им выполнить задание, где фигурируют "программы-профессии" и "программы-должности".

Беседы 28–29

А.А. Дуванов. "Назаметки" Сидорова. № 9. Визуальные редакторы

Нужно ли изучать язык HTML, когда есть такие хорошие визуальные HTML-редакторы, как HomeSite, FrontPage Express, FrontPage 2000?
Ваш знакомый Иван Сидоров продолжает давать рекомендации, касающиеся web-дизайна. Кстати, он считает, что, "свободно владея HTML, вы будете ощущать себя принятым в хорошую компанию".

На стенд в кабинете информатики 31

"Врожденный механик"

Иван Петрович Кулибин был не простым механиком...
О выдающемся изобретателе, "первом механике России", чье творчество оказало весьма существенное влияние на развитие русской технической мысли.

Никогда!

Никогда еще лето не было таким жарким!

Нигде!

Нигде, кроме как на страницах "Информатики", вы не найдете такой замечательной, уникальной подборки интереснейших материалов!

Никто!

Никакое другое издание, никакое другое издательство не предложат вам столько интересных материалов сразу!

Только!

В "Информатике", в серии "Жаркое лето-2001"! Не забудьте: "Информатика" распространяется только по подписке.

"ЖАРКОЕ ЛЕТО-2001"*



А.А. Дуванов

JavaScript-конструирование

Летом мы предложим вашему вниманию газетную версию очередного учебника Роботландского университета. Курс "JS-конструирование" продолжает "HTML-конструирование" (см. № 21, 22/2000). Как всегда, подписчики нашей газеты (и только они!) получат скидку на программные продукты и обучение в Роботландском университете.



В.Н. Пинаев

Олимпиады по информатике. Теория и практика

Даже если вы не очень любите олимпиады по информатике, они — объективная реальность, данная нам в телефонограммах. Как подготовить ребят к олимпиадам? Где взять задачи для проведения школьных олимпиад? Как эти олимпиады проводить? Давайте послушаем знающего человека.



С.М. Окулов

Основы программирования

В течение года мы публиковали материалы (занятия) из новой книги нашего постоянного автора. В летних номерах мы завершим публикацию этой книги, в которую включены новые, никогда ранее не публиковавшиеся в "Информатике" материалы.



English for Computer Science Teachers

How does a mouse work? Why aren't floppy disks floppy? How fast is the fastest computer? Have computers got brain?

Учителю информатики постоянно приходится иметь дело с англоязычными текстами. Конечно, один номер не может служить настоящим учебником, справочником или словарем. Но мы надеемся, что этот необычный номер будет вам очень полезен и интересен.



Л.О. Сергеев

Компьютерные вирусы

Наши постоянные подписчики, конечно, помнят, что более трех лет назад мы опубликовали последнюю редакцию "Компьютерных вирусов". С тех пор изменилось... все! Мы предложим вашему вниманию совершенно новую книгу, которая будет интересна всем — и учителям, и ученикам, и даже их родителям.



Обязательный минимум содержания образования по информатике: и в нем нам хочется дойти до самой сути

Что стоит за разделами минимума? Что имели в виду его авторы? (Такие вопросы тоже нередко возникают.) Давайте прочитаем минимум еще раз. Вместе, спокойно, не торопясь.



Л.Н. Картвелишвили

На стенд в кабинете информатики

Целый номер "на стенд"? Да, целый номер! Причем весь этот номер будет состоять из совершенно новых и очень интересных материалов. Уже 1 сентября стенды в вашем кабинете будут лучшими в школе!



Н.Г. Иванова, О.Л. Русакова Информатика для школьников 5–6-х классов: ЛОГО-логика

Летом мы предложим вашему вниманию очень интересную разработку, созданную пермскими авторами, — рабочие тетради для параллельного изучения основ логики и языка ЛОГО.

Как получить эти номера?

Есть только один (зато стопроцентно надежный!) способ — подписаться на "Информатику".

Индекс подписки — 32291 (по каталогу "Роспечать").

* Данный план является предварительным. Некоторые тематические выпуски выйдут в нескольких номерах газеты.

Основы программирования

С.М. Окулов,
г. Киров

Продолжение. См. № 42, 43, 44, 45, 46, 47, 48/2000;
6, 7, 8, 10/2001

Занятие 12. Функции

План занятия

- описание функций;
- несколько фактов из комбинаторики;
- экспериментальная работа с программами вычисления числа сочетаний из n по m ; получения палиндрома числа путем последовательного его “переворачивания” и сложений; нахождения общей части n отрезков;
- выполнение самостоятельной работы.

Описание функций

Функции, как и процедуры, используются для структурирования программ. В некоторых языках вообще нет разделения подпрограмм на процедуры и функции. В Паскале такое разделение есть, но носит оно преимущественно синтаксический характер. Заголовок функции начинается с ключевого слова `Function` и кончается типом возвращаемого данной функцией значения.

`Function <имя функции>`

`[ (<список параметров>) ] : <тип результата>;`

В теле функции обязательно должен быть хотя бы один оператор присваивания, где в левой части стоит имя функции, а в правой — ее значение. Иначе значение функции не будет определено.

Несколько фактов из комбинаторики

Перестановки

Перестановкой из n элементов назовем упорядоченный набор из n различных натуральных чисел, принадлежащих отрезку от 1 до n . Пусть n равно 3. Перечислим все перестановки из 3 элементов: (1 2 3), (1 3 2), (2 1 3), (2 3 1), (3 1 2), (3 2 1). Количество перестановок из n элементов (обозначается как $P(n)$) равно $n! = 1 \cdot 2 \cdot \dots \cdot n$ (факториалу числа n). Действительно, есть n способов для выбора элемента на первое место, $(n-1)$ способ для выбора элемента на второе место и т.д. Общее количество способов — это произведение $n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$.

Перечислите все перестановки из 4 элементов. Количество перестановок — 24. Подсчитайте вручную значения $n!$ для n из диапазона от 5 до 12.

Размещения с повторениями

Пусть имеется множество элементов, относящихся к n различным видам. Из них составляются всевозможные наборы по m элементов в каждом. При этом в

наборы могут входить и элементы одного вида. Два набора считаются различными, если они отличаются друг от друга или входящими в них элементами, или их порядком. Пусть $n = 3$, $m = 2$ (как обычно, рассматриваем натуральные числа). Размещения с повторениями: (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3). Число таких наборов $R(n, m) = n^m$. Это доказывается методом математической индукции.

Перечислите все размещения с повторениями при $n = 4$ и $m = 2$. Подсчитайте вручную значение $R(n, m)$ для нескольких значений n и m .

Размещения без повторений

Размещением без повторений из n элементов по m называется упорядоченный набор из m различных чисел, принадлежащих отрезку от 1 до n . Два набора считаются различными, если они отличаются составом или порядком элементов. Пусть $n = 4$, $m = 2$. Перечислим все размещения без повторений: (1, 2), (2, 1), (1, 3), (3, 1), (1, 4), (4, 1), (2, 3), (3, 2), (2, 4), (4, 2), (3, 4), (4, 3). Количество размещений $A(n, m)$ вычисляется по формуле:

$$A(n, m) = n \cdot (n-1) \cdot \dots \cdot (n-m+1) \text{ или}$$

$$A(n, m) = \frac{n!}{(n-m)!}.$$

Перечислите все размещения из 5 элементов по 2. Подсчитайте вручную значения $A(n, m)$ для фиксированного значения m , например 4, и n из интервала от 8 до 13.

Сочетания

Сочетанием из n элементов по m называется набор из m различных чисел, выбранных из диапазона от 1 до n . Наборы различаются лишь составом элементов, порядок элементов роли не играет. Пусть $n = 5$, $m = 3$. Перечислим все сочетания: (1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 3, 4), (1, 3, 5), (1, 4, 5), (2, 3, 4), (2, 3, 5), (2, 4, 5), (3, 4, 5). Каждому сочетанию соответствует $m!$ размещений. Обозначим число сочетаний $C(n, m)$. Таким образом,

$$A(n, m) = C(n, m) \cdot m! \text{ или}$$

$$C(n, m) = \frac{A(n, m)}{m!} = \frac{n!}{m!(n-m)!}.$$

Перечислите все сочетания из 7 элементов по 5. Подсчитайте число сочетаний для различных значений n и m (не очень больших). При подсчете как числа сочетаний, так и числа размещений используется операция деления. Обоснуйте, почему в результате получают только целые числа.

Экспериментальный раздел занятия

1. Составить программу подсчета числа сочетаний — $C(n, m)$. Пусть у нас есть функция подсчета факториала числа — $\text{Fact}(n)$. Написание основной программы сводится к программированию формулы

$$C(n, m) = \frac{n!}{m!(n-m)!}$$

```
Program My12_1;
  Var n,m: Integer;
      c: Longint;
  Function Fact(n:Integer):Longint;
  Begin
    End;
Begin
  WriteLn('Введите n и m :');
  ReadLn(n,m);
  c:=Fact(n) / (Fact(m) * Fact(n-m));
  WriteLn(c);
  ReadLn
End.
```

При компиляции этой программы (функцию Fact мы пока не написали, но компилироваться программа должна и без нее) в строке

```
c:=Fact(n) / (Fact(m) * Fact(n-m));
```

возникает ошибка — Error 26: Type mismatch. — несоответствие типов. Вспомним, что типы данных определяют не только диапазон значений переменных, но и допустимые над этим типом операции. Результат операции деления “/” нельзя присваивать переменным целого типа. Необходимо использовать операцию Div , т.е. соответствующая строка должна выглядеть следующим образом:

```
c:=Fact(n) Div (Fact(m) * Fact(n-m));
```

Дополняем программу.

```
Function Fact(n:Integer):Longint;
  Var i: Integer;
      rez: Longint;
  Begin
    rez:=1;
    For i:=1 To n Do rez:=rez*i;
    Fact:=rez
  End;
```

Запускаем программу для различных значений n и m . При $n = 10$, $m = 5$ результат 252 не вызывает сомнений. При $n = 13$, $m = 5$ результат 399, что-то не так... должно получиться большее значение. При $n = 15$, $m = 7$ результат 9, теперь уже видно, что программа не работает. Объясните причину. Подсказка содержится в материале этого занятия.

А можно ли упростить программу? Попробуйте убрать переменную c .

Сколько лишних вычислений (умножений) делает наша программа? Подсчитаем

$C(13, 5) = (1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13) : ((1 \cdot 2 \cdot 3 \cdot 4 \cdot 5) \cdot (1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8))$.

Вычисляя значение $C(13, 5)$ вручную, мы, конечно, не будем выполнять все эти умножения. Достаточно вычислить $9 \cdot 11 \cdot 13 = 1287$. Напишите более эффективную функцию вычисления числа сочетаний. Например, даже нижеприведенный вариант программы даст более обнадеживающие результаты. Правильный результат при $C(13, 5)$, а именно 1287, уже шаг вперед, но это не предел улучшений программы.

```
Program My12_1m;
  Var n,m:Integer;
  Function S(n,m:Integer):LongInt;
  Var i:Integer;
      rez,cht:LongInt;
  Begin
    rez:=1;cht:=1;
    For i:=1 To m Do Begin
      rez:=rez*i;
      cht:=cht*(n-i+1)
    end;
    S:=cht Div rez
  End;
Begin
  WriteLn('Введите два числа: ');
  ReadLn(n,m);
  WriteLn(S(n,m));
  ReadLn
End.
```

На восьмом занятии мы познакомились с основной теоремой арифметики. Пусть нам не требуется вычислять $C(n, m)$, а необходимо представить это значение в виде $p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$, где $\alpha_i \geq 0$, а p_i — простые числа. Ограничения: $1 < n \leq 100$, $1 < m < n$. Вычислять “в лоб” — безнадежное занятие. Числа получаются огромные. Рассмотрим идею решения на примере. Вычисляем $C(8, 5) = 8! / (5! \cdot 3!)$. Представим $8!$ как единицы в соответствующем массиве, например, с именем Sn , $Sn[i] = 1$, если число i есть в произведении. Просматриваем массив Sn , выбираем очередной элемент. Если число i составное, то находим его разложение на простые сомножители. К элементам Sn , соответствующим сомножителям, прибавляем количество единиц, равное числу их вхождений в i , а $Sn[i]$ обнуляем. Выполним эти действия для $n!$, $m!$ и $(n-m)!$. После этого останется только провести вычитание степеней одного и того же простого числа.

Числа \ i		2	3	4	5	6	7	8
8n	8!	1	1	1	1	1	1	1
	4	3	1	0	1	1	1	1
	6	4	2	0	1	0	1	1
	8	7	2	0	1	0	1	0
5m	5!	1	1	1	1	0	0	0
	4	3	1	0	1	0	0	0
3m	3!	1	1	0	0	0	0	0
Результат вычитания		3	0	0	0	0	1	0

Итак, $C(8, 5) = 2^3 \cdot 7^1$.

```

Program My12_2;
Const Q=100;
Type MyArray=Array[1..Q] Of Integer;
Var n,m:Integer;
    P:MyArray;
Procedure Solve(sn,sm:Integer;
                Var Sp:MyArray);
Begin
End;
Procedure Print(pn:Integer;Pp:MyArray);
  Var i:Integer;
  Begin
    For i:=2 To pn Do
      If Pp[i]<>0 Then WriteLn(i:5,Pp[i]:5)
    End;
  Begin
    WriteLn('Ввод чисел n и m');
    ReadLn(n,m);
    Solve(n,m,P);
    Print(n,P);
    ReadLn
  End.

```

Пока у нас есть только основная программа, содержащая вызовы процедур, и простая процедура Print. Программа компилируется. Уточним процедуру Solve.

```

Procedure Solve(sn,sm:Integer;
                Var Sp:MyArray);
  Var i:Integer;
      Sni,Smi,Snmi:MyArray;
Begin
  Calc(sn,Sni);
  {Сформировали массив, соответствующий n!}
  Calc(sm,Smi);
  {Сформировали массив, соответствующий m!}
  Calc(sn-sm,Snmi);
  {Сформировали массив, соответствующий (n-m)!}
  For i:=2 To sn Do Sp[i]:=Sni[i]-Smi[i]-Snmi[i]
End;

```

Сделав “заглушку” из процедуры Calc, содержащую только Begin и End, вы получите компилируемую программу. Вариант процедуры Calc, приведенный ниже, хотя и работает, но не очень хорошо читается. Выполните трассировку этой процедуры в режиме отладки, определите назначение каждого оператора. Ваша задача — улучшить процедуру. Вспомните о том, что в материале восьмого занятия перечислены все простые числа до 100.

```

Procedure Calc(n:Integer; Var X:MyArray);
  Var i,j,t:Integer;
  Begin
    FillChar(X,SizeOf(X),0);
    {SizeOf – вычисляет размер области памяти,
    выделенной для массива X;
    FillChar записывает нулевые значения
    в эту область памяти.}
    For i:=1 To n Do X[i]:=1;
    {Признаки чисел, задействованных
    в вычислении факториала числа.}
    For i:=4 To n Do Begin
      t:=i;
      j:=2;

```

```

      While (j<=(t div 2)) Do
        If t Mod j=0 Then Begin
          {j делит t без остатка.}
          Inc(X[j]); t:=t div j
          If t=j Then Inc(X[j])
        End
        Else Inc(j);
      If t<>i Then Begin
        {Найдены делители числа i.}
        X[i]:=0;If (t<>j) Then Inc(X[t])
      End
    End
  End;

```

Попробуем запустить эту программу при больших значениях n . Пусть $n \leq 10\,000$. Даже если изменить тип Integer на LongInt, появится ошибка Error 202: Stack overflow error. Нам не хватает оперативной памяти для хранения данных программы. Действительно, массивов введено много. Попробуем обойтись одним глобальным массивом.

```

{$R+}
Program My12_2m;
Const Q=10000;
Type MyArray=Array[1..Q] Of Integer;
Var n,m:LongInt;
    P:MyArray;
Procedure Inc_O(n:LongInt);
  Var i,j,t:Integer;
  Begin
    For i:=2 To n Do Begin
      t:=i; j:=2;
      While (j<=(t div 2)) Do
        If t Mod j=0 Then Begin
          Inc(P[j]);
          t:=t div j;
          If t=j Then Inc(P[j])
        End
        Else Inc(j);
      If t<>i Then Begin
        If (t<>j) Then Inc(P[t])
      End
      Else P[i]:=1
    End
  End;
Procedure Dec_O(n:LongInt);
  Var i,j,t:Integer;
  Begin
    For i:=2 To n Do Begin
      t:=i; j:=2;
      While (j<=(t Div 2)) Do
        If t Mod j=0 Then Begin
          Dec(P[j]);
          t:=t div j;
          If t=j Then Dec(P[j])
        End
        Else Inc(j);
      If t<>i Then Begin
        If (t<>j) Then Dec(P[t])
      End
      Else Dec(P[i])
    End
  End;

```

```

Begin
  WriteLn('Ввод чисел n и m'); ReadLn(n,m);
  Inc_O(n);
  Dec_O(m);
  Dec_O(n-m)
End.

```

Мы видим, что код процедур Inc_O и Dec_O почти повторяется. Это плохо. Попробуйте написать программу изящнее. Кстати, значение константы Q можно увеличить, например, до 30 000. Учтите, что для больших значений n и m программа может работать довольно долго.

2. В задании № 11 седьмого занятия требовалось определить, является ли введенное число палиндромом.

Оформим “перевертывание” числа в виде функции.

```

Function Pal(n:LongInt):LongInt;
Var x:LongInt;
Begin
  x:=0;
  While n<>0 Do Begin
    x:=x*10+n Mod 10;
    n:=n Div 10
  End;
  Pal:=x
End;

```

Пусть $n = 59$. “Перевернем” его, получим 95. Найдём сумму чисел 59 и 95, получим 154. “Перевернем” это число, получим 451. Находим сумму: 605. Еще раз: $605 + 506 = 1111$. Получили палиндром. Найдите для всех натуральных чисел из интервала от 50 до 80 количество шагов, необходимых для “сведения” их к палиндромам, с помощью описанной схемы.

```

Program My12_3;
Const a=50;b=80;
Var n,t:LongInt;
    cnt,i:Integer;
Function Pal(n:LongInt):LongInt;
...
End;
Begin
  For i:=a To b Do Begin
    cnt:=0;n:=i;
    While Pal(n)<>n Do Begin
      t:=Pal(n); n:=n+t;
      Write(t,' ',n,' ');
      Inc(cnt)
    End;
    WriteLn(i,' ',cnt)
  End
End.

```

Попробуйте получить палиндром из числа 89. На 16-м шаге результат по-прежнему не является палиндромом, а сумма выходит за пределы типа LongInt. Изменим программу, используя четвертый пример из раздела экспериментальной работы десятого занятия. Повторите материал о работе с большими числами. Для решения этой задачи потребуется их складывать и “перевертывать”.

```

{$R+}
Program My12_3m;
Uses Crt;
Const MaxN=1000;
Type MyArray=Array[0..MaxN+1] Of Integer;
{Числа храним в массиве, они большие
и типа LongInt не хватает.}
Var A,Z:MyArray;
    cnt,n:Integer;
Function Eq:Boolean;
{Является ли число палиндромом?}
Var i:Integer;
Begin
  i:=1;
  While (i<=A[0] Div 2) And (A[i]=A[A[0]-i+1]) Do Inc(i);
  Eq:=(i>A[0] Div 2)
End;
Procedure Swap(Var a,b:Integer);
Var c:Integer;
Begin c:=a;a:=b;b:=c End;
Procedure Rev;{ "Перевертываем" число.}
Var i:Integer;
Begin
  For i:=1 To A[0] Div 2 Do Swap(A[i],A[A[0]-i+1])
End;
Procedure Change(n:Integer);
{Записываем число в массив.}
Var i:Integer;
Begin
  i:=1;
  While n<>0 Do Begin
    A[i]:=n Mod 10; n:=n Div 10;
    Inc(A[0]);Inc(i)
  End
End;
Procedure Add;
{Складываем два больших числа.}
Var i,r,w:Integer;
Begin
  i:=1;r:=0;
  While (i<=A[0]) Or (r<>0) Do Begin
    w:=A[i]+Z[i]+r; A[i]:=w Mod 10;
    r:= w Div 10;
    If A[A[0]+1]<>0 Then Inc(A[0]);
    Inc(i)
  End
End;
Begin
  ClrScr; n:=100;
  While (n<1000) Do Begin
    {Исследуем диапазон от 100 до 999.}
    FillChar(A,SizeOf(A),0);
    Change(n);{Преобразуем число.}
    cnt:=0;
    {Счетчик количества шагов -
преобразований числа.}
    While (Not Eq) And (A[0]<=MaxN) Do Begin
      {Пока не палиндром и нет выхода за
пределы массива A, считаем.}
      Z:=A;Inc(cnt);Rev;Add
    End;
    WriteLn(n,' ',cnt);
    {Вывод числа и количества шагов.}

```

```

If cnt>=MaxN Then ReadLn;
{Фиксируем факт выхода за пределы массива A.}
Inc(n)
End
End.

```

Для чисел 196, 295, 394, 493, 592, 689, 691, 788, 790, 879, 887, 978, 986 массива из 1000 элементов недостаточно для хранения результата. А всегда ли конечен описанный процесс, всегда ли можно получить палиндром?

3. Как находить наибольший общий делитель двух чисел, мы знаем из восьмого занятия. Напишем соответствующую функцию.

```

Function Nod(a,b:LongInt):LongInt;
Begin
Repeat
If a>b Then a:=a Mod b Else b:=b Mod a
Until (a=0) Or (b=0);
Nod:=a+b
End;

```

Рассмотрим следующую задачу. Дано n отрезков, длины которых целые числа. Берем два отрезка и из большего вычитаем меньший. Разность — новый отрезок. Если же длины отрезков совпадают, то один из них исключаем из дальнейшего рассмотрения. Продолжаем процесс. В результате получается одно число, равное длине оставшегося отрезка. Пример: $n = 4$, длины отрезков: 6, 15, 3, 9.

Номер шага	Длины отрезков			
	6	15	3	9
1-й шаг	6	9	3	9
2-й шаг	6	3	3	9
3-й шаг	3	3	3	9
4-й шаг	—	3	3	9
5-й шаг	—	—	3	9
6-й шаг	—	—	3	6
7-й шаг	—	—	3	3
8-й шаг	—	—	—	3

Решать задачу “в лоб” по приведенной схеме можно, но для больших значений n это требует значительных временных затрат. На самом деле все сводится к нахождению наибольшего общего делителя n чисел, а вычитания выполнять не требуется. Напишите соответствующую программу.

Задания для самостоятельной работы

1. Дано n целых чисел. Найти среди них число, у которого приведенная ниже характеристика имеет максимальное значение:

- сумма цифр;
- первая цифра;
- количество делителей;
- сумма всех делителей.

2. Дано n целых чисел. Найти среди них пару чисел, для которых выполняется приведенное ниже условие:

- наибольший общий делитель имеет максимальное значение;
- наименьшее общее кратное имеет наименьшее значение.

Есть ли среди заданных чисел “близнецы”, т.е. простые числа, разность между которыми равна 2?

3. Дано n целых чисел. Написать программу подсчета количества чисел, в записи которых нет цифры 8. При этом требуется использовать функцию вида

```
Function Yes8(x:LongInt):Boolean;
```

Без использования компьютера решите эту задачу для всех трехзначных чисел. Ответ: 729.

4. Даны два числа — n и k ($k < n$). Число n есть основание системы счисления, а k — количество знаков в записи числа. Написать программу подсчета количества натуральных чисел в n -ичной системе счисления, записываемых ровно k знаками. Без использования компьютера решите эту задачу для всех трехзначных десятичных чисел. Ответ: 900.

5. Объявлен конкурс на получение грантов. В конкурсе принимают участие n человек. Гранты выдаются первым пяти участникам, набравшим наибольшее количество баллов. Один участник не может одновременно получить два и более грантов. Написать программу определения количества способов, которыми могут быть распределены гранты. Исследовать область ее применимости.

6. Дана шахматная доска размером $n \times n$ и n ладей. Определить количество способов расположения ладей на шахматной доске так, чтобы они не могли бить друг друга. Исследовать, для каких значений n можно использовать типы данных Integer, Word, LongInt.

7. Дано n различных бусин. Сколько различных ожерелий можно составить из них? Исследуйте область применимости вашей программы.

Примечание. При $n = 7$ число ожерелий равно $(5040 : 7) : 2 = 360$. Делением на 2 учитывается не только поворот (деление на 7), но и переворот ожерелий.

8. Имеются предметы k различных типов. Количество предметов первого типа n_1 , второго — n_2 , ..., k -го — n_k . Написать программу подсчета числа перестановок, которые можно сделать из этих предметов, и исследовать область ее применимости. Предварительно попытайтесь обосновать формулу:

$$P_n = n! / (n_1! \cdot n_2! \cdot \dots \cdot n_k!).$$

Есть текст. Его шифруют (кодируют) с помощью перестановки букв. В результате получается новый текст, называемый анаграммой (слова “лунка” и “кулан” — анаграммы). Для механической расшифровки анаграммы требуется выполнить количество перестановок, вычисляемых по вышеприведенной формуле, где n_i — количество конкретных букв в анаграмме. Предположим, что у вас есть компьютер с миллиардным быстродействием.

вием. Он выполняет миллиард умножений в секунду. Задайте значения величинам $n, n_1, n_2, \dots, n_k$ и подсчитайте, какое время ваш компьютер будет решать задачу.

9. Черепашка перемещается по клеточному полю размером $n \times m$. Ей разрешено двигаться вправо и вверх. В начальный момент времени она находится в нижней левой клетке (А), и ей необходимо попасть в верхнюю правую клетку (В). Сколько различных путей есть у Черепашки? Подсчитайте количество путей для различных значений n и m . Выясните, для каких значений n и m

							В
А							

							B
A							

можно использовать для хранения результата переменные типов Integer, Word, LongInt. Предположим, что для анализа одного пути Черепашки компьютеру с миллиардным быстродействием (миллиард операций в секунду) требуется 100 операций. За какое время он проанализирует все пути для конкретных значений n и m ?

10. Последовательности длины n состояются из t нулей и k единиц, причем таким образом, что никакие две единицы не находятся рядом. Подсчитать количество таких последовательностей (ответ: $C(t+1, k)$). Найти способ перечисления всех таких последовательностей для небольших значений t и k . Например, для $t=3$, $k=2$ это последовательности 00101, 01001, 01010, 10001, 10010, 10100.

Продолжение следует

Подписка круглый год!

Дорогие читатели «Информатики»! Напоминаем вам, что теперь подписка на нашу газету осуществляется круглый год. Имеются четыре способа осуществить подписку.

1. Во всех отделениях почтовой связи РФ. Спрашивайте каталог агентства "Роспечать" на первое полугодие 2001 года. Издания "Первого сентября", в том числе и "Информатика", расположены в алфавитном порядке в разделе "Газеты". По этому каталогу можно осуществить подписку со следующего месяца и на любой срок до конца года с доставкой в школу, на дом или до почтового отделения.

2. Издательская подписка. Периодически, не реже двух раз в месяц, мы публикуем купон на издательскую подписку. В ответ на письмо с заполненным купоном вам будут присланы счет и заполненное извещение для оплаты в любом отделении Сбербанка РФ в течение 10 дней. Во многих случаях издательская подписка оказывается дешевле, так как редакция берет на себя стоимость доставки издания до почтового ящика. В счете будет указана только каталожная цена подписки (40 руб. за месяц).

3. По телефону (095) 249-47-58, с 10 до 17 часов по московскому времени. Вежливый оператор примет заказ и направит по вашему адресу счет, который можно будет оплатить в любом отделении Сбербанка РФ и получать газету с доставкой. И в этом случае редакция берет на себя стоимость доставки по адресу. В счете будет указана только каталожная цена подписки.

4. Через Интернет. Заказ можно осуществить в интерактивном режиме непосредственно на сайте Объединения педагогических изданий "Первое сентября" www.1september.ru.

Ф. СП-1

Министерство связи
Российской Федерации
"Роспечать"

АБОНЕМЕНТ на газету

32291

ИНФОРМАТИКА

(индекс издания)

[illegible]

Куда

(почтовый индекс)

(адрес)

Кому

(фамилия, инициалы)

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ

Me

ли-
тер

на газету

32291

ИНФОРМАТИКА

(индекс издания)

(наименование издания)

[illegible]

Куда

(ПОЧТОВЫЙ ИНДЕКС)

(адрес)

KOMY

(фамилия, инициалы)

Зимние вечера

Информатика для начинающих

Ю.А. Первин,

г. Переславль-Залесский

Содержание выпусков

1 (№ 1/2001), 2 (№ 5/2001), 3 (№ 6/2001), 4 (№ 7/2001),
5 (№ 8/2001), 6 (№ 9/2001) и 7 (№ 10/2001)

Вечер 1. Про программы, пиктограммы и курсор

Вечер 2. Что такое алгоритм

Вечер 3. Строковый редактор

Вечер 4. Побеседуем с компьютером

Вечер 5. Хранить, чтобы помнить и искать

Вечер 6. "Ханойские башни"

Вечер 7. Шахматный этюд

Вечер 8. Исполнители

Вечер 9. "Машинист"

Вечер 10. Компьютерная арифметика

Вечер 11. Классификаторы-"догадалки"

Вечер 12. Классификаторы-"собиралки"

Вечер 13. Промежуточный финиш

ВЕЧЕР ЧЕТЫРНАДЦАТЫЙ. ПЕРЕКРЕСТНАЯ ПРОВЕРКА

Этот вечер у Георгия Аркадьевича тоже состоялся. Но так сложились обстоятельства, что мое место на диване в углу его кабинета в тот вечер осталось незанятым — мне пришлось уехать в срочную командировку.

И только потом мой старый друг рассказал мне, почему он, хотя и не рассказывал детям ничего нового, все же провел это занятие.

Проверить десяток непростых задач у нескольких учеников — нелегкое дело даже для такого опытного преподавателя, как дедушка Фёдоров. Но когда он попросил своих внуков проверить турнирные задачи ребят из других команд Роботландского университета, он думал вовсе не об облегчении своей работы.

Раздел IV. Текстовый редактор

ВЕЧЕР ПЯТНАДЦАТЫЙ. "МИКРОН"

Почти готовое письмо

— Дедушка, а ты знаешь, что наша бабушка завтра утром уезжает? — сказали немного взгрустнувшие Дина и Тим, когда они пришли в кабинет дедушки Фёдорова.

— Конечно, знаю. Я завтра еду ее провожать до автовокзала.

— От того, что ученик смотрит решения другого, — поделился со мной Георгий Аркадьевич, — двойная польза: во-первых, когда он видит отличную работу своего товарища, он получает урок, как надо поступать в трудных для себя случаях; во-вторых, когда он видит ошибку товарища, то своим умением исправить и объяснить ошибку он подтверждает, что сам отлично усвоил материал.

Такой прием (его называют **перекрестной проверкой**) особенно хорош, когда речь идет о коллективе удаленных учеников, с которыми не видишься, и с ними приходится переписываться. Сейчас уже много моих знакомых проводят со своими учениками такие "вечера", рассказывая им о компьютерах и информатике.

Для того чтобы подготовить ребят к перекрестной проверке заданий их сверстников, работающих в сети, мы и провели с ними это занятие.

И он даже рассказал внукам о цели бабушкиного путешествия:

— Бабушка едет в Нижний Новгород. Она давно уже не навещала своего внука Егорку — вашего двоюродного брата. Кстати, вы могли бы написать ему письмо, тогда бабушка сможет ему передать ваше сообщение.

— Вот здорово! — оценила Дина предложение дедушки и уже протянула руку к лежащим на столе шариковой ручке и листу бумаги.

Но дедушка ее остановил:

— Зачем же?

— Что “зачем же”? — не поняла Дина.

— Зачем писать ручкой, когда можно написать письмо, набрав его на клавиатуре?

Дина попыталась возразить:

— Мы-то это письмо увидим на своем экране. А как же его увидит Егорка? Что, бабушка возьмет компьютер с собой, что ли? — попробовала она пошутить.

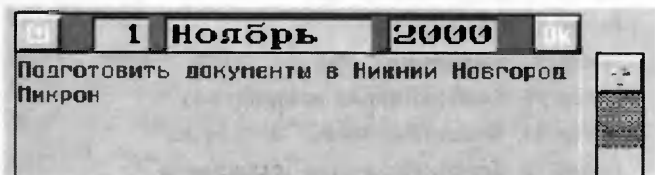
Дедушка оценил шутку:

— Конечно же, нет! Все гораздо проще. Бабушка возьмет с собой листочек бумаги, на котором ваше письмо будет напечатано с помощью вот этого устройства.

И он положил руку на принтер, стоявший на дедушкином столе рядом с компьютером. Хотя дети еще ни разу не пользовались им, они часто видели, как дедушка печатал свои статьи.

— Это принтер! — дружно воскликнули Тим и Дина. — Он напечатает все, что мы напишем на экране.

Тим включил компьютер и увидел на странице программы “Блокнот” короткие дедушкины записи:



У Тима, естественно, возник вопрос:

— Что это такое — “Микрон”?

— Это тема нашего сегодняшнего занятия. Так называется текстовый редактор, в котором можно писать любые тексты, исправлять ошибки и вообще выполнять обработку текстовой информации.

— А разве нельзя написать записку в “Блокноте”? Ведь мы уже писали в “Блокноте” тексты в несколько строк.

— У “Блокнота”, — объяснил дедушка, — довольно много ограничений. Я готов был бы вам рассказать о них, но вы сможете при желании убедиться в этом сами, а “Микрон” вам нужно осваивать все равно: в текстовом редакторе вам нетрудно будет делать такие операции, которые в простом “Блокноте” совершенно невыполнимы.

И дедушка, выйдя из “Блокнота”, попросил Тима найти пиктограмму редактора “Микрон”. Однако картинка, которая появилась на экране, как только Тим вошел в “Микрон”, оказалась не очень выразительной — черное поле во весь экран с двумя голубыми строчками: одна в самом верху экрана, другая, наоборот, в самом низу.

— Перед вами — *рабочее поле* “Микрона”. Единственное яркое место на этом черном поле — маленький белый прямоугольник. Это и есть курсор “Микрона”. Посмотрите, как он слушается вас, когда вы нажимаете клавиши с управляющими стрелками.

Дети взялись за клавиши, а дедушка в это время продолжал:

— Посмотрите, что меняется в нижней информационной строке, когда курсор бежит по рабочему полю.

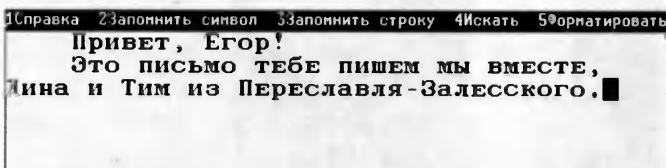
— Ага! Я понял: за словом “строка” светится номер строки, в которой сейчас находится курсор, а после слова “столбец” — номер столбца курсора.

— Значит, эти два числа можно назвать... — начал дедушка и многозначительно остановился, ожидая, что внуки продолжат начатую им фразу.

Дети задумались. И вот Дина воскликнула:

— Это координаты курсора!

Полюбовавшись меняющимися координатами, Тим положил пальцы на клавиши, чуть призадумался и начал писать:



Он приостановился. Набирать длинное и непростое название родного города было нелегко. Он вдруг спросил дедушку:

— А если я ошибусь?

Дедушкин ответ прозвучал как упрек в отсутствии смелости:

— И это говорит человек, который освоил “Правилку”! В “Микроне” действуют все законы этой хорошо знакомой вам программы. Кроме того, не забудь про клавишу стирания, то есть клавишу обратного пробела — BackSpace. Она очень полезна при исправлении оперативных ошибок.

Тим кивнул головой. Мол, все ясно. И задал следующий вопрос — уже по существу:

— А о чем писать?

— Напиши о своих делах, об учебе, об успехах. Напиши о своем городе. Пригласи приехать к нам в гости.

Тим вздохнул:

— Но если писать про Переславль, то придется столько раз набирать длинное слово! А мне и один-то раз это далось непросто.

— Эх, — с улыбкой произнес дедушка Фёров свое любимое изречение: “Все бы проблемы были такими!”

И дал Тиму несколько очень точных указаний:

— Поставь курсор на начало слова.

Тим накрыл курсором букву “П”.

— Обрати внимание на вторую запись в верхней информационной строке.

Тим прочитал:

— “2Запомнить символ”.

— Двойка в этом выражении, — объяснил дедушка, — это порядковый номер клавиши в самом верхнем ряду клавиатуры. Называются эти клавиши *функциональными*. Все другие клавиши — и буквы, и цифры, и знаки препинания — имеют постоянное во всех программах предназначение. А вот функциональные клавиши могут иметь в разных программах различные функции, такие, которые предусмотрел программист при написании этой программы. Так вот, в

верхней информационной строке записаны названия (номера) функциональных клавиш “Микрона” вместе с названиями операций, выполняемых по нажатию этих клавиш. Значит, если сейчас нажать клавишу **[F2]**, то в буфере запоминается буква “П” (потому что на ней установлен курсор).

Отведи курсор на любое свободное место рабочего поля и нажми клавишу переключения регистров.

Тим нажал клавишу **[Shift]** и заметил, что в верхней информационной строке надпись около цифры 2 сменилась: “2Вспомнить символ”.

— Это означает, что, набрав аккорд **[Shift] + [F2]**, ты выведешь на экран содержимое буфера, — сказал дедушка.

Тим тут же убедился в правоте дедушкиных слов:

— Кажется, начинаю понимать! — Тим поставил курсор на вторую букву в слове “Переславль-Залесский”, нажал клавишу **[F2]**, снова перевел курсор в позицию около буквы “П” на ее новом месте и аккордом **[Shift] + [F2]** восстановил букву “е”. — Потом снова пойду за следующей буквой...

— Ну зачем же так сложно, — возразил дедушка, — давай сделаем проще...

Предвидя новое задание, Дина попросила:

— Можно, теперь я?

Тим обменялся взглядами с дедушкой и после этого не очень охотно передал клавиатуру сестре.

Дедушка продолжал:

— Поставь курсор в начало слова “Переславль” и нажми **[F2]**, как делал Тим.

Дина выполнила упражнение.

— Заметь, Дина, что курсор переместился на одну позицию вправо и стоит сейчас на второй букве. Значит, можно нажать на **[F2]** еще раз. И еще. И еще. Так прошлись по всему слову. Теперь в буфере не один символ, а 21. Это вместо с черточкой-дефисом между словами “Переславля” и “Залесского”.

— А теперь, — сообразила Дина, — я на любом новом месте нажму **[Shift] + [F2]** и получу...

Излагая свою мысль, Дина нажала аккорд и с удовлетворением прочитала: “Переславля-Залесского”.

— Точно так же, как было написано у Тима. Даже в том же падеже. Теперь можно легко писать название нашего города сколько угодно раз. Всего-то единственным нажатием. Здорово!

Дина продолжила письмо. Вот уже она добралась до конца. В предпоследней строке она написала: “С приветом”, — а в последней — “Дина и Тим Фёровы”.

— Молодцы, — похвалил дедушка обоих внуков, — ведь письмо, которое вы только что почти закончили, — это первый ваш большой текст, набранный на компьютере.

— Почему это “почти”? — чуть обиженно спросила Дина.

А Тим, наоборот, предложил свои услуги:

— Если надо набрать что-нибудь еще, то теперь моя очередь.

Дедушка согласился с Тимом и сказал детям:

— В начале письма надо было написать отдельной строкой сегодняшнюю дату.

Тим недоуменно спросил:

— Но где же ее написать? Ведь первая строка уже занята. Видишь: в ней стоит курсор — и выше его уже не поднять.

Вместо ответа дедушка, в свою очередь, задал вопрос:

— Ты помнишь, как вставляется символ?

— Да, помню. Надо нажать клавишу вставки **[Insert]**. Тогда строка раздвигается и можно в освободившееся под курсором место вставлять символ.

— Тогда ты сможешь вставить и строку: нажми **[Insert]**, а аккорд **[Shift] + [Insert]**, и ты увидишь...

Но Тим, даже не дослушав до конца, действительно увидел, что перед обращением к Егорке появилась одна пустая строка, а остальные сдвинулись вниз. Это понравилось Тиму, он нажал аккорд еще раз. Теперь перед началом письма стояли две пустые строки. Тим без труда написал в первой строке дату — 7 февраля 2001 года.

— Это хорошо, — сказал дедушка. — Но уж если следовать всем правилам оформления писем, то перед датой надо поставить имя и фамилию отправителя. Или именина, если, как в вашем случае, их два, а не одно.

— Но ведь такую строку Дина уже написала в конце. Вот она.

— Верно. Но это строка подписи. А вам нужна строка заставки — начала письма. К счастью, они совпадают. Этим можно воспользоваться.

— Я знаю, как! — воскликнула Дина. — Надо пройти по строке “Дина и Тим Фёровы”, нажимая все время на **[F2]**, потом вставить одну пустую строчку над датой, а затем на этом месте вспомнить строку из буфера. Правильно?

Но дедушка и тут, вместо того чтобы прийти в восторг от предложения Дины, сделал добавление (впрочем, очень полезное для дальнейших упражнений в “Микроне”):

— Верно. Так можно. Но...

Дина и Тим переглянулись и уже привычно спросили:

— Но можно проще?

— Да. Вот смотрите. В верхней информационной строке есть слова “3Запомнить строку”, а на регистре (и он нажал **[Shift]**) в этом же месте — надпись “3Вспомнить строку”. Вы, наверно, уже догадались, для чего служит клавиша **[F3]**?

Дина высказала предположение:

— Ясно! Теперь можно запомнить всю строку одним нажатием **[F3]**, а затем вспомнить ее на нужном

месте тоже одним нажатием — аккордом **[Shift] + [F3]**.

Она тут же поторопилась осуществить свой совершенно правильный замысел, но... по неосторожности вместо **[F3]** нажала **[Delete]**.

— Ой! — огорченно всхлипнула Дина. — Я нечаянно. Мизинчиком задела. Я совсем этого не хотела.

То, что при этом произошло, Дине уже было понятно: аккорд **[Shift] + [Delete]** стирает целиком всю строку, подобно тому, как аккорд **[Shift] + [Insert]** вставляет строку.

Дедушка это тоже понимал. Изобразив печаль на своем лице, он невесело произнес:

— Строка пропала. Что будем делать?

Тим, сочувствуя сестре, ударился в воспоминания:

— Как было хорошо в простых исполнителях: в “Колнухе”, “Монахе”! Там была возможность вернуться назад, когда сделан неверный ход. Даже “Машинист” умел это делать. Помните? Нам бы сейчас такую возможность.

Тут, оценив, с одной стороны, переживания Дины, а с другой стороны, верное направление мыслей Тима, дедушка Фёдоров решил, что пора менять грустное выражение лица на столь нужную сейчас бодрую уверенность:

— У Тима очень хорошие воспоминания об откатке. И очень уместные. Потому что *откатка* в “Микроне” тоже существует. Более того, она и выполняется точно теми же клавишами, что и у знакомых исполнителей: аккорд из **[Ctrl]** и клавиши обратного стирания — **[Ctrl] + [BackSpace]**. Тебя это должно порадовать, Дина. Точно так же, как и в знакомые вам исполнители, “Микрон” может вернуться не только к результату самой последней операции, но и сделать несколько шагов назад.

Ну-ка, сделай сейчас откатку!

Дина не заставила себя долго ждать. Она тут же нажала **[Ctrl] + [BackSpace]**, и ее глаза, уже готовые заплакать, засияли улыбкой.

— Все в порядке! Спасибо, Тимочка. Спасибо, дедушка! Теперь я смогу завершить письмо Егорке. Вот.

На сей раз очень внимательно и осторожно она повторила операцию запоминания строки (**[F3]**) и восстановления ее на новом месте (**[Shift] + [F3]**).

Вопросы и упражнения

1. Просклоняйте слово “крокодил” во всех падежах, записывая каждый падеж на новой строке. Воспользуйтесь возможностью “Микрона” запоминать информацию в буфере.

2. В первых двух четверостишиях известного стихотворения С.В. Михалкова “Дядя Степа” перепутаны строки:

У заставы Ильича
Самый главный великан
По прозванию Каланча
И по имени Степан

Жил высокий гражданин
В доме восемь дробь один
По фамилии Степанов
Из районных великанов

Восстановите правильный порядок строк в этом стихотворении.

Какими способами это можно сделать?

Как можно добиться результата при наименьшем числе нажимаемых клавиш?

3. В рабочем поле “Микрона” записаны два стихотворения А.А. Барто — “Кораблик” и “Мячик”. Учительница дала задание удалить с экрана стихотворение “Кораблик”, но ученик нечаянно удалил стихотворение “Мячик”. Все четыре строки!

Какую операцию вы порекомендуете выполнить, для того чтобы восстановить стертое стихотворение “Мячик”?

Какие клавиши (или аккорды) и сколько раз вам придется для этого нажать?

Выравнивание строк

Вставив строки, Дина откинулась на спинку кресла, чтобы еще раз полюбоваться представлением письма на экране, и, повернувшись к дедушке, спросила:

— Ну, как тебе кажется, красиво?

— Хм, — ответил дедушка не так, как ожидала услышать Дина, — можно сделать чуточку покрасивее. Вот посмотри, как у тебя написано обращение к Егорке.

— А почему тебе не нравится? — не поняла Дина. — Ни единой ошибочки!

— Это так. Но такое обращение лучше бы написать в середине строки. В твоём письме видно, что ты несколько сместила от края текст первой строки — обращения, но стоит он все-таки не точно по центру.

— Значит, надо сосчитать все пробелы слева и все пробелы справа, а потом вставить несколько пустых символов перед словами “Привет, Егор!”, чтобы число пробелов слева и справа уравнилось.

— Есть совсем простой прием выравнивания слова или выражения строго по центру строки, ничего при этом не подсчитывая. Называется этот прием *центрированием*. Простого клавишного аккорда **[Ctrl] + [Пробел]** достаточно, чтобы текст в той строке, где стоит курсор, аккуратно центрировался.

— Ну, уж если придирается к таким мелочам, — недовольно пробурчал Тим, — то тут можно найти и кучу других недостатков.

— Это каких же? — обиделась Дина.

— Посмотри, с левого края строки ровненькие. Все они, как одна, упираются в край экрана. А справа что творится? Каждая строка заканчивается в разных позициях! Разве же это красиво?

— Ты сделал очень верное наблюдение, Тим, — отметил дедушка. — Края строк, на которые ты обратил внимание, так и называются — *рваные*. Это, согласен, некрасиво. Но для письма родной и любящей вас бабушке все-таки допустимо. А вот при подготовке текста для публикации в книге или газете рваные строки просто недопустимы.

— Да уж. — Тим обрадовался дедушкиной поддержке. — Из газетных строчек получаются такие ровные колонки — приятно посмотреть. А ведь для этого тоже существует какой-нибудь специальный прием?

Дедушка Ферстов с готовностью продолжил так уместно начавшееся обсуждение:

— Вот и дошел черед до *форматирования*. Так называется операция выравнивания строк по ширине. Чтобы ее показать, напомним вам (вы это наверняка знаете), что текст состоит из абзацев.

Тим попытался вспомнить, что говорилось об абзацах в школе на уроках русского языка:

— *Абзац* — это несколько подряд записанных предложений с какой-то общей мыслью, общим содержанием. Чтобы отличить один абзац от другого, первая строка в начале абзаца смещается немножко вправо. — Он показал рукой, в какую сторону должна быть смещена строка.

— Да, — сказал дедушка, не забыв похвалить внука. — Первую строку абзаца называют иногда *красной* строкой, а величину смещения текста в красной строке — *абзацным отступом*. Ты, Тим, сейчас очень верно подметил, что все строки абзаца газетной колонки (кроме красной) имеют одинаковую длину. В “Микроне” (если не делать предварительно никаких настроек, о которых мы еще поговорим) тоже есть принятая длина строки, которая измеряется в количестве символов, а также величина абзацного отступа в красной строке абзаца.

— Я думаю, — предположила Дина, — что длина строки в “Микроне” равна ширине экрана.

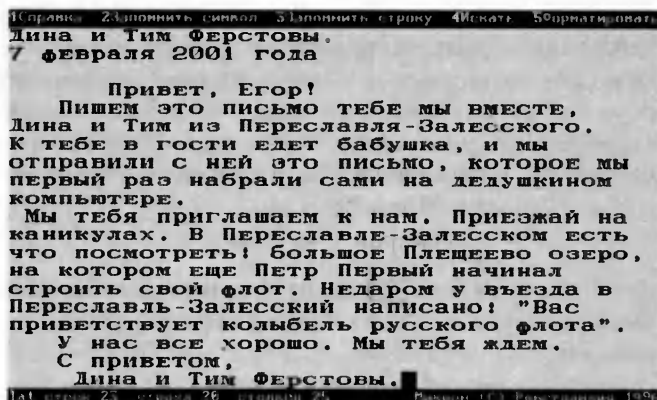
— Совершенно верно, 39 символов в строке и столько же в рабочем поле “Микрона”, — сказал дедушка. — А абзацный отступ равен двум символам.

— И каким же образом можно выровнять строки нашего письма Егорке?

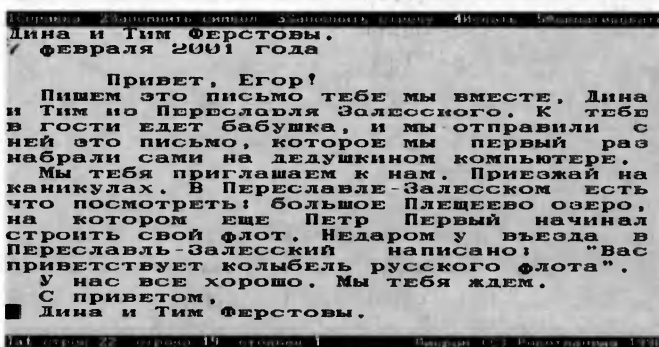
Дедушка снова показал пальцем на верхнюю информационную строку “Микрона”. Там в самой последней позиции Дина прочитала: “5Форматировать”.

— Ага! Значит, надо нажать **[F5]**.

Вот как выглядело письмо Егорке перед операцией форматирования:



А когда форматирование было завершено, письмо стало выглядеть так:



— Да, форматирование — полезная операция, — признала Дина. — Теперь я могу писать текст, не задумываясь о его оформлении, а потом — раз! — и красиво отформатированное письмо. Или сочинение.

Дедушка решил доставить внучке дополнительную порцию удовольствия:

— Коль скоро тебе понравилось форматирование, я покажу тебе еще одну возможность этой операции.

— Надо опять посмотреть в верхнюю информационную строку?

— Да, но на верхнем регистре, то есть при нажатой клавише **[Shift]**.

Нажав клавишу **[Shift]**, Дина увидела на месте “5Форматировать” новую надпись — “5Задать формат”. Она вопросительно подняла глаза на дедушку, и он рассказал:

— С помощью аккорда **[Shift] + [F5]** вы можете изменить формат абзаца, если вам не нравятся стандартная длина строки и величина абзацного отступа.

Тим, молчавший все это время, сразу оживился:

— Как это? Ну-ка, Дина, нажми скорее **[Shift] + [F5]**!

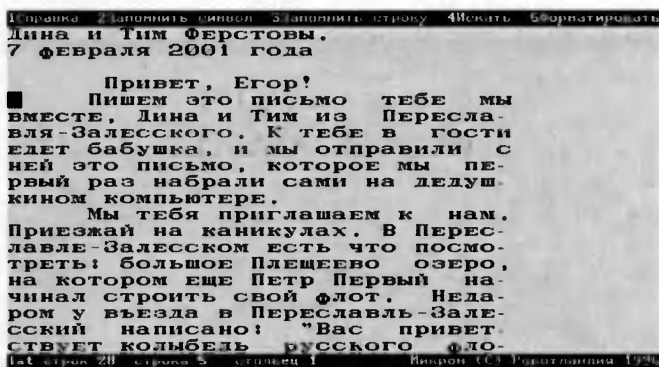
На рабочем поле “Микрона” появилась строка из плюсов и минусов:

— А теперь, введя желаемое количество плюсов, вы можете изменить абзацный отступ, а изменив число минусов, создать для “Микрона” другой *шаблон форматирования*, то есть образец, по которому текстовый редактор выполняет так понравившуюся Дине опе-

рацию. Введите, например, в начале строки не два плюса, а пять. (Заметив вопрос в глазах Дины, он уточнил: "Вот же курсор — в начале строки. Перемещай его и набирай на клавиатуре знак "+""). Так, а теперь сделайте строку на восемь символов короче (например, сотри последние восемь символов клавишей обратного пробела). Что получилось?

— Ну а теперь попробуйте отформатировать письмо Егорке по этому новому шаблону форматирования.

Дина нажала несколько раз **[F5]** (на каждом абзаце письма) и с интересом посмотрела на результат:



Дина решила, что такое представление ей нравится меньше.

— Не беда, — сказал дедушка, — в любой момент ты можешь вернуться к исходному состоянию, восстановив прежний шаблон форматирования.

Вопросы и упражнения

1. Для чего необходимо форматирование текста?
2. Сколько раз надо нажать на клавишу форматирования (**[F5]**), если придется форматировать текст из четырех абзацев? Почему?
3. В приведенном ниже отрывке из рассказа В.Бианки про мышонка Пика — два абзаца. Их надо форматировать. Но при наборе текста допущены две ошибки, которые мешают применить операцию форматирования. Что это за ошибки? Как их исправить?

У Пика получился легкий круглый домик, очень похожий на птичье гнездышко. Весь домик был величиной с детский кулак.

Сбоку мышонку проделал в нем ход, внутри выложил мхом,

листьями и тонкими корешками. Для постели он натаскал мягкого, теплого цветочного пуха. Спаленка вышла на славу.

Принтер

— Ну, теперь-то наконец все! — выдохнули дети после того, как все наставления дедушки Фёрстова были выполнены.

И снова он возразил:

— Но вы же сами хотели иметь письмо на листе бумаги, а не на экране, который бабушка не сможет с собой взять. Чтобы довести дело до конца, нажмите клавишу перехода.

Поскольку в этот момент Дина снова уже сидела за клавиатурой, она так и сделала. В середине экрана, поверх текста, появилось окошко-меню:



— Это меню называется **основным меню** редактора "Микрон". Из нескольких его позиций нам сейчас нужна...

— Печать! — хором воскликнули ребята.

— Прежде чем войти в позицию "Печать", надо убедиться, что в принтере есть бумага, — сказал дедушка, вставил листок и обратился к Дине: "Ну, давай!".

Она нажала на клавишу выполнения, и бумажный листок неторопливо втянулся внутрь устройства и так же плавно появился на выходе из принтера. Но теперь это уже был листок с напечатанным текстом письма.

— Сейчас основное меню пока не понадобится, поэтому его лучше убрать с экрана: еще одно нажатие клавиши перехода.

Вопросы и упражнения

1. Что нужно сделать, чтобы отпечатать текст, написанный в "Микроне"?
2. Напечатайте на принтере свое школьное расписание уроков на неделю.

Справочник

— Дедушка! Ты нам так много рассказал сегодня, а мы ничего не успели записать из-за Егоркиного письма. Нельзя ли о "Микроне" прочитать какую-нибудь книжку? — спросила Дина.

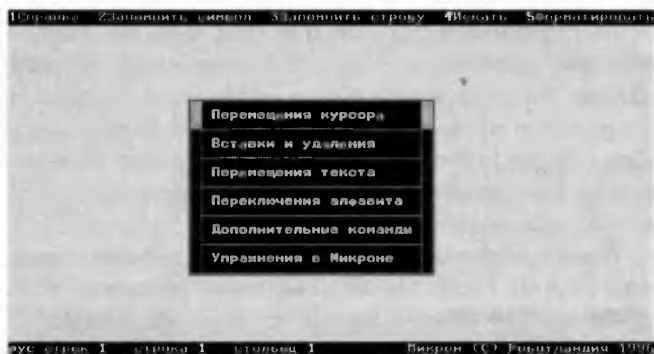
— Можно, — уверенно ответил дедушка. — Такие книжки есть. Но если уж вы сидите за компьютером, то и читать лучше с компьютерного экрана. Не так ли?

Тим за двоих согласно кивнул:

— Только как нам найти в компьютере эту информацию про "Микрон"?

— Еще один взгляд в верхнюю информационную строку, в самое ее начало.

— "1Справка", — прочитал Тим и тотчас нажал **[F1]**.



Он был несколько разочарован: середину экрана заняла таблица, в которой и на этот раз можно было угадать меню.

— Это меню и есть справка по “Микрону”? — с некоторым сомнением спросил Тим.

— Точно. Позиции этого меню — это разделы справочника. Если тебя интересуют, например, приемы вставки и удаления, то надо поставить курсор-подсветку на соответствующую, вторую, строку меню и войти в эту позицию. Тогда вы прочитаете, какими клавишами в “Микроне” надо пользоваться для удаления и вставки символов, слов и строк, для очистки всего экрана.

Заметьте, что некоторые клавиши обозначены в справочнике очень коротко: клавиша выполнения названа по-английски **Enter**. А клавиша перехода еще короче — **Esc**.

Если в окне, где появляется справочная информация по выбранному разделу, не хватает места, то окошко можно “прокрутить” над текстом раздела и просмотреть этот раздел не целиком, а постепенно по нему перемещаясь. Для такого перемещения пользуются управляющими клавишами-стрелками **↑** и **↓**, которые перемещают ползунок на расположенной справа от окна линейке протяжки.



Вы видите, что некоторые сведения из справочника вам уже знакомы. Но большую часть полезной информации из справочника вам придется осваивать самостоятельно.

— Боюсь я этого справочника, — призналась Дина. — Если нужная информация потребуется мне где-нибудь в конце работы, тогда меню справочника или окно выбранного раздела полностью закроют на экране мой текст и придется все набирать заново.

— Вот этого бояться не надо. На каком бы этапе работы с “Микроном” ты ни обратилась к справочни-

ку, после выхода из него (даже если ты будешь читать его долго) программа возвращает тебя в точности на то место, из которого ты запросила справку. При этом никакого искажения информации не происходит. Можешь смело нажимать **F1**.

Вопросы и упражнения

1. Прочитайте разделы микроновского справочника. Отметьте в уме ту справочную информацию, которая вам уже знакома.

2. Испытайте, как выполняются операции, не описываемые в этой книге, но представленные в справочнике, например, стирание слов, расположенных непосредственно слева или справа от курсора.

3. С какой целью используются и как управляются окна в разделах справочника?

ВЕЧЕР ШЕСТНАДЦАТЫЙ. ФАЙЛЫ И КАТАЛОГИ

Диски и файлы

Когда Дина и Тим усадились в кресла, дедушка начал с объявления о том, что на одном из ближайших занятий их ждет серьезная творческая работа — участие в конкурсе микроновских сочинений.

У Тима это известие вызвало беспокойство, которым он поспешил поделиться с Диной и в первую очередь с дедушкой:

— Помните: все прошлое занятие мы потратили на то, чтобы написать короткое письмо Егорке. И то еле-еле успели. А уж на сочинение нам точно времени не хватит. Что посоветуешь нам, дедушка?

— Совет простой: надо будет посидеть за компьютером не один раз, а несколько. Но для этого, как вы понимаете, надо уметь сохранять в памяти ранее написанные тексты. Тогда можно писать продолжение за продолжением, а не начинать каждый раз заново. И хотя совет простой, чтобы им воспользоваться, вам потребуются новые важные знания.

Давайте возьмем какой-нибудь знакомый вам текст, например, стихотворение Н.А. Некрасова “Мужичок с ноготок”. Одно четверостишие этого стихотворения вы напишете сравнительно быстро (хотя, конечно, не за полминуты). Начни-ка ты, Тим.

Через некоторое время с клавиатуры Тима на экран перекочевали стихи:

Однажды в студеную зимнюю пору
Я из лесу вышел; был сильный мороз.
Гляжу: поднимается медленно в гору
Лошадка, везущая хворосту воз.

Теперь представьте себе, что к тому моменту, когда Тим закончил набирать эту порцию информации (большую или маленькую — сейчас это для нас неважно), наше время работы за компьютером подошло к концу.

Ну, например, мама вам говорит, что пора спать, или мне сообщают по телефону, что надо срочно приехать в редакцию.

Дина выступила с гипотезой:

— Может быть, отпечатать на принтере, чтобы в следующий раз было проще набирать?

Тиму эта идея явно не понравилась:

— Но набирать-то все равно придется. Притом тот же самый текст. Это скучно.

К тому же и дедушка поддержал Тима:

— А если речь будет идти не о четверостишии, а о целой поэме? Нет, это предложение не годится. Здесь нужно попросить компьютер записать информацию в свою память на долговременное хранение. Тогда, если информация будет записана, компьютер можно даже выключить, не опасаясь, что записанное стихотворение пропадет.

Тим заинтересовался одним словом в дедушкиной речи:

— Ты хочешь сказать, что хранение бывает долговременным и недолговременным?

— Да, — ответил дедушка, — есть два вида *памяти*, или *запоминающих устройств* компьютера. *Оперативная*, или *внутренняя*, память — это часть компьютера, которая хранит информацию только тогда, когда компьютер включен. И хотя у современных компьютеров оперативная память не такая уж маленькая, ее, конечно, недостаточно, потому что при обработке информации на компьютере всегда возникают ситуации, при которых информация должна быть записана и сохранена на время отключения компьютера. С такими задачами справляются *внешние запоминающие устройства*. Вы, конечно, много раз видели *дискеты*. Они относятся к числу долговременных, или внешних, запоминающих устройств. Дискеты иногда еще называют *гибкими дисками*.

— Это потому, что их можно сгибать?

— Ни в коем случае! — очень серьезно предупредил дедушка. — Даже никогда и не пробуйте их сгибать. Результат будет самый печальный. Но название гибких дисков дано дискетам не случайно. Обозначает оно не возможность изгибов, а возможность очень гибкого использования: записав информацию на дискету, вы можете вынуть ее из компьютерного *дисковод*, положить в карман, уехать с ней в другой город и прочитать там информацию с дискеты на другом компьютере.

— Если их называют дисками, то они должны были бы быть круглыми. Почему же дискеты квадратные?

— А вот тут ты не прав. Дискеты, конечно, круглые. Квадратными являются не сами дискеты, а чехлы, в которые они встроены.

— Какие же еще бывают долговременные запоминающие устройства, кроме дискет? — Вопросам Тима, казалось, не будет конца.

Однако дедушку эти вопросы вовсе не раздражали, он отвечал с удовольствием:

— Наиболее часто используемые внешние запоминающие устройства — это так называемые *жесткие диски*. Жесткими их называли, опять-таки, не из-за их твердости и прочности, а потому что они жестко встроены в корпус компьютера, и снять их оттуда, отвинтив винты креплений, намного труднее, чем вынуть дискету из дисковод.

Дина проявляла меньше интереса к гибким и жестким дискам. И она заметила дедушке, что они с Тимом отвлеклись от более важного, на ее взгляд, вопроса:

— А как же можно попросить компьютер, чтобы он записал информацию на диск? — спросила Дина.

— Следи внимательно за тем, что сейчас будет делать Тим, — сказал дедушка, отвечая Дине. И обратился к Тиму:

— Вызови основное меню.

— Это мы уже делали, — ответил Тим, нажимая клавишу перехода.

— Да, действительно. Приходилось войти в позицию "Печать", когда печатали письмо Егорке. А какая позиция нужна сейчас, как вы думаете?

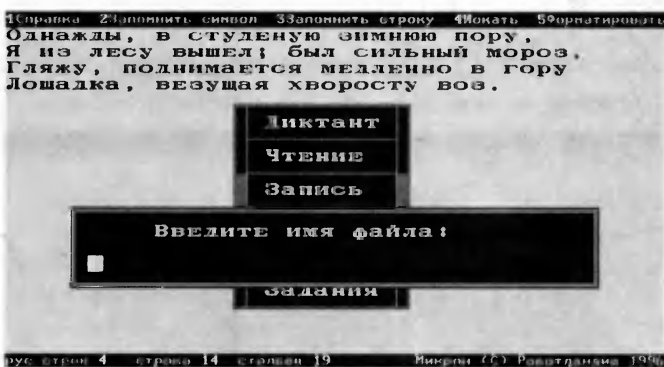
— Я уверена, что надо войти в "Запись", — сказала Дина.

— Правильно говоришь, Диночка, — подтвердил дедушка и сказал Тиму:

— Сделай то, что предложила Дина.

— Сам знаю! — недовольно ответил Тим.

Дальше события развивались по-новому. В середине экрана поверх основного меню легло небольшое окошко с курсором и указанием, которое ярко высвечивалось строкой выше — над курсором:



— Файл? А что такое файл?

— *Файл* — это хранимая на внешнем запоминающем устройстве и именуемая порция информации. Чтобы отличить одну порцию от другой, компьютер должен знать их *имена*, которые придумывает для этой хранимой на диске информации человек, ее использующий, — пользователь. Сейчас на экране видна порция информации, которую мы хотим записать на диск, поэтому естественно, что компьютер спрашивает пользователя, который вошел в позицию Запись, под каким именем он хочет сохранить в памяти файл — стихотворение Н.А. Некрасова. Что бы вы ответили?

Сначала ребята растерялись, но потом Тим проявил инициативу:

уже не произвело на детей такого впечатления, как при первой очистке. Ребята затихли в ожидании новой информации. Так зрители замирают в ожидании очередного фокуса выступающего на сцене мага.

Но дедушка не спешил показывать фокусы. Он стал говорить простые и понятные слова:

— Диск, на котором хранятся файлы, можно сравнить с книгой — сборником рассказов. В книге всегда существует оглавление; оно состоит из названий входящих в книгу рассказов. Заглянув в оглавление, легко обнаружить нужное название, по указанному номеру страницы найти рассказ и прочитать его. Роль книжного оглавления на диске играет список имен файлов, называемый *каталогом*.

Впрочем, это сравнение лучше с самого начала уточнить. Дело в том, что если сборник состоит из нескольких разделов, то нумерация страниц в книге все равно единая, сквозная. А на диске каждому разделу соответствует свой отдельный каталог. Значит, чтобы увидеть имя разыскиваемого файла, надо сначала раскрыть включающий его раздел-каталог.

Вот такой каталог — список имен файлов — я и хочу вам показать. Дина, вызови, пожалуйста, еще раз основное меню “Микрона”.

Эту просьбу было выполнить легко. Вызвав на экран основное меню, Дина тут же поставила подсветку в позицию “Каталог” и, не ожидая указаний, нажала клавишу выполнения.

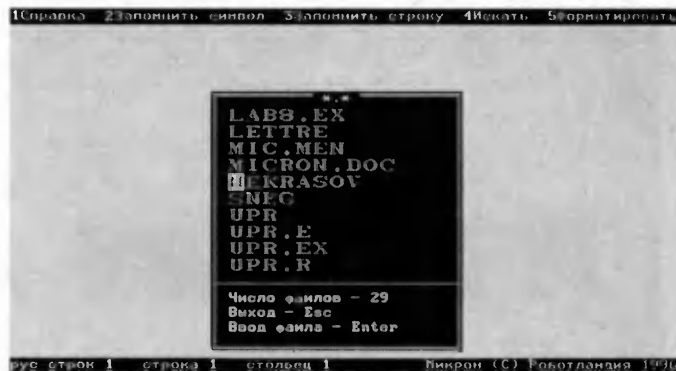


Звездочки здесь означают желание пользователя посмотреть все файлы каталога. Чаще всего именно этого и хочет пользователь. Тогда ему остается для подтверждения своего намерения нажать клавишу выполнения.

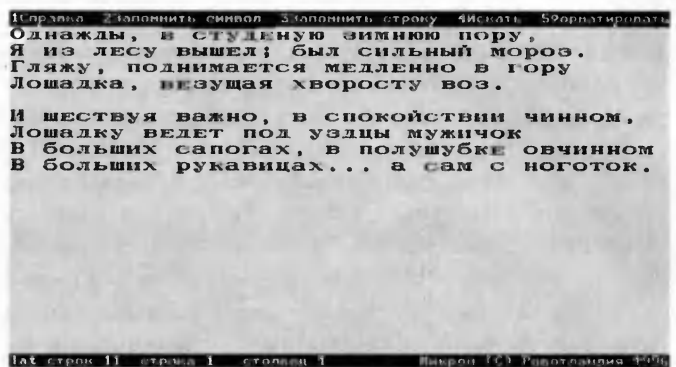
Дина так и сделала:



— Ну а теперь нет ничего проще найти в этом каталоге нужный вам файл. В этом списке надо перемещать курсор стрелкой (↓) до тех пор, пока в окошке не появится имя файла. Все имена внутри каталога упорядочены по латинскому (увы, не русскому!) алфавиту (произносятся эти слова, дедушка листал каталог), значит, мы сейчас до него доберемся. Вот он, наш файл, — сказал дедушка, остановив курсор на имени NEKRASOV:



Когда курсор в каталоге показывает имя нужного файла, нажатие клавиши выполнения вызывает этот файл на экран.



— Это что же получается? Можно прочитать файл, вовсе и не входя в позицию “Чтение”, а главное, не набирая имя файла на клавиатуре? — высказался Тим, внимательно следивший за действиями дедушки.

— Да. Хорошо, что ты заметил эту полезную особенность операции “Каталог”. Если вы забыли, как правильно пишется имя файла (а забыть нетрудно, когда вы пишете длинные имена, да еще и латинскими буквами), то взгляд на список имен сразу напомнит вам забытое слово, и вы читаете файл, не тратя времени на клавиатурный набор его имени.

Вопросы и упражнения

1. Когда (в какой момент) в каталоге — списке имен файлов — появляется новое имя?

2. Вызовите на экран файл с именем GOROD, дайте тексту заголовок (например, Берёзовск, по имени описываемого в этом тексте города), а потом сохраните измененный текст в файле с именем GORODOK. Убедитесь (посмотрев каталог), что сейчас на диске хранятся оба файла — и GOROD, и GORODOK.

3. Напишите записку своему товарищу, приглашая его на день рождения. Отпечатайте приглашение на принтере. Сохраните эту записку в файле DRUG. Используйте тот же файл, для того чтобы распечатать еще одно приглашение другому гостю.

4. Почему вызвать файл из позиции “Каталог” основного меню удобнее, чем из позиции “Чтение”?

Как назвать файл

Обращаясь к Дине, дедушка сказал:

— Я обещал ответить на вопрос, который ты мне задала, — про нерусские имена файлов.

И он прочитал целую лекцию о правилах именования файлов в “Микроне”.

— Да, действительно, у текстового редактора “Микрон” есть одно неудобство: в именах файлов можно использовать только латинские буквы. Можно писать, например, такие имена, которые, наверно, я могу вам не переводить на русский:

CAT
LIST
HOME

Разумеется, русские слова тоже можно применять, но только в том случае, если они написаны латинскими буквами (хотя в повседневной жизни это выглядит довольно непривычно):

KOT
SPISOK
DOM

Правда, в именах могут присутствовать не только буквы, но и цифры, а также некоторые из имеющихся на клавиатуре специальных символов:

SPISOK1
2001
BLACK.CAT

В имени файла запрещены пробелы: каждое имя представляет собой **единое** слово, единый набор символов; не разрешается писать имя, составленное из двух (или более) слов.

Вот примеры недопустимых имен:

NEW FLAT
RED DOG
2001 YEAR

Впрочем, если важно подчеркнуть, что в имени файла фигурируют два слова, это можно сделать без особого труда: надо поставить между словами вместо пробела некоторый разрешенный специальный символ, например, знак подчеркивания “_”. Так, можно использовать имена файлов

BIG_CAT
RED_DOG
2001_YEAR

“Микрон” не различает заглавные и строчные буквы в именах файлов. В трех парах, которые я сейчас запишу, оба имени воспринимаются программой одинаково:

cat и CAT
dog и Dog
spisok и SpiSoK

Правила написания имен дают широкую свободу в выборе достаточно произвольных наборов символов. Следует разумно пользоваться этой свободой, чтобы сделать имена удобными не столько для машины, сколько для человека — содержательными, понятными и выразительными. Так, в качестве имени для списка учеников 2-го класса “А” можно взять, например, “слово” ZY56_7WD (правилами это разрешено!), но гораздо удобнее это имя записать KLAS2A.

Один из специальных символов, разрешенных внутри имен, заслуживает того, чтобы о нем сказать особо. Это точка “.”. Внутри имени может быть не более **одной** точки. Но единственная точка, если она присутствует в имени, должна быть обязательно **внутри** имени: точка не может стоять ни в начале имени, ни в конце его. Вот примеры неразрешенных имен файлов:

NEW.2001.YEAR
HOUSE.
DOG

У этого правила есть разумное обоснование: имена файлов состояются из двух частей — **основы имени** и его **расширения**. Основа является обязательной частью имени, а вот расширение может сопровождать основу, но может и отсутствовать. Когда расширение присутствует в имени, оно следует за основой и отделяется от нее точкой. Вот примеры имен без расширений:

BLACK_CAT
SPISOK
TEXT

А имена

KLASS5A.TXT
CAT.ANI

имеют расширения. (Записывая два последних имени, дедушка выделил их, написав пожирнее.)

Расширения (если они есть) могут содержать не более трех символов. Расширения полезны, чтобы отметить тип файла. Так, для текстовых файлов рекомендуется расширение TXT, для файлов с музыкальными мелодиями — MUS, для файлов, которые хранят рисунки и другие графические изображения, — PIC (от англ. *picture* — “картинка”), PCX или BMP и т.д.

Существует ограничение и на длину основы имени: она не может быть длиннее восьми символов. Поэтому, например, имя BEGEMOT.TXT допустимо, а слово HIPPOPOTAMUS.TXT именем быть не может.

Вопросы и задания

1. Нужно ли знать английский язык, чтобы записывать имена файлов? Почему?

2. Какие правила написания имен нарушены в следующих наборах символов:

ТАБЛИЦА 12.34.56.78
LIST.DLINA CAT AND DOG

В перечисленном ниже списке слов выделите те, которые могут служить именами файлов. Объясните ваши решения.

ELEPHANT.AFRICA SLON.AFR СЛОН.АФР
SLON_AFRICANSKY ELEPHANT.AF SLON.A.F

Объясните, почему слово BLACK.CAT может служить именем файла, а BLACK_CAT — нет.

Книга для учителя

ЗАНЯТИЕ 14. ПЕРЕКРЕСТНАЯ ПРОВЕРКА

Тема занятия. Перекрестная проверка турнира “Компьютерная смекалка”.

Цель занятия. Научить детей оценивать завершённые работы и использовать рефлекссию учебной деятельности.

План

1. Обсуждение целей и задач перекрестной проверки.
2. “Модельная проверка” работ в команде.
3. Организация коллективной работы по проверкам.

Программное обеспечение. Факультативно.

Комментарий

1. Обсуждение целей и задач перекрестной проверки

Перекрестная проверка индивидуальных и командных работ оказалась действенным дидактическим инструментом в дистанционном обучении. Она преследует две педагогические задачи:

- закрепить сформированные в учебном процессе навыки и умения, опосредовав их в анализе завершённых собственных работ и работ своих однокурсников;
- существенно повысить объективность оценок при проверке большого числа заданий.

Решая первую из этих задач, дети, по существу, продолжают изучение завершающейся темы. Они вновь и вновь возвращаются к недавно изученным понятиям и освоенным технологическим приемам. При этом, рассматривая присланные на проверку работы однокурсников, учащиеся имеют возможность сравнить полученные ими результаты с результатами своих товарищей. Оба результата такого сравнения — положительный и отрицательный — имеют высокую педагогическую ценность:

- если учащиеся обнаруживают, что их работы отличаются от остальных в лучшую сторону, — это лучший аргумент столь нужного молодому человеку самоутверждения;
- если же собственные результаты уступают по каким-то показателям решениям сверстников, такой исход можно рассматривать как наименее обижавшую подсказку к необходимому исправлению погрешностей или попросту совершенствованию проекта.

Важен и воспитательный аспект перекрестных проверок — научить ребенка относиться к незнакомому человеку точно так же, как он хочет ощущать отношение окружающих к себе самому.

Отмеченная выше объективность оценок перекрестных проверок объясняется не только увеличением ко-

личества проверяющих, но и, что менее важно, строгостью учеников-проверяющих. Многолетняя практика показывает, что оценки экспертов методической комиссии дистанционного центра оказываются, как правило, более высокими, чем оценки, выставляемые детьми в перекрестных проверках. Этот казавшийся поначалу парадоксальным факт имеет под собой реальное психологическое обоснование: для методистов центра все команды потенциально равны, тогда как ученики-проверяющие видят в каждой проверяемой команде своего соперника.

Сегодняшняя, первая в курсе, перекрестная проверка сравнительно проста. Во всяком случае, каждое задание оценивается здесь единственной оценкой (в отличие от экспертных многокритериальных оценок, используемых в последующих конкурсах). Тем не менее для отработки новых навыков этой простой перекрестной проверке выделено отдельное занятие. Последующие проверки встраиваются в текущие тематические занятия в качестве одного из фрагментов.

2. “Модельная проверка” работ в команде

Перекрестные проверки проводятся на материалах далеких друг от друга и незнакомых детей. Эту важную и новую по форме деятельности работу следует подготовить, моделируя ее в коллективе своей команды (если есть время, то такую моделирующую проверку можно провести перед текущим занятием). Для такого моделирования практически нет количественного ограничения: даже дедушка Фёрстов, работая всего с двумя внуками, может организовать модель перекрестной проверки, попросив Дину и Тима проверить работы друг друга. Еще проще реализуется такая модельная ситуация в реальных командах, насчитывающих более десятка школьников.

“Моделирующей” проверке рекомендуется предпослать самооценку выполненных заданий. Каждый ученик или, еще лучше, группа детей, совместно решавших задачу, выставляют оценки за собственную работу по обсужденным еще на прошлом занятии критериям. С одной стороны, такая самооценка является в чистом виде рефлексией, роль которой в дистанционном обучении высоко оценивают многие педагоги и психологи, а с другой стороны, такая самооценка является хорошей технологической репетицией к предстоящей объемной работе по перекрестным проверкам.

Детям можно объяснить значимость самооценки как формального критерия, по которому можно будет впоследствии (после завершения перекрестных проверок и подведения итогов конкурса) более или менее объективно (и, во всяком случае, количественно) сравнивать свои работы на фоне итоговых таблиц всего курса.

3. Организация коллективной работы по проверкам

Дети уже знакомы с критериями. На сегодняшнем занятии им предстоит не только оценить работы сверстников, но и зафиксировать эту оценку. Большое количество одновременно проверяемых работ заставляет использовать таблицы. Не выходя на уровень обобщений в технологии обработки таблиц, сейчас важно показать полезные практические приемы работы с таблицами.

Определяются назначения строк и столбцов таблицы (строки — команды, столбцы — конкурсные задания). И хотя незаполненные таблицы учитель готовит до урока, рассмотрение структуры таблицы должно быть воспринято детьми как результат их коллективного обсуждения технологии проверок. Учитель подчеркивает компактность именования строк и столбцов таблицы: не название или город команды, а ее регистрационный номер (по существу, код), не формулировка и даже не название задания, а его номер. Такая компактность и формализованность информации особенно важна потому, что заполненные таблицы будут обрабатываться (в дистанционном центре) с помощью компьютеров.

От детей не требуется проводить суммирование по строкам. Для них это избыточная нагрузка. Такую работу компьютер дистанционного центра выполнит без труда (для этого тоже, в частности, необходимо табличное представление результатов проверки). Если команда тем не менее ощущает необходимость подведения итогов в проверяемых командах (например, для сравнения с результатами самооценки), то арифметику суммирования должен взять на себя учитель (не отказываясь от инструментария — калькулятора или компьютера).

После завершения ученической перекрестной проверки турнира “Компьютерная смекалка” остается еще один этап — отправление результатов куратору курса. Сейчас этот этап можно не афишировать (если этого не потребуют любознательные ученики). Привлечь внимание школьников к технологии отправления сообщения в методический центр имеет смысл позднее — при обсуждении темы “Электронная почта”.

ЗАНЯТИЕ 15. ТЕКСТОВЫЙ РЕДАКТОР “МИКРОН”

Тема занятия. Основные операции текстового редактора “Микрон”.

Цель занятия. Познакомить детей с первичными навыками обработки текстовой информации.

Программное обеспечение. Текстовый редактор “Микрон”.

План

1. Актуальность и значимость текстовой обработки информации.

2. Структура рабочего поля редактора, основные операции текстового редактирования.

3. Форматирование в “Микроне”.

4. Справочник.

Комментарии

1. Актуальность и значимость текстовой обработки информации

Возвращение к задачам по обработке текстовой информации на уровне текстового редактора после

- освоения строкового редактора в “Правилке”;
- закрепления приемов строкового редактирования в диалоговых программах “Привет” и “Сказка” и в исполнителях;
- работы на текстовых страницах “Календаря” и “Записной книжки” в “Блокноте” представляет собой выход на новый виток дидактической спирали в такой фундаментальной теме курса информатики, как “Обработка текстов”.

Учитель должен объяснить этот интерес к теме, вызванной прежде всего широтой предметной области, в которой используются понятия, методы и приемы обработки текстовой информации. В школьной информатике эта тема методически важна потому, что, с одной стороны, она позволяет раньше, чем в других темах конструктивно представить детям такие фундаментальные понятия алгоритмического направления информатики, как именование информационных объектов, файл, каталог, редактирование, а с другой стороны, практически ввести ребенка в мир информационных технологий на примере прикладных задач текстового редактирования.

Вынужденно скромный объем курса “Зимние вечера. Информатика для начинающих” определил жесткий отбор инструментальных средств для этой темы. Впрочем, это не представляется опасным, поскольку наш курс исходно рассматривается как элемент непрерывного информатического образования, в котором по уже упоминавшемуся принципу дидактической спирали предполагается многократное возвращение к проблемам обработки текстовой информации.

В частности, поэтому основным инструментом темы выбран проверенный педагогической практикой Роботландии учебный адаптированный текстовый редактор “Микрон”, ценность которого видится здесь не столько в простоте используемых механизмов редактирования и структуры файловой системы, презентабельности шрифта и минимизации системы управления и настройки, сколько в учебной ориентации редактора и прямом совмещении его с прикладными технологическими задачами компьютерного конструирования на последующих курсах дистанционного Роботландского университета, рассчитанных на более взрослый контингент обучающихся.

Понятно, что многие элементы “Микрона” в современной практике школьной информатики требуют обновления. Использование этого программного средства в рамках

небольшой, но весьма насыщенной темы в курсе “Зимние вечера. Информатика для начинающих” позволит выработать обоснованный набор рекомендаций по обновлению учебного адаптированного текстового редактора.

На первом занятии по “Микрону” вопросам актуальности текстовой обработки информации отводится роль краткого введения.

2. Структура рабочего поля редактора, основные операции текстового редактирования

Более подробно и внимательно обсуждаются понятия, связанные со структурой рабочего поля редактора. Эти понятия необходимы в сегодняшнем и последующих занятиях. Кроме того, они относятся к инвариантам редактирования, не зависящим от типа редактируемой информации: понятия рабочего поля, окна, информационного строк, координат активной точки и проч. будут столь же актуальны в последующем изучении графического редактора и других технологических систем за рамками курса.

Общая рекомендация: при изучении темы, связанной с обработкой текстов, легко скатиться к “кнопчному” изложению, когда беседа учителя вместо описания методов обработки информации сводится к рассказу о назначении отдельных клавиш. Это опасно! Следует уделять основное внимание инвариантам информационного редактирования, а не конкретной реализации механизмов такого редактирования.

На примере некоторого текста — письма юному родственнику — дедушка Фёров показывает такие важнейшие операции, как копирование и перемещение информационных элементов, символов и строк. С этой целью он использует операции, “запрограммированные” на функциональных клавишах и отмеченные в информационной строке рабочего поля.

Эти операции важно не только продекларировать и показать. Они должны быть закреплены в упражнениях учащихся. Поэтому настоятельно рекомендуется выполнить все те упражнения, которыми завершается каждый из пунктов-параграфов в занятии. Вероятно, нет надобности повторять в рабочем поле письмо Егорке, однако все три упражнения в конце параграфа — склонение имени существительного по падежам, перестановка строк и восстановление стёртого стихотворения — следует выполнить полностью (а не только убедиться в понимании детьми изучаемых механизмов). При этом все три упражнения (они невелики по объёму клавиатурного набора) лучше не привязывать к хранящимся на диске текстовым файлам, а непосредственно ввести эти тексты под диктовку учителя. Последнее из этих упражнений ориентировано на демонстрацию и закрепление откатки. Понятие откатки знакомо школьникам по ранее изученным программным исполнителям, однако по-настоящему откатку дети смогут оценить именно в текстовой обработке информации. Откатка должна убедить их, что они могут чувствовать себя смело при клавиатурных операциях. Многих учеников откатка раскрепостит.

3. Форматирование в “Микроне”

Операции форматирования в текстовом редакторе очень наглядны. Демонстрации их актуальности может служить любая газетная полоса или страница книги с ровными колонками текста. Можно продемонстрировать и обратную картину — текст с “рваными” правыми краями в рабочем поле “Микрона”. Наглядность операций форматирования упрощает функции учителя.

Совсем легко объяснить центрирование заголовков.

Несколько более трудоемко объяснить операцию форматирования. Здесь предварительно потребуют объяснения такие понятия, связанные со структурой текста, как абзац, красная строка, абзацный отступ. К этим структурным понятиям добавляется шаблон форматирования. Учитель должен его продемонстрировать и раскрыть смысл символов, используемых в шаблоне форматирования.

В качестве объекта для демонстрации и эксперимента можно предложить файл GOROD — текст о городе Берёзовске (который будет использоваться и на следующем занятии тоже). Надо обеспечить появление этого файла на экранах ученических машин, но без привлечения внимания к дисковым операциям. Это не значит, что учитель должен демонстративно покрывать эти операции таинством магии, однако пока они не требуют акцентирования.

Текст удобен тем, что он обозрим, структурирован на абзацы. Тем не менее учитель вправе использовать любой другой дидактический материал, удовлетворяющий примерно тем же требованиям.

На примере этого текста дети должны проделать несколько операций форматирования, определить область действия операции форматирования (один абзац), обнаружить и объяснить (возможно, при вмешательстве учителя) некоторые типовые ошибки форматирования, которым посвящены контрольные вопросы и упражнения в конце параграфа.

Технически простым, но волнующим моментом урока служит вывод текста на принтер. Для того чтобы дети прочувствовали эмоциональную составляющую этой операции, нужен некоторый текст, который заинтересует учеников. Дедушка Фёров использовал для этой цели письмо внуку Егорке. Но у любого учителя таких поводов для демонстрации возможностей принтера предостаточно. Например, открытка, поздравляющая с днем рождения одного из учеников, или визитная карточка каждого сидящего у компьютера школьника. Здесь, правда, придется потратить некоторое время на демонстрацию визитной карточки и рассказ о ее роли в обществе. Зато психологический эффект от использования принтера для печати визитки, безусловно, окупит этот расход учебного времени.

За этим внешним эффектом надо не потерять демонстрацию первого обращения учеников к важному элементу микронового интерфейса — основному меню “Микрона” (позиция “Печать”).

4. Справочник

Рассказ о справочнике “Микрона” не должен вызывать трудностей. Учитель показывает приемы управления справочником, рассказывает о его иерархической структуре, меню, окне и его элементах — протяжке и ползунке. Демонстрируются те строки справочника, которые содержат уже знакомую детям информацию (простейшие перемещения курсора, удаления и вставки, операции форматирования). Все демонстрации имеет смысл делать на фоне заполненного рабочего поля, чтобы убедить детей в том, что вход в справочник и возвращение из него в рабочее поле не искажают текст.

Чтобы закрепить умение работать со справочником (а заодно и расширить набор представлений о редакторе) можно предложить детям игру, которую они ведут, разбившись на пары или тройки. Каждая группа находит в справочнике некоторую пока не обсуждавшуюся на занятии информацию и предлагает другой группе отыскать, понять и объяснить эту информацию. Разумеется, эта игра допустима только при самой активной роли учителя в качестве консультанта обеих групп — как формулирующей вопрос, так и отвечающей на него.

ЗАНЯТИЕ 16. ФАЙЛЫ И КАТАЛОГИ

Тема занятия. Файлы и операции с ними.

Цель занятия. Познакомить школьников с понятием файла и правилами именования файлов.

Программное обеспечение. Текстовый редактор “Микрон”.

План

1. Оперативная память и внешние запоминающие устройства. Дискеты и диски.
2. Сохранение файлов на внешнем запоминающем устройстве.
3. Чтение файлов и каталог.
4. Правила формирования имен.

Комментарии

1. Оперативная память и внешние запоминающие устройства

Дискеты и диски

Различие между оперативной памятью и внешними запоминающими устройствами легко показать, моделируя конец школьного урока, когда предстоит выключение компьютеров. Ничто не мешает разыграть такую “модельную” ситуацию в начале урока. Примерно так поступил дедушка Фёрстов со стихотворением Н.А. Некрасова “Мужичок с ноготок”, которое набирали на клавиатуре его внуки.

Без претензий на строгость важнейших терминов из архитектуры компьютера — оперативная память, внешние запоминающие устройства, гибкие и жесткие диски — можно воспользоваться из текста учебника.

2. Сохранение файлов на внешнем запоминающем устройстве

Центральная ситуация занятия возникает в тот момент, когда учитель просит школьников сохранить на диске компьютера информацию, созданную в рабочем поле. Диалог с компьютером по поводу записи на диск требует осознанного использования понятия файла. В этот момент дедушка Фёрстов на некоторое время приостанавливает динамический процесс информационного обмена, для того чтобы определить, что такое файл.

Файл — это хранимая на внешнем запоминающем устройстве и именуемая порция информации.

Это определение не может с первого чтения прочно осесть в головах юных учеников, тем более что один из важнейших элементов этого определения — имя файла — еще подлежит толкованию и уточнению. Поэтому к этому определению придется еще несколько раз вернуться в ходе сегодняшнего занятия, например, при объяснении двух следующих операций основного меню редактора — “Чтение” и “Каталог” и, конечно, при подведении итогов занятия.

Для того чтобы осуществить первую из дисковых операций — “Запись”, не потребуется излагать свод правил формирования имен в “Микроне”. Однако от важнейшего ограничения — в именах не допускаются русские буквы, можно писать лишь латинские — уйти невозможно. Важный аргумент, оправдывающий такое ограничение, — это понимаемая детьми необходимость переноса файлов с одной машины на другую, при которой приходится учитывать, что многие из используемых в школах машин (прежде всего старые машины) не могут воспринимать русские буквы в именах файлов.

3. Чтение файлов и каталог

Итак, файл записан. Теперь последует переход к операции “Чтение”. Эту операцию рекомендуется первый раз произвести самому учителю. Дело не столько в новизне этой операции для учащихся, сколько в том, что у учителя, как правило, в распоряжении самый быстрый компьютер в классе. Скорость нужна для того, чтобы без лишних потерь времени вернуться в “Микрон” после полного выключения и нового включения или хотя бы перезагрузки компьютера. Между тем для “чистоты эксперимента” такое выключение демонстрационного компьютера желательно: если обратиться к редактору с просьбой прочитать с диска нужный файл, не перезагружая предварительно компьютер, а ограничившись, например, полной очисткой экрана (аккордом **[Shift] + [Ctrl] + [O]**), то при входе в позицию “Чтение” редактор “Микрон” выводит в рабочее поле окно, в котором уже высвечено имя файла, который был использован последним в операциях информационного обмена. Следовательно, школьник лишен в этом случае необходимости набрать имя в поле ввода имен при первом знакомстве с операцией. Если технологи-

чески это по умолчанию формируемое имя файла удобно, то в методическом плане такое формирование создает определенные сложности. Во-первых, при такой ситуации требуются дополнительные (не принципиальные и, следовательно, методически избыточные) объяснения, почему имя файла уже написано, а во-вторых, это обстоятельство лишает ученика непосредственно оценить трудоемкость набора имени файла (записываемого латинскими буквами, вообще говоря, длинного слова), тогда как на этот непосредственный опыт ученика было бы удобно опираться несколькими минутами позднее, когда можно было бы говорить о технологических на сей раз преимуществах операции “Каталог” перед операцией “Чтение” при вызове файла с диска в рабочее поле редактора.

Впрочем, если демонстрационный компьютер недостаточно быстр, то методические преимущества могут поблекнуть перед значительными потерями времени, которое весь класс потеряет, ожидая перезагрузки. В таком случае приходится попросить школьника гипотетически представить, насколько сложно в операции “Чтение” отвечать на запрос имени файла.

Перед тем как продемонстрировать еще одну позицию основного меню — “Каталог”, потребуется введение этого нового термина. Бесспорно, определение каталога как списка имен файлов нельзя считать предельно строгим, оно потребует своего расширения даже при переходе к другому развитому редактору в нашем курсе. Однако сейчас определение каталога — понятия, вообще говоря, достаточно сложного для первого восприятия юным пользователем — можно считать вполне достаточным для решения всех методических задач темы.

Главное достоинство позиции “Каталог” состоит в том, что необходимость набора имени читаемого файла отпадает и заменяется гораздо более простой операцией выбора имени из списка. Такая замена (выбор имени вместо клавиатурного набора) — это прежде всего гарантия отсутствия ошибок при вызове файла. Это важно вообще, а для новичков — особенно.

Между тем конечный итог обеих операций — “Чтение” и “Каталог” — одинаков. Отсюда рекомендация: отдавать предпочтение операции “Каталог” во всех тех случаях, когда требуется обращение к файлу.

4. Правила формирования имен

Этот фрагмент занятия обязателен. Более того, он может занять достаточно много времени: формулировка правил (даже без записи их в тетрадь), демонстрация правильно и неверно записанных имен, разбор типовых ошибок именования, решение детьми контрольных упражнений, ответы на вопросы. Однако этот материал несложен и достаточно подробно описан в учебнике.

Новым понятием этого занятия можно считать структуру имени, его разделение на две составляющие — обязательную основу и факультативное (во всяком случае, в “Микроне”) расширение. Однако для методических нужд сегодняшнего и следующего занятий расширения имен файлов не принципиальны, и для простоты в ближайших упражнениях можно без них обойтись.

Продолжение следует

ИНФОРМАТИКА

Газета “Информатика” и Роботландский сетевой университет
продолжают первый этап совместной акции

“Подписчикам везде у нас дорога... и скидка!”

Данный купон дает право на скидку в размере 10% при приобретении:

- интерактивного учебника-лаборатории “HTML-конструирование” (см. № 21, 22/2000);
- программного пакета “Зимние вечера” (см. № 1, 5, 6, 7, 8, 9, 10/2001).

В дальнейшем наши подписчики получают скидки и на другие продукты Роботландского сетевого университета.

Для получения скидки необходимо выслать заявку на приобретение того или иного продукта по адресу: 152025, г. Переславль-Залесский, ул. Октябрьская, д. 43, кв. 112, Дуванову Александру Александровичу.

В письмо необходимо вложить оригинал данного купона или ксерокопию купона вместе с ксерокопией подписной квитанции на “Информатику”. Для быстрого получения программ рекомендуется дополнительно отправить электронное письмо с заявкой по адресу: kurs@robotland.botik.ru. В электронном письме требуется указать дату отправки бумажного письма с купоном или ксерокопиями.

Данное предложение действительно до 30 апреля 2001 года.



Внеклассная работа по информатике

Д.М. Златопольский,

Москва

Величины и единицы их измерения

Вам будут предложены используемые в информатике величины, их количественная характеристика и единицы их измерения. Необходимо указать номера величин, для которых неправильно указано значение или (и) единица измерения.

1. Разрешение монитора — 80 × 25 строк.
2. Скорость передачи информации — 33,6 килобита/сек.
3. Размер файла — 25 см.
4. Количество информации — 2,3 килобайта.
5. Быстродействие процессора — 1,1 гигагерца.
6. Размер папки (каталога) — 245 килобайт.
7. Разрешение сканера — 600 точек на дюйм.
8. Размер монитора — 80 символов.
9. Размер символов в текстовом редакторе Microsoft Word — 12 пикселей.
10. Скорость доступа к диску — 2000 слов в секунду.
11. Вместимость дискеты — $3,14 \cdot (3,5 \cdot 2,54)^2 = 248 \text{ см}^2$.
12. Размер файла — 2435 килобайт.
13. Разрешение монитора — 800 × 600 пикселей (точек).
14. Разрешение сканера — 10 символов на дюйм.
15. Размер папки (каталога) — 24 файла.
16. Скорость передачи информации — 2,4 км/сек.
17. Количество информации — 23 000 пикселей.
18. Размер символов в текстовом редакторе Microsoft Word — 20 пунктов.
19. Размер монитора — 800 × 600 пикселей (точек).
20. Размер файла, созданного графическим редактором, — 31 789 пикселей.
21. Количество информации — 2,3 бита.
22. Быстродействие процессора — 1000 бит в секунду.

23. Разрешение монитора — 17 дюймов.
24. Размер файла — 2 см².
25. Скорость передачи информации — 2,4 пикселя/сек.
26. Размер папки (каталога) — 2400 см³.
27. Количество информации — 2,3 килобита.
28. Диаметр дискет — 8 см.
29. Разрешение монитора — 35 см по диагонали.
30. Количество информации — 2333 квантибайта.
31. Разрешение сканера — 5 символов на сантиметр.
32. Скорость доступа к диску — 20 000 бит в секунду.
33. Размер файла, созданного графическим редактором, — 71 789 байт.
34. Разрешение монитора — 1024 × 768 мм.
35. Количество информации — 2333 бита.
36. Размер папки (каталога) — 2400 квантибайт.
37. Размер символов в текстовом редакторе Microsoft Word — 20 мм.
38. Диаметр дискет — 3,5 дюйма.
39. Разрешение монитора — 1024,5 × 768,5 пикселей.
40. Размер файла — 2,4 мегабайта.
41. Скорость доступа к диску — 20 000 бит в секунду.
42. Разрешение монитора — 600 точек на дюйм.
43. Вместимость дискеты — 1,44 мегабайта.
44. Размер текстового файла — 71 789 символов.
45. Количество информации — 2,3 гигабайта.
46. Размер текстового файла — 71 789 слов.
47. Размер монитора — 800 × 600 пикселей (точек) на дюйм.

Ответы (номера величин, для которых неправильно указано значение или (и) единица измерения): 1, 3, 8, 9, 10, 11, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 28, 30, 31, 32, 34, 36, 37, 39, 41, 42, 44, 46, 47.

“Программы-профессии” и “программы-должности”

Задание может предъявляться учащимся в одном из шести уровней.

Уровень 1

Назовите типы программ и встроенные возможности офисных программ, названия которых связаны с профессиями и должностями.

Возможные ответы: редактор, Проводник Windows, Мастер, Помощник, Конструктор.

Примечание. Можно также предложить учащимся рассказать о назначении указанных программ и возможностей.

Уровень 2

Вам будут предложены определения устройств компьютера, типов программ и встроенных возможностей офисных программ, названия которых аналогичны названиям профессий, должностей и т.п. Назовите эти устройства, программы и возможности.

1. Приложение Windows, предназначенное для работы с папками и файлами.
2. Средство офисных программ пакета Microsoft Office, с помощью которого происходит, например, построение диаграмм и графиков, создание форм и запросов в базах данных и т.п.

3. Средство офисных программ пакета Microsoft Office, снабжающее пользователя подсказками, советами, справочной информацией.

4. Средство системы управления базами данных Microsoft Access, с помощью которого создаются и меняются макет и структура объекта (таблицы, запроса, формы, отчета).

5. Тип программ, предназначенных для подготовки текстовых или графических документов.

6. Устройство, управляющее работой системной шины персонального компьютера.

Ответы: 1. Проводник. 2. Мастер. 3. Помощник. 4. Конструктор. 5. Редактор. 6. Арбитр.

Уровень 3

Вам будут перечислены комментарии к названиям профессий, должностей и т.п. Термины, аналогичные этим названиям, используются также в названиях устройств компьютера, типов программ и встроенных возможностей офисных программ.

Назовите эти устройства, программы и возможности.

1. Профессия работника железнодорожного транспорта.
2. Профессия работника, готовящего книги, газеты и т.п. к изданию.
3. Работник, занимающийся разработкой новых устройств, изделий и т.п.
4. Специалист, достигший высокого искусства в своем деле.
5. Спортивный судья.
6. Должность человека, который помогает кому-нибудь в чем-либо.

Ответы: 1. Проводник (Windows). 2. Редактор (текстовый или графический). 3. Конструктор (таблиц, форм, запросов, отчетов). 4. Мастер (диаграмм, форм и т.п.). 5. Арбитр (шины). 6. Помощник.

Примечание. Можно также предложить учащимся рассказать о назначении указанных устройств, программ и возможностей.

Уровень 4

Перед вами две колонки. В первой приведены названия устройств компьютера, типов программ и встроенных возможностей офисных программ, во второй — определения терминов, перечисленных в первой колонке. Необходимо найти пары номеров (по одному из каждой колонки), устанавливающие соответствие между термином и его определением.

- | | |
|----------------|---|
| 1. Арбитр | 1. Средство системы управления базами данных Microsoft Access, с помощью которого создаются и меняются макет и структура объекта (таблицы, запроса, формы, отчета) |
| 2. Компилятор | 2. Средство офисных программ пакета Microsoft Office, с помощью которого происходит, например, построение диаграмм и графиков, создание форм и запросов в базах данных и т.п. |
| 3. Конструктор | 3. Тип программ, предназначенных для подготовки текстовых или графических документов |
| 4. Мастер | 4. Приложение Windows, предназначенное для работы с папками и файлами |
| 5. Помощник | 5. Разновидность системной программы, переводящей алгоритм, записанный на языке высокого уровня, в машинные коды |
| 6. Проводник | 6. Устройство, управляющее работой системной шины персонального компьютера |
| 7. Редактор | 7. Тот, кто помогает кому-нибудь в чем-либо |

Ответы: 1 — 6 (арбитр), 2 — 5 (компилятор), 3 — 1 (конструктор), 4 — 2 (Мастер), 5 — 7 (Помощник), 6 — 4 (Проводник), 7 — 3 (редактор). Первыми указаны номера терминов в левой колонке.

Уровень 5

Перед вами — две колонки. В первой приведены определения профессий и должностей. Во второй колонке приведены названия типов программ и встроенных возможностей офисных программ. Необходимо найти пары номеров (по одному из каждой колонки), относящиеся к терминам-омонимам.

- | | |
|---|---|
| 1. Спортивный судья | 1. Средство системы управления базами данных Microsoft Access, с помощью которого создаются и меняются макет и структура объекта (таблицы, запроса, формы, отчета) |
| 2. Человек, обобщающий результаты чужих исследований | 2. Средство офисных программ пакета Microsoft Office, с помощью которого происходит, например, построение диаграмм и графиков, создание форм и запросов в базах данных и т.п. |
| 3. Работник, занимающийся разработкой новых устройств, изделий и т.п. | |
| 4. Специалист, достигший высокого искусства в своем деле | |

5. Должность человека, который помогает кому-нибудь в чем-либо
6. Профессия работника железнодорожного транспорта
7. Профессия работника, готовящего книги, газеты и т.п. к изданию

3. Тип программ, предназначенных для подготовки текстовых или графических документов
4. Приложение Windows, предназначенное для работы с папками и файлами
5. Разновидность системной программы, переводящей алгоритм, записанный на языке высокого уровня, в машинные коды
6. Устройство, управляющее работой системной шины персонального компьютера
7. Средство офисных программ пакета Microsoft Office, снабжающее пользователя подсказками, советами, справочной информацией

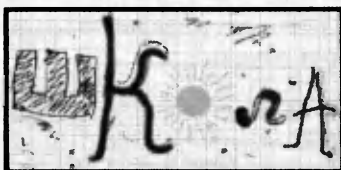
Ответы: 1 — 6 (арбитр), 2 — 5 (компилятор), 3 — 1 (конструктор), 4 — 2 (Мастер), 5 — 7 (Помощник), 6 — 4 (Проводник), 7 — 3 (редактор). Первыми указаны номера определений в левой колонке.

Уровень 6

Ниже приведены определения терминов-омонимов. Назовите эти термины.

1. Устройство, управляющее работой системной шины персонального компьютера — Спортивный судья.
2. Разновидность системной программы, переводящей алгоритм, записанный на языке высокого уровня, в машинные коды — Человек, обобщающий результаты чужих исследований.
3. Средство системы управления базами данных Microsoft Access, с помощью которого создаются и меняются макет и структура объекта (таблицы, запроса, формы, отчета) — Работник, занимающийся разработкой новых устройств, изделий и т.п.
4. Средство офисных программ пакета Microsoft Office, с помощью которого происходят, например, построение диаграмм и графиков, создание форм и запросов в базах данных и т.п. — Специалист, достигший высокого искусства в своем деле.
5. Средство офисных программ пакета Microsoft Office, снабжающее пользователя подсказками, советами, справочной информацией — Должность человека, который помогает кому-нибудь в чем-либо.
6. Приложение Windows, предназначенное для работы с папками и файлами — Профессия работника железнодорожного транспорта.
7. Тип программ, предназначенных для подготовки текстовых или графических документов — Профессия работника, готовящего книги, газеты и т.п. к изданию.

Ответы: 1. Арбитр. 2. Компилятор. 3. Конструктор. 4. Мастер. 5. Помощник. 6. Проводник. 7. Редактор.

**2001**Москва,
КВЦ "Сокольники"**12–15 апреля 2001 года****V Московская
международная выставка****Всё об образовании, всё для образования**

Приглашаем на вторую ежегодную конференцию нашей газеты!

Дорогие коллеги! Как и в прошлом году, в рамках выставки "Школа-2001" пройдет конференция, организуемая газетой "Информатика". В этом году на обсуждение будет вынесен вопрос

"Итоговая аттестация по информатике"

В конференции примут участие представители Министерства образования, авторы учебников по информатике, региональные представители и постоянные авторы нашей газеты.

Вход на конференцию свободный.

Конференция пройдет 14 апреля, в субботу, в два этапа. С 11⁰⁰ до 13⁰⁰ состоится пленарное заседание в выставочном комплексе "Сокольники", а в 15⁰⁰ начнется "круглый стол" в редакции нашей газеты.

По всем вопросам, связанным с конференцией, пожалуйста, обращайтесь к ответственному секретарю редакции "Информатики" Наталье Павловне Медведевой по телефону (095) 249-48-96 (с 16⁰⁰ до 18⁰⁰ по рабочим дням, время московское) или по электронной почте inf@1september.ru.

“Назаметки” Сидорова

А.А. Дуванов,
г. Переславль-Залесский

Продолжение. См. № 6, 7, 8, 10/2001

“Назаметка” 9. Визуальные редакторы

При создании страниц я рекомендую вам удержаться от использования таких программ, как FrontPage, HomePage, PageMill, SiteMill и т.п. Если вы одной рукой мешаете в кастрюле борщ, другой утомительно замачиваете белье, а третьей собираетесь делать сайт, то это ваши программы (они для домохозяек).

Артемиий Лебедев

Легкость, с которой Word сохраняет файл в формате HTML, заставляет начинающего web-мастера задаться неизбежным вопросом: “зачем изучать язык HTML?”.

Впрочем, Word — явно неудачный пример. Как раз в этом случае головная боль обеспечена на все 100%.

При использовании Word в качестве визуального HTML-редактора возникают следующие проблемы:

- Практически невозможно добиться точного соответствия визуального представления документа Word и HTML-документа,

- Word вставляет в HTML-коды много лишнего. Процент “мусора” порой достигает 50%, а пересылка “довеска” по сети стоит времени и денег,

- Word не справляется с конвертированием в HTML больших и сложных по структуре файлов,

- Word не умеет записывать HTML-тексты “шрифтом по умолчанию”, он обязательно вставляет тег FONT с атрибутом face, в котором прописывает конкретный шрифт. Если на компьютере клиента указанные шрифты не содержат русских букв, то прочитать текст нельзя. Вероятно, вы встречались в Интернете с такой ситуацией, пробовали менять кодировки, но все с нулевым эффектом — браузер показывает текст шрифтом, в котором нет русских букв, и смена кодировки в этом случае, естественно, не помогает.

Существуют более “продвинутые” по сравнению с Word визуальные HTML-редакторы. Среди них — очень хороший и мощный продукт HomeSite, а также очень простой в обращении FrontPage Express.

Последний редактор, как и Word, переформатирует текст HTML-программы по-своему, причем не лучшим образом. Переформатирует заново *весь* код, а не только ту вставку, которая сделана в этом редакторе.

Можно добавить еще, что FrontPage не все умеет делать даже на уровне стандартного HTML 3.2 и, бывает, просто чудит. Например, отказывается вставлять пустую строчку между двумя таблицами, примыкающими вплотную друг к другу.

“Назаметки” Сидорова — это одно из учебных пособий Ротландского сетевого университета. Сообщение о предварительном приеме заявок на 2001/2002 учебный год будет опубликовано в одном из апрельских номеров газеты.

Но тем не менее редактор FrontPage Express — хороший. Работать с ним удобно.

Последнее достижение фирмы Microsoft — FrontPage 2000. Этот продукт — существенный прорыв на рынке визуальных HTML-редакторов.

FrontPage 2000 не выкидывает те теги, которые не может понять и осмыслить, и не вписывает множество тегов, которые никому не нужны. А главное, FrontPage 2000 не трогает то, что трогать не надо. Код, который создается в визуальном режиме, он форматирует по-своему, а код, написанный вручную, переписывает без изменений.

В комплект FrontPage 2000 входит огромное количество скриптов, формы, гостевые книги, картинки, таблицы и всякие другие “полуфабрикаты” для изготовления HTML-блюда на разный вкус.

Так нужно ли изучать HTML и заниматься ручной “вспашкой”, когда есть такие мощные “землеобрабатывающие” комбайны?

Мнение Сидорова и авторитетных web-дизайнеров в этом вопросе однозначно — да!

Если вам нужно создать оригинальное HTML-произведение, которое не будет выглядеть как набившие оскомину интернетовские штампы, если вы хотите получить оптимальный код и добавить на страницу интерактивность за счет собственных скриптов и таблиц стилей, знания тегов и принципов построения HTML-кода просто необходимы.

Вы не будете попадать в беспомощную ситуацию, когда визуальный редактор делает что-то не то, а подправить код вручную вы не в состоянии.

Наконец, свободно владея HTML, вы будете ощущать себя принятым в хорошую компанию.

HTML — это латынь гипертекстовой разработки, и знать ее необходимо, хотя бы для того, чтобы понимать, о чем говорят и пишут люди в этой области.

Ну а визуальные редакторы, конечно, вещь полезная. Почему бы и не использовать их, когда действительно трудно. Например, никто не заставит меня вручную размечать графическую карту гипертекстовыми ссылками.

Конкурсы Ротландского университета — это просто бездонный колодец поучительнейших иллюстраций ко всем мыслимым ошибкам и погрешностям начинающих web-проектировщиков.

Вот стакан визуального кодирования. Примите как лекарство: вы всерьез начнете учить “матчасть” (теги и техники их использования).

1. Пробел, который центрируется в пустой строке.

```
<p align="center">&nbsp;</p>
```

Визуальному редактору все равно, что центрировать, можно и пробел.

2. Ужасный абсолютный адрес

```
<p align="center">
```

```
</p>
```

Совет для любителей визуального редактирования: обязательно перед публикацией в Интернете перенесите свой сайт на другое место жесткого диска (еще лучше — на другой компьютер). Вы с удивлением обнаружите, что картинки не показываются, а ссылки не работают. Причина в абсолютных адресах.

И еще один совет: введите на своем сайте строгую дисциплину имен файлов и каталогов, ведь на сервере, который обычно работает под UNIX, регистр символов имеет значение! Адреса, прекрасно уживающиеся с Windows, на сервере могут перестать работать. Если картинка, например, имеет имя background.jpg, а в ссылке на нее написано Background.jpg, то изображения в Интернете не увидят.

Самое лучшее, когда имена всех файлов и каталогов внутри сайта имеют формат 8.3: не более 8 символов в имени, не более 3 в расширении. Никаких русских букв, пробелов и спецсимволов. Все имена каталогов и файлов должны быть записаны в одном регистре (например, нижнем).

3. Полуужирный пробел с собственным лицом.

Визуальный редактор сотворил такой код:

```
<font color=
  "#FF0080" size="6" face="Courier New">
  <strong> </strong></font>
```

Для пробела (пустого места) определяются новый шрифт, цвет, и он записывается полуужирным начертанием!

4. "А" — это совсем не то, что "ккордики"!

```
<font color="#0000FF" size="5"
  face="Times New Roman">А</font><font
  color="#0000FF" size="5"
  face="Times New Roman">ккордики</
  font><font
  color="#0000FF" size="5"
  face="Times New Roman">

</font>
```

Шрифт Times New Roman браузер использует по умолчанию. Задавать его явно не нужно и даже вредно, как вообще задавать шрифт атрибутом face. Цвет шрифта надо задавать один раз в теге BODY, а не в каждом слове абзаца (тем более один и тот же). Зачем выделять букву "А" в отдельный font-блок, когда для нее задается то же самое, что и для продолжения "ккордики"? Зачем задавать размер, вид и цвет шрифта для картинки?

Приведенный выше бред переписывается так:

```
<FONT size=5>Аккордики</FONT>
<IMG src=Star.jpg width=14 height=11>
```

5. "Хорошего" должно быть много!

Вот плотно набитый код табличной строки:

```
<td width="34%"><font color="#0000FF" size="4"
  face="Courier New"><strong>Местонахождение</
  strong></font></td>
<td width="18%"><font color="#0000FF" size="4"
  face="Courier New"><strong>Количество MB</
  strong></font></td>
<td width="17%"><font color="#0000FF" size="4"
  face="Courier New"><strong>E-Mail</strong>
</font></td>
<td width="7%"><font color="#0000FF" size="4"
  face="Courier New"><strong>CGI </strong></font></td>
```

```
<td width="23%"><font color="#0000FF" size="4"
  face="Courier New"><strong>Примечание</strong>
</font></td>
```

"Руками" это записывается так:

```
<FONT color=#0000FF size=4 face="Courier New">
<STRONG>
...
<TD width=34%>Местонахождение</TD>
<TD width=18%>Количество MB</TD>
<TD width=17%>E-Mail</TD>
<TD width=7%>CGI</TD>
<TD width=23%>Примечание</TD>
...
</STRONG>
</FONT>
```

6. Бррр!

```
<br><br><br><br><br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br><br><br><br><br>
```

Если вам нужен большой отступ, используйте картинку-распорку.

7. Двойное центрирование: а вдруг однократное не поможет?

```
<div align="center"><center>
...
</center></div>
```

8. Визуальные узоры длиной в километр.

```
<table>
<ul>
<tr><td rowspan=9 width=20></td>
<td><li><font size=+1 face="courier new">
<i><b><small><a href="biog.htm">биография
</a></small></b></i></font></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="friend.htm">
друзья</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="hobby.htm">
увлечения</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="favorite.htm">
любимое</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="educat.htm">
образование</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="future.htm">
future life</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="dream.htm">
мысли по поводу...</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="map.htm">
карта сайта</a></small></b></i></td></tr>
<tr><td><font size=+1 face="courier new">
<i><b><li><small><a href="ps.htm">P.<small>oct
</small> S.<small>сриктуп</small></a> </b>
</i></td></tr>
</ul>
</table>
```

Как видите, проводить "уборку" после визуального редактора очень утомительно и требует больших временных затрат. Гораздо быстрее, качественнее и безопаснее писать сразу в обычном текстовом редакторе.

Продолжение следует

**ЧТОБЫ ПРИОБРЕСТИ ЗАИНТЕРЕСОВАВШИЕ ВАС
КНИГИ НАЛОЖЕННЫМ ПЛАТЕЖОМ:**

1. Укажите в купоне вашу фамилию, имя, отчество, индекс и почтовый адрес.
2. Выберите нужные вам книги из списка (не более 5 названий в один купон) и укажите соответствующий номер лота и количество экземпляров.
3. Вырежьте и вышлите купон в конверте с пометкой «Книга — почтой» по адресу: 121165, Москва, ул. Киевская, д. 24, «Первое сентября».

Телефоны для справок: (095) 249-28-77, (0852) 45-07-77.

КУПОН Книга – почтой
ЗАПОЛНЯЕТСЯ ПЕЧАТНЫМИ БУКВАМИ!

[illegible]

00-3-00

ОБРАЗЕЦ:

[illegible]

Указанная в купоне цена включает в себя стоимость доставки наземным транспортом от издательства до вашего почтового отделения (при авиаперевозках сумма наложенного платежа увеличивается на авиатариф). Рассылка производится только на территории РФ. Если полученная бандероль содержит неполный заказ, это означает, что заказ разбит на две или более бандероли.

Номер лота	Название	Цена
10-01-0	С.Л. Соловейчик. Пушкинские проповеди	55 руб.
10-02-0	С.Л. Соловейчик. Последняя книга	65 руб.
10-05-0	С.Л. Соловейчик. Педагогика для всех	75 руб.
▼ 10-03-0	Школа сотрудничества. (Сборник статей по педагогике сотрудничества)	38 руб.
27-01-9	Я иду на урок физики. 7 класс (в 3 книгах)	114 руб.
29-01-0	Я иду на урок химии. 8–11 классы	38 руб.
29-02-0	Я иду на урок химии. Летопись важнейших открытий. XVII–XIX вв.	38 руб.
21-01-0	Я иду на урок математики. 5 класс	38 руб.
21-01-7	Я иду на урок математики. Тесты. 5 класс	15 руб.
12-01-0	Я иду на урок биологии. Зоология. Беспозвоночные	38 руб.
12-02-0	Я иду на урок биологии. Зоология. Рыбы и земноводные	38 руб.
12-03-0	Я иду на урок биологии. Зоология. Пресмыкающиеся	38 руб.
12-04-0	Я иду на урок биологии. Человек и его здоровье	38 руб.
19-02-0	Я иду на урок истории. Древнейшая и древняя история	38 руб.
19-04-0	Я иду на урок истории. Материалы по истории средних веков	38 руб.
▼ 19-03-0	Л.А. Кацва. Россия в 1917–1918 годах. 10–11 классы	38 руб.
14-01-0	Я иду на урок географии. Физическая география материков и океанов	38 руб.
14-02-0	Я иду на урок географии. История географических открытий	38 руб.
14-03-0	Я иду на урок географии. Физическая география России	38 руб.
20-02-0	Я иду на урок литературы. 5 класс	38 руб.
20-03-0	Я иду на урок литературы. 9 класс	38 руб.
20-01-0	Я иду на урок литературы. 10 класс	38 руб.

▼ Тираж ограничен.

Указанные цены действительны до 1 апреля 2001 года

“В р о ж д е н н ы й м е х а н и к”

В 1769 году императрице Екатерине II преподнесли необычные часы, по форме и размерам напоминающие утиное яйцо. Они не только показывали и отбивали время, но и давали представление. Каждый час в корпусе раскрывались миниатюрные дверцы и изящные фигурки под музыку разыгрывали сценку. Кроме того, в полдень часы исполняли гимн [1, 2]. Эти часы изготовил Кулибин.

Иван Петрович Кулибин (1735—1818) был не простым механиком. Более 30 лет (начиная с 1770 года) он возглавлял все мастерские Петербургской академии наук, то есть являлся первым механиком России [1]. Кулибин изготавливал и чинил всевозможные механические, оптические, электрические, акустические и иные приборы, оснащал ими академические лаборатории и экспедиции, помогал ученым, создавая по их заказу нестандартное оборудование. Он был универсальным мастером.

Все приборы, созданные Кулибиным, отличались высоким качеством. В одном из протоколов собрания Академии наук записано: “Осмотрены вновь сделанные механиком Кулибиным электрические машины и найдено, что сделаны они очень хорошо и очень сильны”.

Однако работа в мастерских при всем своем многообразии и объеме не могла исчерпать всех его творческих сил.

В 1770-х годах Кулибин занялся проектированием одноарочного моста через Неву. Он предложил принципиально новую конструкцию деревянного моста, подробно описал работы, необходимые для постройки этого сложнейшего сооружения, подготовил методику испытания отдельных его частей, изобрел соответствующие приборы. Причем Кулибин не ограничился экспериментами, а разработал и изложил теорию своей конструкции. Наконец, он первый поднял вопрос о применении металла в качестве материала для мостов (в то время мосты строили из дерева и камня) [2].

Проект принес славу Кулибину и в России, и за границей; он был награжден специально отчеканенной медалью. Почетный член Петербургской академии наук Даниил Бернул-

ли писал ученику Леонарда Эйлера: “...то, что Вы сообщаете о врожденном механике г. Кулибине по поводу деревянного моста через Большую Неву, имеющую ширину 1057 английских футов (около 300 м. — **Прим. ред.**), внушает мне высокое мнение об этом талантливом строителе и искусном плотнике, воспитанном между простыми крестьянами и обязанном своими высшими познаниями только своего рода наитию... Главный строитель чаще всего должен полагаться на свое собственное чутье. Здесь я и ощущаю всю выгоду иметь такого человека, как г. Кулибин, к которому я проникнут уважением, но не могу победить своего недоверия, когда речь идет о таком огромном мосте”.

Бернулли был настолько высокого мнения о способностях Кулибина, что, когда Эйлер опубликовал свою работу, посвященную исследованию упругих свойств балки, написал секретарю академии: “...не могли бы Вы поручить г. Кулибину подтвердить теорию г. Эйлера опытами, без чего его теория останется верной лишь гипотетически?”

В 1791 году Кулибин создает “самобеглую” коляску, которая приводилась в движение силой двух человек. Экипаж имел рулевое управление, “коробку передач” и тормозное устройство, состоящее из двух пружин (при торможении они закручивались). Чтобы повозка с двумя пассажирами и двумя обеспечивающими ее движение людьми ехала со скоростью 10 км/час по булыжной мостовой, требовалась мощность около 0,5 л. с. (лошадиных сил). Разумеется, долго “держат” такую мощность было трудно. Нужен был двигатель, и Кулибин работал над его созданием, но безуспешно, поскольку он пытался изобрести перпетуум-мобиле.

* Лошадиная сила — внесистемная единица мощности; 1 л. с. $\equiv$ 736 Вт.

Может показаться странным, что такой практичный человек, как Кулибин, изобретал вечный двигатель, но тем не менее это так. К концу жизни “врожденный механик” понял перспективность своего занятия.

Кулибин изобрел также кресло для подъема Екатерины II на верхние этажи Зимнего дворца; создал оптический (или семафорный) телеграф, подобный телеграфу французского изобретателя Клода Шаппа [3], но со своей системой приведения в действие и своим кодом, более простым, чем код Шаппа; предложил целый ряд устройств для механизации производства зеркал больших размеров; сконструировал станки для шлифовки оптических стекол, а



Стела на могиле И.П. Кулибина в Нижнем Новгороде

также станки для обработки деталей металлических многопролетных мостов собственной конструкции. В списке устройств, которые он создал: прожекторы; “водоходные” суда, способные идти против течения; механическая сеялка; плавучая мельница; насосы и многое другое.

Творчество Кулибина оказало огромное влияние на развитие русской технической мысли. И народ не забыл великого механика. Его имя стало нарицательным для “обозначения” талантливых изобретателей, а иногда и просто талантливых людей [1].

Литература

1. *Лишевский В.П.* Рассказы об ученых. М.: Наука, 1986.
2. *Детская энциклопедия.* М.: Издательство Академии педагогических наук РСФСР, 1960. Т. 5.
3. *Шаппе Д.Л.* По всему земному шару: Прошлое, настоящее и будущее кабелей связи. М.: Радио и связь, 1985.

Двадцатилетие

Окончание. Начало на с. 1

эта простая в использовании система продается по цене всего 1565 долларов, причем "предлагается множество усовершенствованных возможностей, а с дополнительным программным обеспечением могут использоваться сотни популярных прикладных программ" [1].

Тогда впервые на полуофициальном уровне было применено словосочетание Personal Computer (PC), ставшее вскоре столь популярным во всем мире. Данное словосочетание было не только удачно подобранным названием для новой машины, но еще и ключевым термином для совершенно новой концепции.

Следует отметить, что на этот раз фирма IBM не стала разрабатывать все компоненты своего компьютера собственными силами. В частности, в качестве основы для IBM PC был взят уже существовавший 16-разрядный микропроцессор Intel 8088 [2, 3, 4]. Такая политика позволила не расходовать слишком много денежных средств и сократить время на создание машины.

К весне следующего года, к удивлению многих, спрос на IBM PC намного превысил предложение. Стало ясно, что эту машину ждет большой успех. И действительно, в индустрии микрокомпьютеров IBM PC был быстро признан безусловным лидером.

Успеху во многом способствовала так называемая открытая архитектура IBM PC [1, 3], позволяющая другим фирмам проявлять соответствующим образом свои творческие и предпринимательские способности. Появились сотни новых компаний, которые вы-

пускали машины, совместимые с этим компьютером, или предлагали для него дополнительные платы, адаптеры и периферийные устройства.

Разумеется, есть и другие причины необычайной популярности IBM PC. К ним относятся [3, 5]:

- сравнительно невысокая стоимость;
- относительная простота использования;
- возможность "индивидуального взаимодействия" с компьютером без посредников и ограничений;
- достаточно высокие возможности, касающиеся обработки информации;
- наличие программного обеспечения, охватывающего многие сферы человеческой деятельности, и мощных систем для разработки нового программного обеспечения;
- высокая надежность, простота ремонта;
- возможности расширения и адаптации (один и тот же компьютер может быть оснащен различными периферийными устройствами и разными программами).

Кроме того, почти сразу стало ясно, что для решения целого ряда задач использовать IBM PC просто выгоднее, чем большие машины и мини-ЭВМ.

Сейчас машины семейства IBM PC (главным образом 32-разрядные) составляют подавляющее большинство всех производимых в мире персональных ЭВМ. При этом компьютеры — во многом благодаря данному семейству — уже стали почти такими же обычными предметами, как телефон или телевизор, и теперь можно с полной уверенностью утверждать, что с появлением IBM PC информатика перешла на новую ступень развития [1, 5, 6, 7].



IBM PC "Junior".
Классический персональный компьютер фирмы IBM образца 1983 г.

Литература

1. *Петроченков А.В.* Десятилетний юбилей IBM PC // Новое в жизни, науке и технике. Сер. "Вычислительная техника и ее применение". № 12/91.
2. *Нортон П.* Программно-аппаратная организация IBM PC: Пер. с англ. М.: Радио и связь, 1992.
3. *Фигурнов В.Э.* IBM PC для пользователя. Изд. 6-е, перераб. и доп. М.: ИНФРА-М, 1995.
4. IBM и IBM PC // Информатика, № 38/99.
5. *Кренкель Т.Э., Коган А.Г., Тараторин А.М.* Персональные ЭВМ в инженерной практике. М.: Радио и связь, 1989.
6. *Частиков А.П.* От калькулятора до суперЭВМ // Новое в жизни, науке, технике. Сер. "Вычислительная техника и ее применение". № 1/88.
7. Знакомьтесь: компьютер: Пер. с англ. М.: Мир, 1989.

Гл. редактор С.Л. Островский Зам. гл. редактора И.Н. Фалина Редакция: Е.В. Андреева Н.Л. Бельская Л.Н. Картелишвили Н.П. Медведева Дизайн и верстка: Н.И. Пронская Корректоры: Е.Л. Володина, С.М. Подберезина	©ИНФОРМАТИКА 2001 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 Комитете РФ по печати 20.03.1997. Регистрационный номер 015873. Отпечатано в типографии ОАО ПО "Пресса-1". 125865, ГСП, Москва, ул. "Правды", 24. Тираж 6600 экз. Срок подписания в печать по графику 07.03.2001. Номер подписан 07.03.2001. Заказ № 11573 Цена свободная
ИНДЕКС ПОДПИСКИ для индивидуальных подписчиков 32291 комплекта приложений 32744			
Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84			
Internet: inf@1september.ru WWW: http://www.1september.ru			

Номера "Информатики" (в ограниченном количестве) бесплатно распространяются в офисе ведущей российской антивирусной фирмы "ДиалогНаука".

Адрес: Москва, ул. Вавилова, д. 40, офис 103. Тел. 137-01-50.

Английский язык (Е.В. Громушкина), индекс подписки — 32025; **Библиотека в школе** (О.К. Громова), индекс подписки — 33376; **Биология** (Н.Г. Иванова), индекс подписки — 32026; **Воскресная школа** (монах Киприан (Яценко), индекс подписки — 32742; **География** (О.Н. Коротова), индекс подписки — 32027; **Здоровье детей** (А.У. Лекманов), индекс подписки — 32033; **Информатика** (С.Л. Островский), индекс подписки — 32291; **Искусство** (Н.Х. Исмаилова), индекс подписки — 32584; **История** (А.Ю. Головатенко), индекс подписки — 32028; **Литература** (Г.Г. Красухин), индекс подписки — 32029; **Математика** (И.Л. Соловейчик), индекс подписки — 32030; **Начальная школа** (М.В. Соловейчик), индекс подписки — 32031; **Немецкий язык** (М.Д. Бузоева), индекс подписки — 32292; **Русский язык** (Л.А. Гончар), индекс подписки — 32383; **Спорт в школе** (Н.В. Школьникова), индекс подписки — 32384; **Управление школой** (А.И. Адамский), индекс подписки — 32652; **Физика** (Н.Д. Козлова), индекс подписки — 32032; **Французский язык** (Г.А. Чесновицкая), индекс подписки — 33371; **Химия** (О.Г. Блохина), индекс подписки — 32034; **Школьный психолог** (М.Н. Сартан), индекс подписки — 32898.