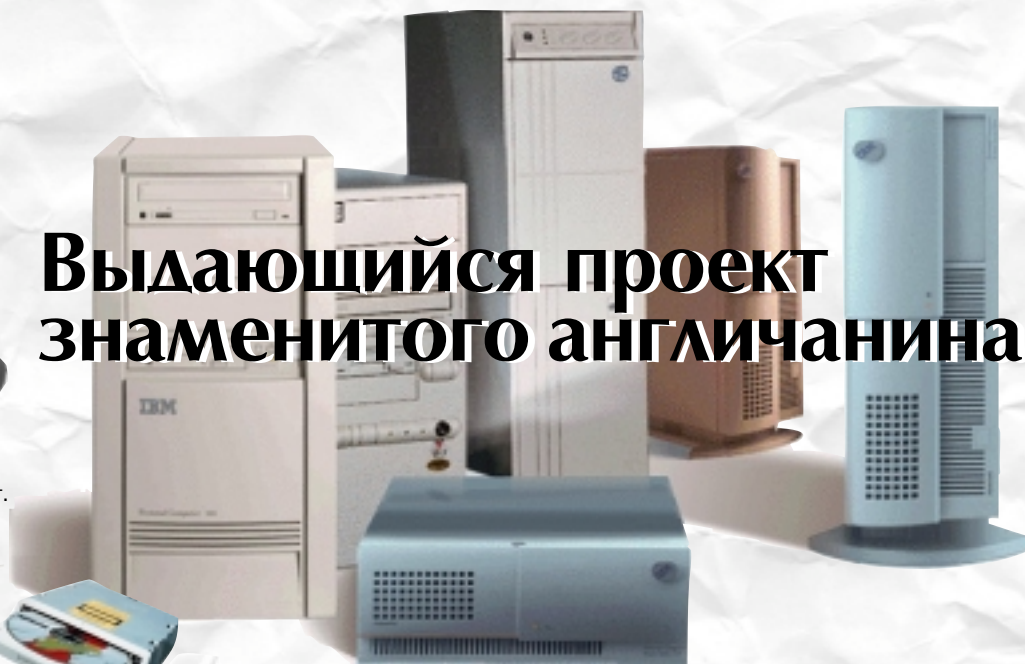


ИНФОРМАТИК



Чарльз Бэббидж. 1847 г.

Выдающийся проект знаменитого англичанина



165 лет назад Чарльз Бэббидж (1792—1871) определил состав и назначение функциональных средств компьютера в своем неосуществленном проекте Аналитической машины

О Чарльзе Бэббидже читайте на последней странице

Читайте в номере

Олимпиады 2–4

И.Л. Миронов, И.В. Романовский. Задача о круге

Рассматривается решение первой из шести задач XI Всероссийской олимпиады школьников по информатике (проходившей весной в Санкт-Петербурге), об итогах которой рассказывалось в предыдущем номере.

Информатика в лицах 5–6

А.Н. Колмогоров, В.А. Залгаллер. Леонид Витальевич Канторович. К 70-летию со дня рождения

Мы продолжаем публиковать материалы из сборника "Очерки истории информатики в России". В этом номере вашему вниманию предлагается статья о выдающемся ученом, лауреате Нобелевской премии Л.В. Канторовиче.

Тематический выпуск

Информатика в виртуальной школе

Весьма необычный, "направленный в будущее" пробный учебник для 7-го класса общеобразовательной школы, уже прошедший достаточно тщательную апробацию на самых обычных учениках самой обычной школы. По мнению автора, "это чуть ли не единственный учебник информатики, который можно применять так, как применяются учебники по любому другому предмету".

Задача о круге

И.Л. Миронов, И.В. Романовский

На последней Всероссийской олимпиаде по информатике была предложена следующая задача (см. № 33/99).

Плоскость разбили на одинаковые прямоугольники размера $M \times N$ со сторонами, параллельными осям координат, и вершинами, расположенными в точках $(M-i, N-j)$, где i и j принимают всевозможные целые значения. Пусть на этой плоскости задана точка $P(x, y)$ с целочисленными координатами. Назовем расстоянием от точки P до некоторого прямоугольника наименьшее из расстояний от P до точек этого прямоугольника, включая его границу. В частности, расстояние от точки до прямоугольника, в котором она содержится, равно 0.

Требуется написать программу, перечисляющую прямоугольники, удаленные от P на расстояние, не превосходящее L . Прямоугольники должны быть перечислены в порядке неубывания этого расстояния.

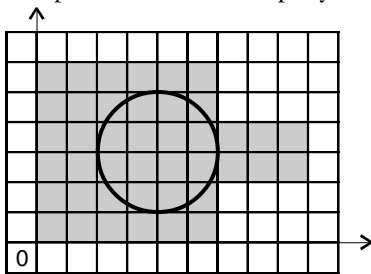
Формат входных данных

Во входном файле содержатся целые числа M, N, L, x и y ($0 < M \leq 10, 0 < N \leq 10, 0 \leq L \leq 300, -30\,000 < x, y \leq 30\,000$), разделенные пробелами и/или переводами строк.

Формат выходных данных

Введите в выходной файл координаты левых нижних углов искомых прямоугольников в описанном выше порядке. Прямоугольники, равноудаленные от P , могут выводиться в произвольном порядке.

Ниже приведен пример входного и выходного файлов для случая, представленного на рисунке.



Пример входного файла

```
3 2 2
4 3
```

Пример выходного файла

```
3 2
3 0
0 2
3 4
0 0
0 4
6 2
```

Решение

Не будем комментировать начало, обозначения для входных данных и их ввод. Обратите внимание на то, что при хорошем стиле программирования отдельно выделяются процедуры ввода данных, инициализации и собственно алгоритм решения задачи. Это облегчает отладку и понимание программы.

Обозначения соответствуют формулировке задачи, и единственное, о чем следует сказать, — это “сдвиг” начала системы координат как можно ближе к начальной точке.

```
xInit := x mod M;
if xInit < 0 then Inc(xInit, M);
yInit := y mod N;
if yInit < 0 then Inc(yInit, N);
xOffset := x - xInit;
yOffset := y - yInit;
```

Вычисленные смещения $xOffset$ и $yOffset$ используются дальше только в процедуре вывода очередной точки `outPoint`.

Всю вычислительную работу выполняет процедура `Solve`. Она разбивает плоскость на кольцевые области, последовательно загружает в буфер все точки очередного кольца, сортирует точки в буфере по расстоянию от прямоугольника до начальной точки и выводит отсортированные точки в файл результата. Для сортировки используется процедура

```
shellSort(nSize: integer),
```

ее легко написать, и она достаточно эффективна.

Загрузка в буфер точек очередного кольца с внутренним радиусом R_{min} и внешним радиусом R_{max} заключается в просмотре всех возможных значений координаты y , пока на очередной горизонтали еще могут встретиться искомые точки, и просмотре при каждом y возможных значений x влево и вправо от центра. Для упрощения просмотра мы фиксируем для каждого y первое “предлагаемое” значение x при движении влево и вправо.

Если вы пробовали написать свое решение этой задачи, то оцените краткость и скорость работы приведенной ниже функции определения расстояния от начальной точки до прямоугольника с заданным левым нижним углом (`dist2`). В решениях победителей Всероссийской олимпиады аналогичные функции занимают 20—30 строк.

```

const nBuf = 1000;
      MAXL = 300;
      MAXN = 10;
type
  TPoint = record { тип Точка }
    x, y: integer;
    d2: longint; { квадрат расстояния }
  end;
  TLines = array[-MAXL-2*MAXN..MAXL+2*MAXN] of integer;
var
  M, N, x, y: integer;
  XOffset, yOffset: integer;
  XInit, yInit: integer;
  buf: array[1..nBuf] of TPoint; { содержит точки кольца }
  xL,xR: TLines; { уже просмотренные значения x налево и направо от центра }
  iB,iE,Lim: integer;
  L,d2Lim,L2,deltaLim: longint;
  procedure shellSort(nSize: integer); { сортировка Шелла }
  { процедура сортирует буфер по возрастанию значения d2 }
  var step,i,j: integer;
      tmpP: TPoint;
  function sCondSwap(i,j: integer): boolean; { сравнение + обмен }
  var result : boolean;
  begin
    result := buf[j].d2 > buf[i].d2;
    if result then begin
      tmpP := buf[j]; buf[j] := buf[i]; buf[i] := tmpP;
    end;
    sCondSwap := result;
  end; {sCondSwap}
begin
  step := 4;
  while (3*step < nSize) do step := 3*step+1;
  while (step>1) do begin {step = 364,121,40,13,4,1 = (3^k-1)/2 }
    step := step div 3; { формула рекомендована Д.Кнутом }
    for i := step + 1 to nSize do begin
      j := i; { здесь работает сортировка простыми вставками }
      while (j>step) and sCondSwap(j,j-step) do Dec(j,step);
    end
  end
end; {shellSort}
procedure updateBuffer;
var i: integer;
    d: longint;
function dist2(xP,yP: integer): longint;
{ расстояние от точки до прямоугольника }
var xx,yy: longint;
begin
  xx := xP; yy := yP;
  if xP = 0 then xx := xInit else if xP < 0 then Inc(xx,M);
  if yP = 0 then yy := yInit else if yP < 0 then Inc(yy,N);
  dist2 := sqr(xx-xInit)+sqr(yy-yInit);
end; { dist2 }
procedure addPoint(xP,yP: integer; d: longint);
begin
  Inc(iE);
  with buf[iE] do begin
    x := xP; y := yP; d2 := d;
  end
end; { addPoint }
procedure advanceLines(var l: TLines; xStart,yStart,xD,yD : integer);
begin
  y := yStart;
  repeat
    d := dist2(l[y],y);
    if d <= d2Lim then begin { пока расстояние не превосходит предельного }
      addPoint(l[y],y,d); { скользим по этой полосе }
      Inc(l[y],xD);
    end
    else Inc(y,yD); { переходим на следующую полосу }
  until (y <> yStart) and (l[y-yD] = xStart);
  {если и на предыдущей полосе никуда не продвинулись, то выходим }
end; { advanceLines }

```

```

begin
  iB := 1; iE := 0;           { опустошаем буфер           }
  advanceLines(xR,M,N, M, N); { просматриваем I квадрант   }
  advanceLines(xL,0,N,-M, N); { просматриваем II квадрант  }
  advanceLines(xL,0,0,-M,-N); { просматриваем III квадрант }
  advanceLines(xR,M,0, M,-N); { просматриваем IV квадрант  }
  shellSort(iE);
end; {updateBuffer}
function nextDist2(var xNext,yNext: integer): longint;
{ В своих параметрах функция возвращает координаты очередной точки }
{ а результат функции - квадрат расстояния от прямоугольника до исходной точки}
begin
  while iB > iE do begin      { если буфер пуст }
    if d2Lim = L2 then begin  { если достигнут край круга}
      nextDist2 := L2 + 1;    { можно выходить }
      exit;
    end;
    Inc(d2Lim,deltaLim);      { пересчет внешнего радиуса }
    if d2Lim > L2 then d2Lim := L2;
    updateBuffer;
  end;

  with buf[iB] do begin
    nextDist2 := d2;
    xNext := x; yNext := y;
  end;
  Inc(iB);
end; {nextDist2}
procedure ReadData;
begin
  assign(input,'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  read(M); read(N); read(L);
  read(x); read(y);
end;
procedure Init;
var i : integer;
begin
  xInit := x mod M; if xInit<0 then Inc(xInit,M);
  yInit := y mod N; if yInit<0 then Inc(yInit,N);
  xOffset := x - xInit; yOffset := y - yInit;
  L2 := L*L;           { квадрат предельного расстояния }
  iB := 1; iE := 0; { начало и конец заполненной части буфера }
  Lim := 0; { верхний радиус последнего кольца }
  d2Lim := -1; { его квадрат }
  deltaLim := (longint(nBuf)*M*N) div 6;
  { шаг изменения квадрата внешнего радиуса кольца }
  { константа 6 достаточно произвольна, при ней буфер заполнялся примерно на 1/2 }
  for i := Low(xL) to High(xR) do begin { уровень просмотра на вертикалях }
    xL[i] := 0; xR[i] := M;           { xL - влево, xR - вправо}
  end;
end;
procedure outPoint(xP,yP:integer);
begin
  writeln(xP+xOffset,' ',yP+yOffset);
end;
procedure Solve;
var xNext, yNext : integer;
begin
  while nextDist2(xNext,yNext) <= L2 do {берем новую точку из буфера }
    outPoint(xNext,yNext);           { выводим ее}
end;
begin
  ReadData;
  Init;
  Solve;
end.

```



Выпуск 3

ИНФОРМАТИКА **в виртуальной школе**

А.И. СЕНОКОСОВ

**Пробный учебник
для 7-го класса
общеобразовательной школы**



СОДЕРЖАНИЕ

Слово к учителю
Предисловие

ГЛАВА 1. ВВЕДЕНИЕ В ИНФОРМАТИКУ

- § 1. Информация
- § 2. Компьютер
- Практикум № 1. Первый раз в компьютерном классе
- § 3. Кодирование символьной информации
- § 4. Кодирование графической информации. Часть 1
- § 5. Видеокарта
- § 6. Кодирование графической информации. Часть 2
- § 7. Кодирование графической информации. Часть 3. (Для дополнительного чтения)
- § 8. Телекоммуникации
- § 9. Файловая система

ГЛАВА 2. КОМПЬЮТЕРНАЯ ГРАМОТНОСТЬ

- Практикум № 2. Прогулка по файловому дереву
- § 10. Электронные документы
- Практикум № 3. Моя первая страничка в “Виртуальной школе”
- § 11. HTML — язык описания электронных документов
- § 12. Первое знакомство с “отмеченными объектами”
- § 13. Работа в многооконной многозадачной системе
- Практикум № 4
- § 14. Компьютерная обработка графической информации. Графические редакторы
- Практикум № 5. Стандартные инструменты CorelDraw!
- Практикум № 6. Схема пути до дома. Часть 1
- Практикум № 7. Работа с текстом
- § 15. Компьютерная обработка графической информации.
Знакомство с растровым редактором
- § 16. Понятие о буфере обмена
- § 17. Использование тега <Table> для оформления HTML-странички
- § 18. Обработка фотоизображений в Adobe Photoshop
- § 19. Слияние двух изображений с помощью Adobe Photoshop.
(Для дополнительного чтения) 3
- § 20. Основные форматы графических файлов в Интернете 6
- Практикум № 8. Создание картинка с прозрачным фоном 8
- Практикум № 9. Создание кадров для мультипликации 10
- Практикум № 10. Механизм управления в файле GIF-формата 12
- § 21. Текстовый редактор 14
- Практикум № 11. Набираем любимое стихотворение 16
- § 22. Стандартная панель инструментов и некоторые дополнительные возможности
редактора Microsoft Word 18
- § 23. Публикация документов, подготовленных в Microsoft Word,
в “Виртуальной школе” 21
- Заключение 23

§ 19. Слияние двух изображений с помощью Adobe Photoshop (Для дополнительного чтения)

Головной болью любого фотографа при съемках под открытым небом всегда является это самое небо. Будучи естественным фоном, оно очень сильно влияет на само настроение фотографии, делая ее либо солнечной и приветливой, либо мрачной и жуткой.

Если вы, по счастью, живете в таких местах, как Забайкалье с его почти тремястами солнечными днями в году, то вам нелегко будет понять, например, уральцев, которые месяцами могут не видеть голубого неба.

И, разумеется, очень не хочется, чтобы чудесные уголки родного города смотрелись весьма угрюмо только потому, что их сфотографировали в самую обычную погоду. То есть при сером пасмурном небе.

Возможностей Adobe Photoshop вполне хватает, чтобы помочь нам и в этом случае. Итак, возьмем цветную фотографию (см. *рис. 69*).



Рис. 69. Исходная фотография часовенки

Глазом, наметанным в предыдущем параграфе, вы сразу же определите, что:

1. Фотография требует небольшого поворота против часовой стрелки.
2. Ее ракурс не очень удачен: здание часовни не является абсолютным центром внимания. Оправданным было бы увеличение области кадра по вертикали и уменьшение по горизонтали.
3. Фотография нерезкая.
4. Серое небо без облаков делает изображение плоским и невыразительным.

Несмотря на то что первые три проблемы решаются довольно просто и, мало того, извест-

ными вам способами (кстати, какими?), начнем мы все же с четвертой. И немного погодя вы поймете почему.

Дело в том, что в стандартных изображениях, поставляемых с редактором Adobe Photoshop, есть изображение чудесного голубого неба с живописными облаками. Вот именно им мы и воспользуемся, чтобы вклеить в него фрагмент нашей картинки.

Мы научились вырезать часть картинку и вклеивать ее в другую, когда в § 16 засаживали деревцами пустынный скверик. Но там все было довольно просто благодаря прямоугольному размеру фрагмента. В нашем же случае вид фрагмента довольно-таки замысловатый: необходимо вырезать все, кроме “никакого” неба...

Однако давайте начнем работу по порядку.

Итак, запускаем программу Adobe Photoshop и открываем два файла: один — с исходной фотографией, а другой — с изображением неба. Он называется BIGSKY.TIF. Не обращайте внимания на неведомый вам тип этого файла. TIF — один из наиболее распространенных стандартов сохранения графического изображения на жестком диске, а хорош он тем, что его правильно понимают практически все современные графические редакторы.

Теперь убедимся в том, что изображение неба заведомо больше изображения с часовней. Для этого просто закажите просмотр обеих картинок в реальном размере и сравните их (см. *рис. 70*).

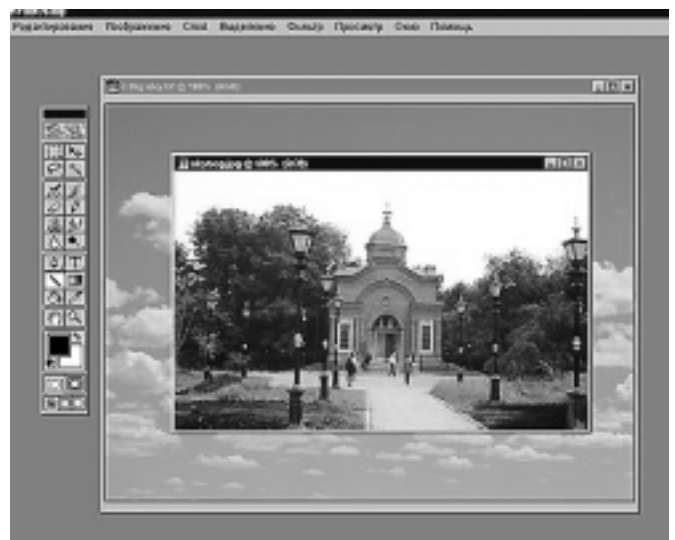




Рис. 70. Сравнение двух изображений

Если изображение часовни оказалось больше, увеличьте размер картинку с небом (надемся, вы помните, как это делается, а если нет — прочитайте еще раз § 16).

Теперь нам потребуется инструмент, позволяющий вырезать довольно сложный контур. Он называется **лассо** и бывает различных типов. Нам потребуется **прямоугольное лассо** (см. рисунок справа). 

Для того чтобы выбрать этот инструмент, нажмите на изображение обычного лассо (см. рис. справа) и, не отпуская кнопки мыши, сдвиньте указатель вправо. Точно так же мы выбирали инструмент **рамка** в § 18. 

Прямоугольное лассо работает так же, как и инструмент **карандаш** в редакторе CorelDraw! То есть контур чертится отрезками прямых линий, чьи концы закрепляются на рисунке щелчками левой кнопки мыши.

Итак, наша цель — отрезками прямых линий очертить довольно сложную верхнюю часть фрагмента, включающую деревья и купол. Что же касается нижней и боковых границ, то их можно сделать абсолютно прямыми.

Эта работа требует терпения, аккуратности и спокойствия. Начните, например, с левой границы, с точки, где зелень дерева переходит в серость неба, и не торопясь обводите короткими прямыми линиями контур деревьев и купол часовни.

Будьте осторожны! Если вы нечаянно дважды нажмете на левую клавишу мыши, не сдвинув ее ни на миллиметр, вырезаемый контур замкнется — автоматически будет проведена прямая линия от точки, в которой вы щелкнули дважды, до самой первой точки, с которой и начиналась работа. Тогда ничего не останется, как выбрать пункт верхнего главного меню **Выделение**, подпункт **Ничего не выбрано**, и начать все с самого начала.

После того как верхний контур ценой изрядных усилий пройден, проведите прямую линию вдоль правой границы, вдоль нижней границы и дважды щелкните на последней точке, располагающейся в нижнем левом углу картинке. Вы получите обведенный пунктиром фрагмент. Примерно такой, как показан на *рис. 71*.



Рис. 71. Исходный рисунок с выделенным пунктиром фрагментом

Следующая операция вам уже знакома. Необходимо скопировать выделенный фрагмент в буфер обмена (**Редактировать — Скопировать**), перейти на рисунок облаков и вставить в него содержимое буфера (**Редактировать — Вставить**).

Получится примерно вот такая картина (см. *рис. 72*).




Рис. 72. Вид картины с облаками после вклеивания фрагмента

Теперь возьмите инструмент **перемещение** (на *рис. 72* на него показывает черная стрелочка) и с его помощью подвигайте вклеенный фрагмент, добываясь наиболее удачного расположения облаков на заднем плане.

После этого:

1. поверните всю картинку примерно на один градус ПРОТИВ ЧАСОВОЙ СТРЕЛКИ;
2. с помощью **рамки** обрежьте изображение облаков снизу и немного измените ракурс, увеличив высоту и уменьшив ширину фотографии.

Прежде чем продолжить обработку фотографии с помощью известных вам инструментов и эффектов, необходимо сгладить резкий переход на линии соприкосновения двух изображений. Он отчетливо виден, если использовать инструмент **лупа** (инструмент — на рисунке справа, изображение — на *рис. 73*). 

Объединим два рисунка в один, чтобы дальнейшие операции совершались теперь уже со всеми видимыми пикселями. До этого момента на нашей картине совместно уживались два слоя, в одном находилось изображение неба, а в другом — фрагмент с часовней.



Рис. 73. Резкий переход между изображениями из-за “обрезанных” краев

Для этого выберите пункт меню **Слой**, подпункт **Объединить видимые слои**. Вообще работа со слоями — это захватывающее занятие, раскрывающее громадные возможности... Впрочем, об этом вам придется прочитать в другой книжке, посвященной исключительно редактору Adobe Photoshop.

Теперь с помощью кисточки и еще одного инструмента, называющегося **палец**, сгладьте границу перехода. Делать это лучше всего с использованием лупы и дополнительного окна **Кисти**, которое можно отобразить на экране, выбрав пункт меню **Окно**, подпункт **Показать кисти**.

Ну, что делают кисти, долго объяснять не надо. С их помощью, выбирая цвет краски пипеткой, можно аккуратно подкрасить белесые полосы, оставшиеся на границе рисунков от старого неба.

Палец же просто размазывает краску, сдвигая ее в сторону своего движения. Иными словами, если вы поставите **палец** на голубой фон, нажмете левую клавишу мыши и, не отпуская ее, начнете двигать в сторону купола, то голубая краска будет “наплывать” на купол.

Впрочем, не надо стремиться совсем уж избавляться от белого ореола вокруг купола. Небольшая белесая полоска придаст куполу эффект свечения.

После сглаживания границы добавьте резкость и примените другие средства выправления фотографии, которые мы освоили в предыдущем параграфе.

Под конец, как обычно, сравним исходную фотографию и ту, что получилась после “операции с пересадкой” (см. рис. 74).



Рис. 74. Сравнение исходной и полученной фотографий

Не говоря о цветовой коррективке, которая не видна в черно-белом издании, отметим, что четкое изображение часовни стало важнейшим, завершающим элементом композиции (профессионалы бы сказали: *доминантой*). Это подчеркивается и рядом ажурных фонарей, сходящихся вдоль дорожки к зданию, и более высоким небом над куполом, и легким эффектом сияния. Остаточная небольшая белизна вдоль границ рисунков придает фотографии объемность, а изменение яркости и контрастности добавляет солнечности.

Подведем итоги:

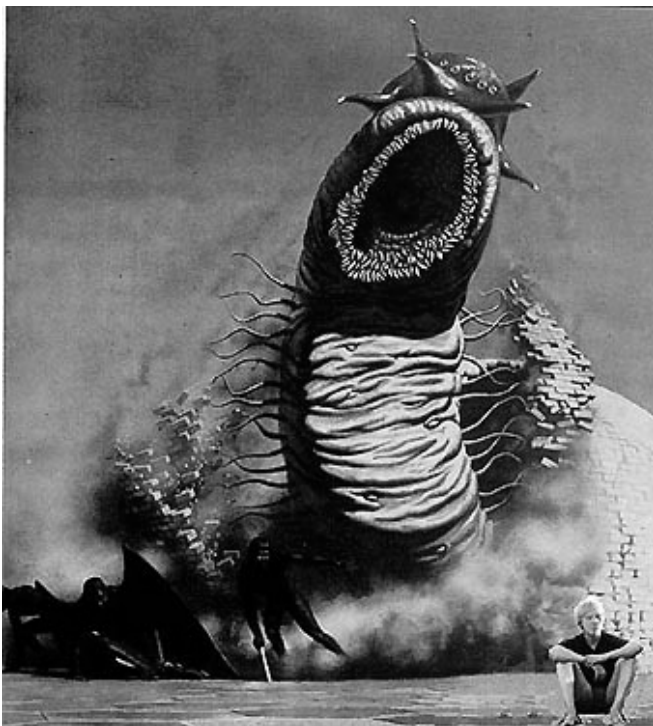
- Профессиональные растровые редакторы предоставляют возможность не просто

улучшить фотографию, но и художественно изменить ее, добиваясь более сильной ее выразительности.

- Одним из средств такого художественного изменения является **коллаж**, или объединение нескольких фотографий в одну. С появлением компьютерных средств обработки фотоизображений коллаж перестал быть уделом профессиональных фотомастеров. Теперь им с успехом могут заниматься и семиклассники.

Вопросы и задания:

1. Создайте свой собственный коллаж. Это может быть работа по изменению фона (в частности, неба) или любое другое объединение — слияние нескольких изображений. В качестве еще одного примера может служить рисунок, на котором в фантастический пейзаж помещена обычная фотография школьника.



§ 20. Основные форматы графических файлов в Интернете

Работая с “Виртуальной школой”, вы постоянно имеете дело с картинками, записанными с двумя расширениями — GIF и JPG. Напоминаем, что расширения говорят о различных спо-

собах кодировки графического изображения. И сегодня мы поговорим об этих двух кодировках поподробнее.

Как вы помните, любое графическое изображение кодируется все теми же ноликами и единичками. И если мы видим его на экране монитора, то информация о каждом пикселе в несжатом виде находится в видеопамяти (см. § 4—5).

Иными словами, в компьютере каждому цвету соответствует некоторое число, и если первые, скажем, полторы тысячи пикселей на экране темно-синие, в видеопамяти записаны полторы тысячи совершенно одинаковых чисел.

Понятно, что прямая запись содержимого видеопамяти на “винчестер” привела бы к совершенно неоправданным затратам внешней памяти: вместо тех же полутора тысяч абсолютно одинаковых чисел вполне можно было бы записать нечто вроде 1500 (37), где 37 — это номер темно-синего цвета.

Такая запись называется сжатием информации, и чем больше в графическом изображении пикселей с одинаковыми цветами, тем сильнее можно сжать информацию, тем меньше получится файл, соответствующий данной картинке.

Форматы, соответствующие файлам с расширением GIF и JPG, наиболее удачно на сегодняшний день сжимают видеоинформацию и именно поэтому широко используются в Интернете. Однако между ними есть довольно много различий, которые сведены в таблицу, расположенную на с. 7.

Прокомментируем отдельные части таблицы.

Наверняка у вас возник вопрос, каким же образом можно регулировать качество графического изображения и, соответственно, уменьшать (или увеличивать) размер файла, содержащего информацию о картинке.

Не уменьшая размеров картинки, достичь этого можно достаточно просто: уменьшив количество цветов, использующихся в изображении. Посмотрите еще раз внимательно на текст в таблице. Он означает, что мы можем выбрать какое-то количество цветов из более чем 16 миллионов. Ну, скажем, 120 тысяч или просто 120. И в том, и в другом случае каждый цвет будет кодироваться тремя байтами, но во втором случае программа сжатия сработает неизмеримо эффективнее, резко уменьшив размер файла.

Конечно, 120 цветов — это далеко не 120 тысяч, и качество такого изображения, если это, например, фотография, несравнимо хуже качества фотографии со 120 тысячами цветов. Но это плата за небольшой размер файла.

Особенности и возможности форматов, соответствующих расширениям GIF и JPG

*.JPG	*.GIF
В файле сохраняется информация только о картинке.	В файле сохраняется информация о картинке и палитре, в которой она нарисована (это называется индексированным режимом).
Каждый цвет кодируется 3 байтами. Максимально возможное количество цветов — более 16 миллионов.	Каждый цвет кодируется либо 1 битом (2 цвета), либо двумя (4 цвета), либо тремя (8 цветов), ... максимум — восемью битами (256 цветов).
При записи существует возможность регулировать качество с соответственным изменением размера файла на диске.	Изображение записывается в память на внешних носителях без потери качества.
	Возможно объединение нескольких изображений в один файл с их чередованием при показе.
	Один из цветов сохраняемой палитры можно объявить “прозрачным”.

С другой стороны, несмотря на то, что файлы с расширением GIF вроде бы сохраняют ту же самую фотографию без потери качества, общего количества цветов в них (максимум — 256) явно недостаточно для хорошего воспроизведения изображения.

Мало того, зачастую даже картинки, содержащие заведомо меньше 256 цветов, гораздо лучше сжимаются именно в файлах с расширением JPG, причем их качество при этом практически не страдает.

Однако есть два свойства формата GIF, которые делают его поистине уникальным. При этом оба они прекрасно вам известны.

1. Наверняка в вашей страничке или страничке вашего приятеля используется (может, даже не одно) движущееся (или, говоря профессиональным языком, **анимированное**) изображение. И создано оно благодаря возможностям формата GIF.

2. Очень может быть, что вы использовали и изображения с прозрачным фоном, когда картинка как будто нарисована на выбранном вами общем фоне.

Конечно, уже сейчас есть форматы, позволяющие продемонстрировать на страничке живое полноцветное видеоизображение, а в будущем наверняка появятся форматы, сжимающие графическую информацию лучше, чем это делает-

ся в файлах с расширением JPG. Но пока с помощью файлов этих двух типов передается почти вся графическая информация в Интернете.

Подведем итоги:

- Почти вся графическая информация в Интернете передается файлами с расширениями GIF и JPG.
- Соответствующие им форматы обеспечивают наилучшее на сегодняшний день сжатие картинок (JPG), выразительное использование прозрачного фона (GIF) и эффекты анимации (GIF).

Вопросы и задания:

1. Запустите программу Adobe Photoshop, откройте в ней какую-либо яркую цветную фотографию и сохраните ее в новом файле с расширением JPG, используя максимальный уровень сжатия (или “нулевое” качество). Откройте этот новый файл и сравните, насколько ухудшилось изображение.
2. Ту же самую цветную фотографию переведите в индексированный режим (выбрав пункт главного меню **Изображение**, подпункт **Режим**, подпункт **Индексированные цвета**), выберите вариант 256 цветов и со-

храните в новый файл с расширением GIF. Откройте этот новый файл и сравните исходное изображение и два только что полученных.

3. Сформулируйте выводы, которые можно сделать после такого сравнения.

Практикум № 8. Создание картинка с прозрачным фоном

Итак, из предыдущего параграфа вы должны были понять, что

1. В файле формата GIF сохраняется не только информация о графическом изображении, но и о наборе цветов (палитре), используемом в этой картинке.
2. Один из цветов палитры (например, фоновый) может быть объявлен прозрачным, и тогда при размещении картинка на веб-страничке сквозь него будет просвечивать фон этой самой странички.

Конечно же, вам не терпится поскорее создать свою собственную картинку с прозрачной основой, но сначала давайте порассуждаем самую малость.

Итак, можем ли мы вот так, с ходу, объявить на какой-нибудь картинке, а тем более фотографии какой-либо цвет прозрачным?

Понятно, что ни на одной фотографии практически невозможно найти однородный фон, составленный из пикселей одного цвета. Но и на рисованной картинке очень часто цвет фона используется и при рисовании отдельных фрагментов интересующего нас объекта. И если синим цветом нарисован не только фон, но и, скажем, глаза, то вовсе не хочется пугаться, увидев на своей страничке фон, просвечивающий сквозь дырки глазниц.

Итак, возьмем в качестве начальной картинка фотографию автомобиля (см. *рис. 75*).



Теперь нам потребуется отделить изображение автомобиля от всего лишнего: фрагмента здания, растительности и асфальта.

Для этого необходим инструмент, позволяющий вырезать довольно сложный контур. Он называется **лассо**, и о нем мы уже достаточно подробно писали в § 19. Возможно, вы не только мельком глянули на этот материал для дополнительного чтения, но и внимательно его изучили и даже смогли сделать описанную в нем работу.



Рис. 75. Фотография автомобиля

Тогда можете со спокойной совестью пропустить пару абзацев. Для остальных же мы расскажем об этом инструменте еще раз.

Итак, лассо бывает различных типов. Нам потребуется **прямоугольное лассо** . Для того чтобы выбрать этот инструмент, нажмите на изображение обычного лассо  и, не отпуская кнопку мыши, сдвиньте указатель вправо. Точно так же мы выбирали инструмент **рамка** в § 18.

Прямоугольное лассо работает так же, как и инструмент **карандаш** в редакторе CorelDraw! То есть контур чертится отрезками прямых линий, чьи концы закрепляются на рисунке щелчками левой кнопки мыши.

Итак, наша цель — отрезками прямых линий очертить автомобиль. Эта работа требует терпения, аккуратности и спокойствия. Не торопясь обведите короткими прямыми линиями контур машины.

Будьте осторожны! Если вы нечаянно дважды нажмете на левую клавишу мыши, не сдвинув ее ни на миллиметр, вырезаемый контур замкнется — автоматически будет проведена прямая линия от точки, на которой вы щелкнули дважды, до самой первой точки, с которой и начиналась работа. Тогда ничего не останется, как выбрать пункт меню **Выделение**, подпункт **Ничего не выбрано**, и начать все с самого начала.

После того как вы закончите работу, в окне редактора Adobe Photoshop будет примерно такая картинка с контуром автомобиля, очерченным мерцающей пунктирной линией (см. *рис. 76*).



Рис. 76. Контур автомобиля, обведенный мерцающей пунктирной линией

Теперь, для того чтобы немного сгладить угловатости вашего нарисованного контура, используем *растушевку*, т.е. как бы равномерно размажем края обведенного контура.

Для этого выберите пункт главного меню **Выделение**, подпункт **Растушевка...**, и в появившемся окне задайте радиус растушевки границы в 1 пиксель (см. рис. 77).

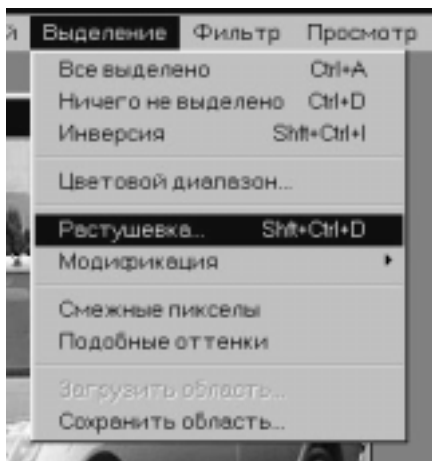


Рис. 77. Расположение пункта меню **Растушевка**

Скопируйте выделенный фрагмент с растушеванными границами в буфер обмена, используя уже давно известную операцию **Редактировать — Скопировать**.

Воспользуемся теперь известным фактом, что если в буфере обмена находится графическая информация, то Adobe Photoshop предлагает создать новую картинку как раз под ее размеры.

Выберите пункт меню **Файл**, подпункт **Новый**, и согласитесь с предложенными размерами. Единственное, на что следует обратить особое внимание, — **НЕОБХОДИМО ПОСТАВИТЬ ТОЧКУ** в белом кружке, напротив которого на-

писаны слова “Прозрачная основа” (на рис. 78 на это место показывает стрелочка).

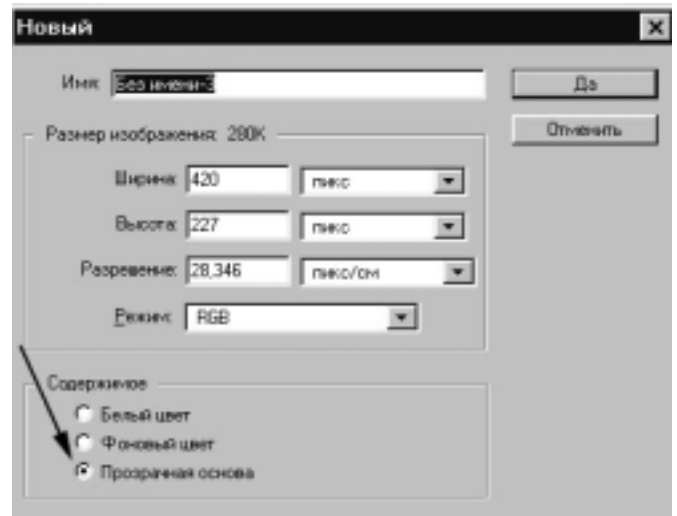



Рис. 78. Окно создания нового полотна

Заключительные операции совсем уж простые.

1. Вставьте в новое полотно содержимое буфера обмена (**Редактировать — Вклеить**).
2. С помощью инструмента **перемещение**, изображенного на рисунке , передвиньте, если это необходимо, вклеенную картинку.
3. Экспортируйте получившуюся картинку в формат GIF89a (**Файл — Экспорт — GIF89a**), записав ее в свою папку в “Виртуальной школе”.
4. С помощью обычного тега `` вставьте изображение с прозрачным фоном в свою web-страничку.

Если у вас получилось нечто похожее на рис. 79, можете считать, что создание прозрачного фона вы освоили успешно.

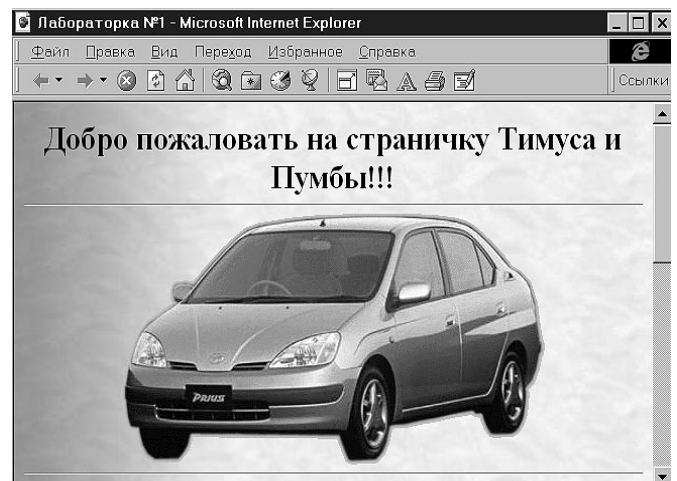


Рис. 79. Вид странички с “прозрачной” иллюстрацией

Практикум № 9. Создание кадров для мультипликации

Напомним, что в § 20 речь шла о достоинствах формата GIF. Одно из них — прозрачный фон — мы уже научились создавать в предыдущем практикуме. Со вторым познакомимся поближе сегодня.

Итак, в файле формата GIF хранится не просто одно изображение, как в файле с расширением JPG. И не просто изображение с палитрой. Оказывается, там можно расположить целый склад картинок с механизмом их извлечения на свет.

Схема этого склада приведена на *рис. 81*.

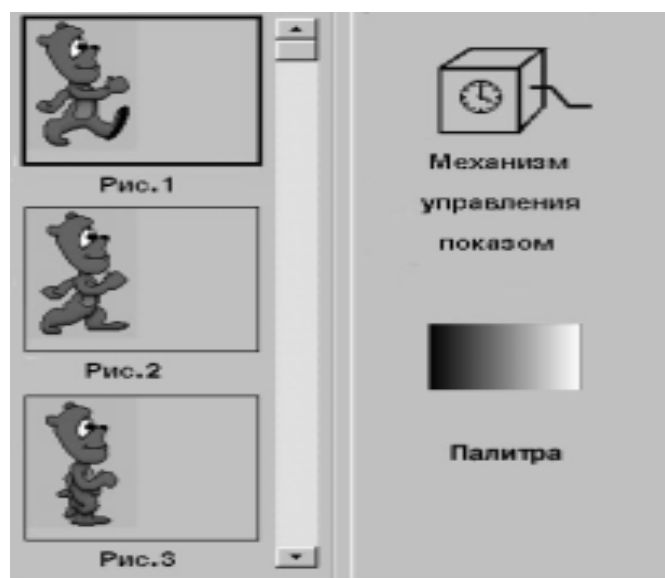


Рис. 80. Схема графического файла GIF-формата

И если с картинками и палитрой вам, наверное, все более или менее понятно, то механизм управления показом картинок выполняет следующие действия:

1. Определяет, сколько времени (в сотых долях секунды) данный рисунок находится на экране.
2. Устанавливает, каким образом меняется картинка. Возможен такой случай, когда, скажем, рисунок № 1 постоянно будет находиться на экране, а все остальные будут чередоваться на его фоне. Это очень часто используется в “движущихся” картинках, когда, например, неподвижный человек машет рукой.
3. Определяет, сколько раз будет повторяться показ всех картинок, начиная с первой. Возможно, эти повторения будут бесконечными.

У механизма есть и некоторые другие функции, с которыми вы при желании легко разберетесь и самостоятельно. Нам же пока хватит и этих трех.

Можно, конечно, расположить на этом своеобразном GIF-складе картинки, имеющие друг с другом мало общего. Ну, скажем, виды родного города. А можно (и это используют гораздо чаще) поместить картинки, представляющие фазы движения какого-либо предмета. Такие, например, как на нашем рисунке.

Как вы знаете, любой художественный фильм или мультик представляет собой набор неподвижных изображений-кадров, которые меняются с частотой 24—25 раз в секунду. Человеческий глаз воспринимает такое мелькание как иллюзию движения, на чем, собственно говоря, и держится весь кинематограф.

Иными словами, если мы быстро сменим даже всего лишь две-три картинки, этого вполне достаточно для создания *анимационного*, или движущегося, изображения.

По счастью, в нашем распоряжении находится целый склад самых разнообразных фрагментов веселых картинок. Он располагается в “Виртуальной школе” в папке DETALI. Из него вполне можно выбрать несколько картинок, объединить их и создать анимационный пакет — GIF-изображение.

В качестве примера возьмем, скажем, три фазы полета птички и одну голову (см. *рис. 81*). Конечно, вы можете взять и любые другие детали рисунков. Никто не мешает объединять их и с изображениями, находящимися в самых различных папках “Виртуальной школы”.



Рис. 81. Исходные рисунки из папки DETALI, взятые для создания анимированного GIF-изображения

Перед нами, таким образом, стоит достаточно сложная задача:

1. Необходимо присоединить изображение головы к изображению туловища.
2. Обработать картинку так, чтобы они стали картинками с прозрачным фоном.
3. Объединить картинки в соответствии с фазами движения в одном файле.
4. Вставить картинку в собственную web-страничку.

Сегодня мы постараемся выполнить хотя бы первые две задачи.

Итак, запустите уже известную вам программу Adobe Photoshop и откройте в нем файлы со всеми четырьмя картинками.


Теперь обратим внимание на то, что размер полотна с птичьими “тушками” явно недостаточен, чтобы к ним пристыковать еще и голову. Щелкните мышкой по одной такой картинке и с помощью пункта меню **Изображение**, подпункта **Размер изображения**, определите этот самый размер. Пусть, для примера, он оказался 285 пикселей в ширину и 119 в высоту.

Значит, перво-наперво нам надо создать три пустых полотна размером несколько больше этого.

Выберите пункт меню **Файл**, подпункт **Новый** и создайте три новых полотна размером, скажем, 340×190 пикселей. Не забудьте при этом заказать прозрачную основу.

Теперь воспользуемся одним из пустых полотен для создания первого рисунка — объединения туловища и головы.

Конечно, вполне можно воспользоваться нашим предыдущим опытом и аккуратно вырезать туловище по контуру, используя инструмент **прямоугольное лассо**. Но в том случае, когда рисунок расположен на одноцветном фоне, можно добиться того же гораздо проще.

Для этого нам потребуется инструмент **волшебная палочка** . Действует он следующим образом.

После того, как **волшебной палочкой** ткнули в какой-нибудь пиксель, отмечаются все граничные с ним пиксели того же цвета. Затем проверяются пиксели, лежащие рядом с граничными, и так до тех пор, пока не будут отмечены, а затем и выделены в единое целое все пиксели того же цвета, что и исходный.

Для простоты представьте себе, что в большой праздничной толпе поступила команда взяться за руки всем людям в синих кепочках, начиная с какого-либо конкретного человека. Сходить с места при этом запрещается. И если

таковых окажется много, в каком-то месте возникнет “толпа внутри толпы” людей, взявшихся за руки.

Вот такой процесс объединения и начинается **волшебная палочка**. В выбранную область при этом не попадут пиксели одного цвета с исходным, если их везде разграничивают пиксели других цветов.

Итак, возьмите **волшебную палочку** и прикоснитесь к фону первой картинке. Вы получите нечто похожее на *рис. 82*.

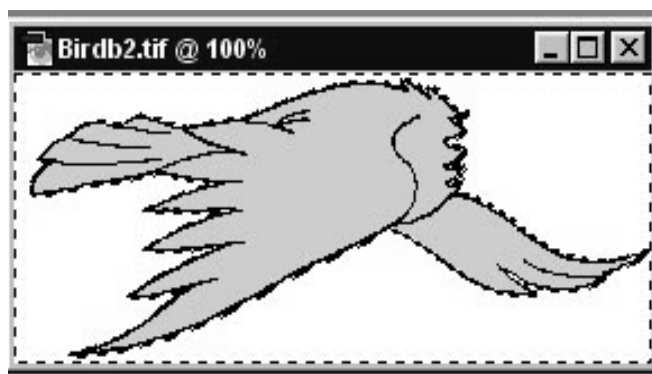


Рис. 82. Выделение с помощью **волшебной палочки**


Видно, что выделилось вовсе не изображение тела птицы, а фон, его окружающий.

Для того чтобы выделить нужные нам пиксели, воспользуемся пунктом меню **Выделение**, подпунктом **Инверсия**.

Вообще слово “инверсия” означает “переворачивание, перестановка”. В нашем случае те пиксели рисунка, которые не были выделены, станут выделенными, и наоборот.

С помощью **волшебной палочки** и инверсии нам удалось чуть ли не мгновенно очертить контур птички без всякого **прямоугольного лассо**. Повторяем, такое возможно лишь в том случае, когда фон картинки одноцветный.

Теперь скопируем выделенный фрагмент в буфер обмена (**Редактирование — Копировать**), перейдем на одно из пустых полотен и вклеим его туда (**Редактирование — Вклеить**).

Если это необходимо, подправьте расположение вклеенного фрагмента с помощью известного вам инструмента **перемещение**  и повторите операцию с **волшебной палочкой**, **инверсией**, **копированием** и **вклейкой**, но теперь уже с изображением головы.

Если вы все сделали правильно, то в результате у вас должна получиться примерно такая картинка (см. *рис. 83*).

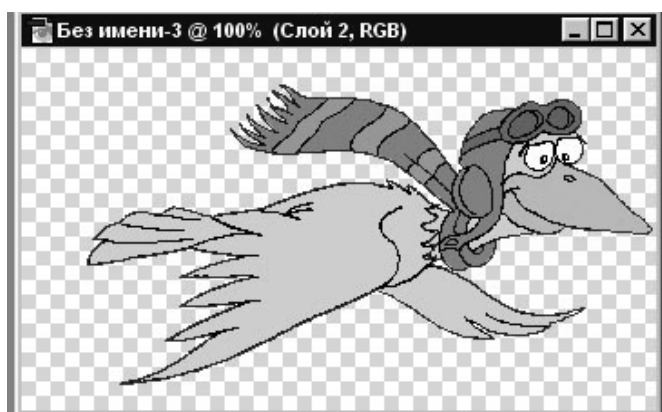


Рис. 83. “Тушка” птички с пристыкованной головой на прозрачном фоне

Теперь осталось только экспортировать эту картинку в формат GIF89a (**Файл — Экспорт — GIF89a**) и записать его в свою папку в “Виртуальной школе”.

Повторите эту операцию еще два раза, и вы получите три фазы полета птички, на основе которых в ближайшем будущем мы создадим файл формата GIF, содержащий анимационную картинку.

Практикум № 10. Механизм управления в файле GIF-формата

В предыдущем практикуме мы с вами создали три фазы полета птички, которые теперь самое время объединить в один анимированный файл GIF-формата.

Для этого существует довольно много программ, известных под общим названием “GIF-аниматоры”. Основной их функцией является не только объединение нескольких картинок в одну, но и настройка механизма управления сменой кадров, о котором говорилось в предыдущем практикуме.

Мало того, некоторые из них способны сами создавать последовательность кадров, с помощью которых, например, создается иллюзия листания страничек, содержащих разные картинки.

Одним из самых простых и понятных GIF-аниматоров является программа Microsoft GIF Animator, с которой мы сегодня и познакомимся.

Несмотря на то, что она не переведена на русский язык, вам не потребуется знание английского для того, чтобы успешно в ней работать.

Предположим, картинка, изображающая птичку с максимально поднятыми крыльями, называется РТАНА1.GIF, с крыльями на среднем уровне — РТАНА2.GIF и с крыльями внизу — РТАНА3.GIF.

Понятно, что для достижения эффекта движения менять их нужно в следующем порядке:

РТАНА1.GIF
РТАНА2.GIF
РТАНА3.GIF
РТАНА2.GIF

Дублирование кадра РТАНА2.GIF нужно потому, что мы собираемся повторять смену картинок вновь и вновь. Стало быть, после последней картинки снова будет показана первая. Если бы мы не добавили среднюю фазу положения крыльев, они из самого нижнего положения без промежуточной стадии взметнулись бы вверх, нарушив иллюзию плавного полета.

Итак, запускаем программу Microsoft GIF Animator и собираем пакет из четырех картинок.

Собирать их лучше с конца. Поэтому сначала щелкните мышкой по пиктограмме открытия файла, изображенной на рисунке слева, и откройте файл РТАНА2.GIF.



Далее надо добавить в формируемый пакет картинки, сохраненные в файлах РТАНА3.GIF, снова РТАНА2.GIF и, наконец, РТАНА1.GIF. Это делается с помощью пиктограммы **Добавить (Insert)**, изображенной на рисунке справа.



После всех этих операций окно программы будет выглядеть следующим образом (см. рис. 84):



Рис. 84. Вид окна программы Microsoft GIF Animator после того, как в пакет вставлены все картинки

В окне выделяются три части:

1. Верхняя инструментальная панель, в которой мы уже воспользовались кнопками **Открыть файл** и **Вставить файл**.
2. Изображение всех картинок, включенных в пакет (слева).
3. Панель настройки механизма управления картинками, представляющая собой блокнот с тремя закладками — **Options**, **Animation**, **IMAGE**.

Рассмотрим более подробно пиктограммы инструментальной панели. На ней выделяются: группа кнопок, отвечающих за работу с файловой системой, кнопки управления буфером обмена и кнопки, с помощью которых можно изменять пакет картинок.

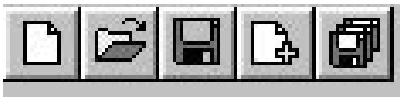


Рис. 85. Группа кнопок, отвечающих за работу с файловой системой

На рис. 85 изображены следующие кнопки (слева направо):

- **Новый файл** — прекращает работу над формированием пакетного файла GIF, если необходимо, предлагает его сохранить и закрывает.
- **Открыть файл** — открывает любой файл в формате GIF.
- **Сохранить файл** — сохраняет файл с тем именем, которое в данный момент отображается на самой верхней синей панели. **ВНИМАНИЕ!** Если использовать эту кнопку в процессе формирования файла из отдельных картинок, пакетный файл запишется на место самого первого, который мы открывали. В частности, в ситуации, изображенной на рис. 84, анимированный GIF запишется с именем РТАНАЗ.GIF.
- **Вставить картинку из файла в пакет.**
- **Сохранить как...** — дает возможность сохранить готовый анимированный файл с новым именем.

Рассмотрим следующую группу кнопок:



- **Вырезать картинку и поместить ее в буфер обмена** — здесь и далее работа идет с картинкой, выделенной синей рамкой. Любую картинку можно сделать таковой, щелкнув по ней мышкой.

- **Скопировать картинку в буфер обмена.**
- **Вставить картинку из буфера обмена.** Картинка вставляется ПЕРЕД выделенной синей рамкой.
- **Удалить картинку.**

И, наконец, оставшиеся 4 кнопки:



Выделить в синюю рамку все картинки. Эта функция очень нам понадобится чуть позже.

- **Переместить выделенную картинку вверх.**
- **Переместить выделенную картинку вниз.**
- **Предварительный просмотр анимации.**

А теперь, когда с помощью описанных клавиш картинки расположились в нужном порядке, примемся за механизм их показа.

Начнем с порядка смены картинок.

Сделаем так, что все картинки будут показываться десятую долю секунды, после чего полностью исчезнут.

Для этого:

- Щелкните по уже описанной пиктограмме .
- Щелкните по закладке **IMAGE**.
- В рамочке напротив слов **Duration (1/100 s)** — “Длительность показа в сотых долях секунды” — поставьте число 10.
- В рамочке под словами **Undraw Method** — “Способ появления” — выберите пункт **Restore Background** — “С предварительным восстановлением фона”.

Вообще-то установка всех этих параметров действует только на картинки, выделенные синей рамочкой. И мы не случайно с помощью специальной кнопочки сделали таковыми каждую из них.

Теперь обеспечим бесконечный показ последовательности картинок.

Для этого:

- Щелкните по закладке **Animation** на блоке управления.
- Щелкните мышкой в пустом квадратике, напротив которого написано **Looping** — “Цикличность”, поставив этим самым в нем галочку.
- Щелкните мышкой в пустом белом квадратике перед надписью **Repeat forever** — “Повторять беспрестанно”.

Вот теперь можно нажать кнопочку предварительного просмотра анимации и, если необходимо, изменить параметры управления.

Так, очень может быть, что для достижения выдающегося художественного эффекта необходимо разное время показа для разных кадров.

Вообще необходимо учесть, что на ваших веб-страничках анимированные картинки будут двигаться несколько быстрее, чем при предварительном просмотре.

Надеемся, наше объяснение было достаточно полным, вы без особых трудов создадите свой анимированный GIF-файл и вставите в свою страничку.

§ 21. Текстовый редактор

Вам наверняка приходилось читать чужой рукописный текст, и скорее всего это оказывалось значительно труднее, чем читать книгу. Сколько времени пройдет, пока разберешь все слова! К тому же какое-нибудь слово можно просто неправильно прочитать и в итоге — неверно понять общий смысл.

Давайте теперь проведем небольшой эксперимент.

- Возьмите часы с секундомером, засекайте время и напишите пару-другую предложений своим обычным почерком так, как вы пишете только для себя.
- А теперь снова засекайте время и напишите тот же текст так, как если бы писали его на поздравительной открытке — разборчиво и аккуратно.
- Посчитайте количество символов в тексте, включая, естественно, знаки препинания и пробелы, и определите свою скорость письма в символах в минуту в обоих случаях.

Вспомним начало учебного года и работу на тренажере. Если вы успешно его освоили, то наверняка ваша скорость клавиатурного ввода текста превосходит скорость разборчивого письма. И при этом даже самый разборчивый почерк не идет ни в какое сравнение с аккуратным, почти типографским шрифтом, которым печатает принтер.

Нет, мы вовсе не хотим сказать, что поздравления своим родным и друзьям надо непременно печатать на принтере. Аккуратный рукописный текст на поздравлении выглядит живым, доверительным.

Но когда речь идет о деловой переписке и подготовке документов, в том числе и электронных, тут просто необходимо совместить скорость создания текста, разборчивость и аккуратность, что очень даже непросто.

Итак, давайте более пристально рассмотрим технологию написания писем, книг и документов.

Конечно, можно сказать, что уж где-где, а здесь прогресс шел постоянно. Сначала надписи вырубали на камнях. Затем выдавливали острым стилем на глиняных дощечках. Царапали на бересте. Писали красной охрой на керамике и палочками на папирусах, кисточками на шелке и перьями на бумаге. Гусиные перья сменялись перьевыми ручками, те — авторучками. А если вспомнить шариковые и гелевые ручки, фломастеры, “мокрый шарик”, особо тонкие автоматические карандаши!.. А сколько типов пишущих машинок!..

Однако не менялось главное: чтобы внести изменения в текст, его надо было заново переписывать (если, конечно, стремиться к аккуратности). Вы по себе знаете, что такое черновик и сколько сил и времени отнимает переписывание начисто обычного школьного сочинения. Может быть, поэтому многие ученики больше думают не о содержании сочинения, а о том, как бы не допустить помарок.

А каково писателям! Переделка и переписывание книг еще совсем недавно были поистине каторжным трудом. Конечно, это шутка, но в каждой шутке есть доля истины.

Появление компьютеров коренным образом изменило технологию письма. С помощью специальной программы, которая называется **редактором текстов** (или **текстовым процессором**), на экране ЭВМ можно увидеть любой текст и внести в него (при необходимости) любые изменения. Хочешь — букву заменяй, хочешь — целую страницу.

С одним из них — Блокнотом — вы уже довольно хорошо знакомы. Это один из самых простых текстовых редакторов, но если вы в Практикуме № 3 хорошо освоили его стандартные инструменты, то у вас не возникнет никаких особых трудностей и при работе с гораздо более мощными редакторами.

Конечно, мы не использовали Блокнот для того, чтобы, например, написать с его помощью сочинение, а затем напечатать его на принтере, однако это вполне возможно. Между прочим, именно такие простейшие текстовые редакторы с небогатым набором стандартных инструментов и совершили переворот в тяжелой работе подготовки текстов, принципиально изменив ее характер.

Нынче, правда, стандартными инструментами дело не ограничивается. Точно так же, как и графический редактор, современный текстовый редактор — это принципиально новый инструмент для работы, имеющий множество уникаль-

ных функций, о которых и мечтать-то не приходится при использовании традиционных методов письма.

Эти функции иллюстрирует *рис. 86*, и скоро мы поговорим о них более подробно.

В дальнейшем мы будем использовать текстовый редактор Microsoft Word. Он стал фактически стандартным средством редактирования текстов в нашей стране. Это вовсе не означает, что, кроме него, не существует других редакторов или этот редактор — самый лучший. Просто так получилось, что он — самый распространенный. Итак:

Стандартные инструменты

В этом редакторе работают стандартные сочетания клавиш, о которых шла речь в Практикуме № 3 и с которыми вы привыкли работать. Единственное, но достаточно важное отличие касается абзаца.

В Блокноте мы можем отметить новый абзац только красной строкой, несколько раз нажав клавишу пробела, или пропустить перед его началом дополнительную строку. При этом невидимый символ “перевод строки”, о котором мы говорили в Практикуме № 3, вставляется в конец каждой строки.

В редакторе Microsoft Word символ “перевод строки” имеет смысл окончания абзаца, поэтому, нажимая на клавишу “Ввод” (**Enter**), вы говорите редактору, что необходимо либо сделать красную строку, либо чуть увеличить расстояние между строками... Одним словом, начать новый абзац. О переносе же на новую строку редактор заботится сам.

Именно из-за этого при работе с текстами, подготовленными в простых редакторах, Microsoft Word считает каждую строку новым абзацем, что изрядно добавляет хлопот при оформлении документов.

Панели инструментов

Очень многие люди пугаются, увидев впервые рабочее окно редактора (см. *рис. 87*). Однако большинство надписей и кнопок имеют очень простой и понятный смысл.



Стандартные инструменты



Табличный редактор и калькулятор



Набор шрифтов



Вставка графических объектов



Автоформатирование и автоперенос



Файловая система



Текстовый редактор



Объединение документов, многооконность



Проверка орфографии



Предварительный просмотр печатаемых страниц.

Рис. 86. Основные компоненты текстового редактора

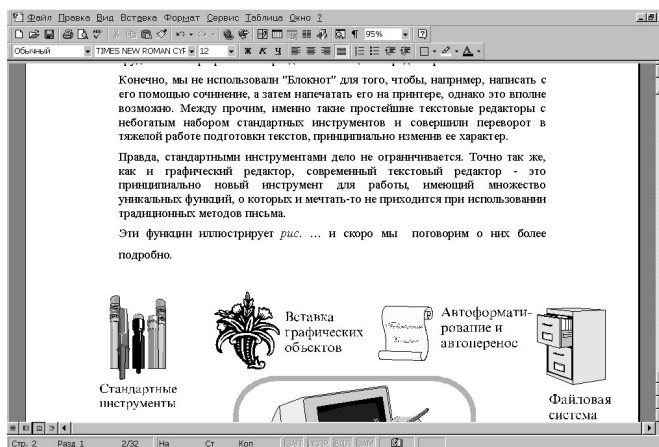


Рис. 87. Вид окна редактора Microsoft Word

Итак, присмотримся внимательнее к верхней части экрана. Ее условно можно разделить на следующие части:

1. Оконное меню, которое устроено точно так же, как и в Блокноте, и в графическом редакторе Adobe Photoshop. При нажатии мышкой на любую из надписей мы увидим вертикально выпадающую рамку подпунктов.
2. Первый горизонтальный ряд кнопок, который называется **Стандартной панелью**.
3. Второй горизонтальный ряд кнопок, который называется **Панелью форматирования**.
4. Может быть еще и третий ряд. Он называется **Панелью пользователя** и создается человеком для собственного удобства.

Далее начинается текст, в верхней части и слева от которого при необходимости располагаются горизонтальные и вертикальные линейки, позволяющие ориентироваться в реальных размерах документа. Линейки включаются и выключаются с помощью выбора пункта меню **Вид**, подпункт **Линейка**.

Более подробно мы рассмотрим эти панели, а также другие полезные функции Microsoft Word на практикумах.

Вопросы и задания

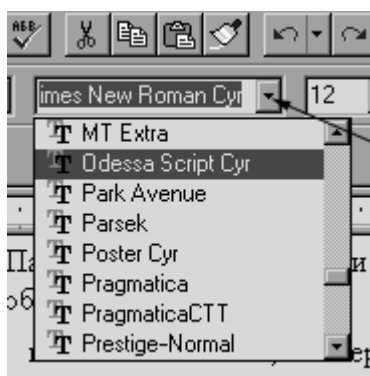
Какие, на ваш взгляд, свойства текстового редактора привели к революционному изменению подготовки и публикации текстов?

Практикум № 11. Набираем любимое стихотворение

Сегодня, используя редактор Microsoft Word, мы постараемся набрать свое любимое стихотворение. И не просто набрать, а воспользовавшись богатыми возможностями этой программы, сделать публикацию красивой и привлекательной. Каковы же эти возможности? Начнем по порядку.

Выбор шрифта

Если нажать левой клавишей мышки на кнопку с изображением смотрящего вниз треугольника, ту самую, на которую показывает черная стрелка, то появляется выпадающая рамочка.



Здесь можно выбрать шрифт, которым будет написан наш текст. Это как раз те шрифты, о которых рассказано в Практикуме № 7 нашего учебника.

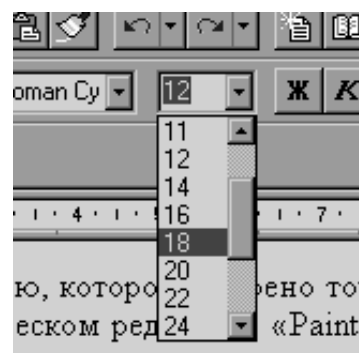
Все шрифты, которые вы видите в рамочке, установлены в вашем компьютере и доступны в любых программах, будь то CorelDraw!, Adobe Photoshop или, как сегодня, Microsoft Word.

Просмотреть весь довольно-таки обширный список шрифтов можно, «схватив» мышкой бегунок, расположенный в полосе прокрутки (т.е. перевести указатель мыши на бегунок, нажать левую клавишу мыши и, не отпуская ее, двигать бегунок вверх или вниз). Можно и просто нажимать на кнопки с треугольниками в верхней и нижней частях полосы прокрутки.

Выбранный вами шрифт, на названии которого вы щелкнули левой клавишей мыши, выделяется черной полосой.

Размеры шрифта

Кроме начертания, можно задать и размеры шрифта. Для этого точно так же, как и в случае выбора шрифта, надо нажать на кнопку, расположенную возле числа, показывающего размер шрифта, и выбрать нужное значение.



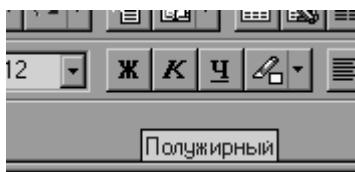
Необходимо отметить, что выбранный вами шрифт и его размер будут действительными только для того текста, который только еще будет набираться.

Для того чтобы изменить параметры уже набранного текста, нужно предварительно выбрать его так, как вы это делали, например, в графическом редакторе. То есть установить курсор мыши на начало выбираемого фрагмента, нажать на левую клавишу и, не отпуская ее, тянуть до тех пор, пока весь нужный кусок текста не поместится в выделенную область (в ней текст отображается белыми буквами на черном фоне).

Выделение части текста

Следующие четыре клавиши немного изменяют выбранный нами шрифт и позволяют выделить отдельные слова или целые абзацы.

Отметим, что все кнопки в редакторе снабжены всплывающей подсказкой, и если чуть подольше подержать мышку на кнопке, не нажимая ее, появится надпись, поясняющая смысл этой кнопки.



Вот и на нашем очередном рисунке надпись в рамочке относится к кнопке с изображением буквы “Ж”. Она означает, что текст будет выделен полужирным начертанием.

Кнопочка “К” означает выделение курсивом, а “Ч” — подчеркиванием.

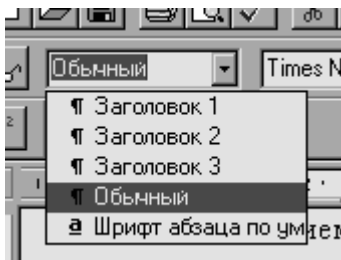
На последней кнопке нарисован маркер. Он позволяет выделить часть текста ярко-желтой подсветкой. Впрочем, цвет можно выбрать из 15 предлагаемых вариантов, если нажать кнопку, расположенную рядом с изображением маркера.

Выравнивание

Кнопки панели форматирования, изображенные на следующем рисунке, позволяют выровнять текст по левому краю, по центру, по правому краю, по ширине. В последнем случае будут выровнены оба края текста.



Стиль абзаца



Самое левое окошко панели форматирования определяет стиль различных абзацев. Оно позволяет не вспоминать каждый раз, как мы выделяем, скажем, заголовки или подзаголовки, а с ходу применить к ним какие-либо параметры абзаца, разные для каждого типа.

Количество этих стандартных типов абзацев не ограничено, и всегда можно определить новый или переопределить старый тип, заказав для него шрифт, его размеры, выравнивание и т.п.

Дополнительные возможности

Следующие четыре кнопки панели форматирования позволяют тем или иным способом выделить отдельные абзацы.



1. Первая кнопка обеспечит создание так называемого “нумерованного списка”, когда каждый следующий абзац будет нумероваться очередным числом, а весь текст в нем слегка сдвигается вправо. Так оформлен абзац, который вы читаете.

• Вторая — это создание нумерованного списка. Вместо числа абзац отмечается каким-либо символом. Например, жирной черной точкой — как читаемый вами абзац.

Четвертая кнопочка сдвигает весь абзац вправо, а третья — влево. И ту, и другую кнопку можно применить к абзацу несколько раз, увеличив величину сдвига.

И, наконец, с помощью последней кнопки можно нарисовать рамочки внешних границ. Впрочем, с обрамлением и заливкой мы познакомимся несколько позже.

Пока что ограничимся этим.

Ну а теперь приступим к набору текста стихотворения. Например, такого:

Отрывок из поэмы “Евгений Онегин” А.С. Пушкин

*Гонимы вешними лучами,
С окрестных гор уже снега
Сбежали мутными ручьями
На потопленные луга.
Улыбкой ясною природа
Сквозь сон встречает утро года;
Синея блещут небеса.
Еще прозрачные леса
Как будто пухом зеленеют.
Пчела за данью полевой
Летит из кельи восковой.
Стада шумят, и соловей
Уж пел в безмолвии ночей.*

Подумайте, какие именно возможности текстового редактора из разобранных выше вам понадобятся, чтобы добиться именно такого его вида. Возможно, вы решите оформить текст, используя и какой-нибудь оригинальный шрифт...

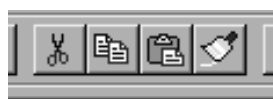
Не забудьте после работы сохранить стихотворение в папке своего класса или в своей личной папке. Для этого необходимо, как обычно, выбрать пункт главного меню **Файл**, подпункт **Сохранить как...**, после чего появится уже знакомое вам окно сохранения документа.

§ 22. Стандартная панель инструментов и некоторые дополнительные возможности редактора Microsoft Word

Из стандартной панели нам потребуются поначалу далеко не все кнопки.

Работа с буфером обмена

Уже известные вам операции работы с буфером обмена осуществляются в Microsoft Word с помощью следующих кнопок:



- Клавиша с изображением ножниц удаляет выделенный текст в буфер обмена. Как легко понять, этот самый текст исчезает с вашей страницы.
- Для того чтобы он появился снова в ТОМ МЕСТЕ, ГДЕ СТОИТ КУРСОР, необходимо нажать левой клавишей мыши кнопку с изображением папки (третья слева). Она вставляет содержимое буфера обмена в ваш документ.
- Кнопка с изображением двух листочков бумаги копирует выделенный текст в буфер обмена. При этом из документа выделенный кусок не пропадает.

Напоминаем, что буфер обмена — один на все программы, работающие в среде Windows. Поэтому в нем вполне может быть объект, полученный отнюдь не в текстовом редакторе. Например, рисунок из редактора CorelDraw! или Adobe Photoshop.

Работа с файлами

На стандартной панели находятся три кнопки, позволяющие работать с файлами.



Их смысл точно такой же, как и в программе Microsoft GIFAnimativ:

- Первая — создает новый документ. Он автоматически становится доступным для редактирования, но это вовсе не означает, что предыдущий документ теряется. Microsoft Word — многооконный редактор, и, для того чтобы переключиться между различными окнами, в которых, возможно, содержатся различные документы, достаточно указать курсором мыши на

слово “Окно” в верхнем меню и нажать левую клавишу мыши.

- Вторая кнопка вызывает специализированную программу просмотра файлов на диске, вид которой изображен на *рис. 88*.

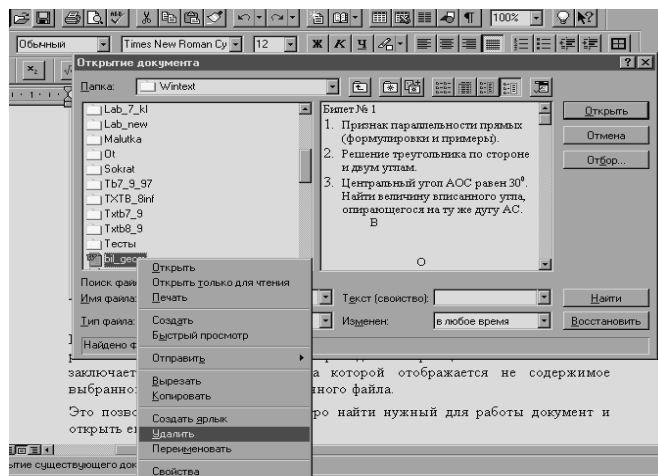


Рис. 88. Внешний вид файлового монитора редактора Microsoft Word

Работа с этим файловым монитором (то есть программой, позволяющей путешествовать по файловому дереву, отыскивать необходимые файлы, создавать папки, удалять, копировать, перемещать файлы) вряд ли представит для вас какую-либо сложность. С похожими системами вы сталкивались и в программе просмотра изображений, и в графических редакторах.

Точно так же, как и в знакомых вам программах, на одной из панелей может отображаться не содержимое выбранной папки, а текст из отмеченного файла. Это позволяет довольно-таки быстро найти нужный для работы документ и открыть его.

Если нажать на название файла левой клавишей мыши, то появится дополнительное выпадающее меню (см. *рис. 88*), позволяющее производить с файлом различные действия. В частности, удалить его или переименовать.

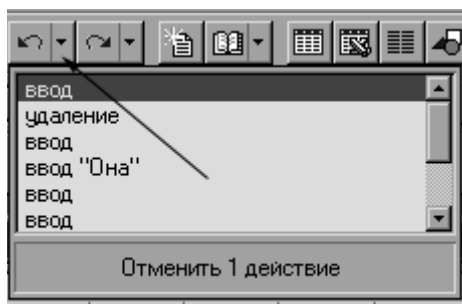
Последняя кнопочка с изображением дискетки предназначена для сохранения документа в файл на диске. Причем, если этот документ был недавно создан и еще не сохранялся на диск, редактор предложит ввести имя файла, а также определить папку, в которую необходимо его сохранить. Имя файла может быть любым, не превосходящим 256 символов, т.е. довольно-таки информативным. Например, “Реферат по биологии”. Правда, если в дальнейшем вы планируете опубликовать свой текст прямо в формате Microsoft Word в “Виртуальной школе”, реко-

мендуем придерживаться правил наименования файлов с вашими страничками: не более 8 символов, только латинские буквы и/или цифры.

Откатка

Набор из двух пар клавиш позволяет не просто отменить последнюю операцию, а предоставляет довольно обширные возможности редактирования текста, вернувшись, быть может, на несколько десятков шагов назад.

Такой пошаговый возврат обеспечивает клавиша с нарисованной искривленной стрелочкой влево (см. рисунок ниже). Более быстрый “скачок в прошлое” можно совершить, нажав небольшую кнопочку рядом (на рисунке именно на нее показывает стрелка).

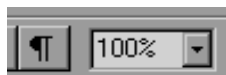


В появившемся окошке приведен список большого количества совершенных вами действий. Достаточно с помощью мышки найти описание того действия, к которому необходимо вернуться, и щелкнуть по нему указателем.

Вторая пара клавиш повторит при необходимости те действия, которые были отменены нажатием предыдущих клавиш.

Отображение непечатаемых символов

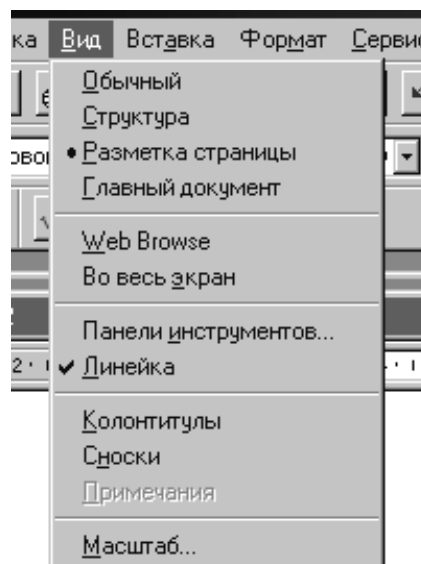
Изображенная на этом рисунке клавиша с изображением стилизованной буквы “П” делает видимыми те самые символы, о которых было рассказано в Практикуме № 3, — пробел, конец абзаца и табуляцию. Это позволяет качественно и быстро преобразовать текст, подготовленный в редакторе типа Блокнот, в формат Microsoft Word.



Рядом с кнопкой отображения невидимых символов находится масштабная шкала (см. тот же рисунок справа). С ее помощью можно увеличить или уменьшить вид страницы, сделав удобной работу с документом, который чаще всего просматривается в одном из двух режимов.

Режим просмотра документа

Нам приходится несколько отклониться от рассмотрения стандартной панели, поскольку масштабная шкала непосредственно связана с режимом просмотра документа. Он задается выбором в выпадающем меню при выборе пункта **Вид** из основного меню.



В первом приближении нас будут интересовать всего лишь два режима: “Обычный” и “Разметка страницы”. В первом очень удобно набирать и править текст, выбирая достаточное увеличение в масштабной шкале.

Второй режим позволяет увидеть то, как будет выглядеть страница, если мы вдруг решим напечатать ее на принтере.

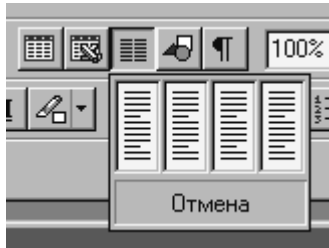
Необходимо отметить, что постраничный вид документа очень сильно зависит от того, на каком принтере вы собираетесь его печатать, что в общем-то вполне понятно. Поэтому бессмысленно добиваться идеального вида страницы в системе с одним принтером, а затем печатать его на принтере совсем другого типа.

Это относится и к переносу документов с одного компьютера на другой. Если вы планируете подготовить текст, скажем, у себя дома, а распечатать его в школе, позаботьтесь о том, чтобы ваша система поддерживала школьный принтер, даже если у вас его нет и никогда не будет.

Расположение текста в несколько колонок

В той же группе кнопок, что и кнопка включения отображения непечатаемых символов, находится кнопка разбиения текста на колонки. При нажатии на нее (см. рисунок ниже) появляется выпадающая таблица. Нажатие на первый столбик означает, что **ПРЕДВАРИТЕЛЬНО**

ВЫДЕЛЕННЫЙ ТЕКСТ будет расположен в одну колонку. Нажатие на второй столбик расположит его в две колонки и т.д.

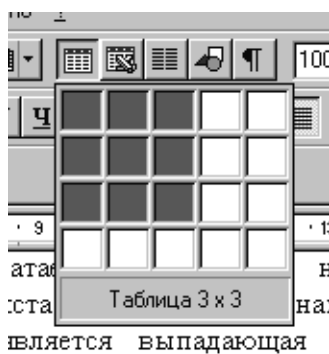


При необходимости расположить текст в пять колонок, а также задать дополнительные параметры этих колонок (таких, как их ширина и интервал между ними), необходимо выбрать в главном меню пункт **Формат**, далее выбрать подпункт **Колонки** и щелкнуть по нему указателем мыши. Эти дополнительные возможности большинству пользователей текстового редактора нужны довольно-таки редко.

Работа с таблицами. Создание таблицы

Для того чтобы вставить небольшую таблицу, вполне можно воспользоваться специальной кнопкой стандартной панели, расположенной в той же самой группе из пяти клавиш, которые мы рассматривали выше.

Достаточно нажать ее мышкой и, не отпуская клавишу, провести ее вправо и вниз, выбирая необходимое количество строк и столбцов (см. рисунок). На нашем рисунке таким образом выбрана таблица размером 3×3.



К сожалению, частенько необходимо вставить таблицу, размеры которой превосходят 4×5, предлагаемые этой кнопкой. Тогда необходимо:

- Щелкнуть мышкой на пункте главного меню **Таблица**.
- Выбрать подпункт **Вставить таблицу**, тоже щелкнув по нему мышкой.
- В появившейся рабочей панели (см. рисунок ниже) ввести необходимое количество строк и столбцов вновь создаваемой таблицы.

- Подтвердить создание таблицы нажатием на клавишу **ОК**.



Нет никакой необходимости совершенно точно подсчитывать количество строк. Их всегда можно добавить, воспользовавшись подпунктом **Вставить строки** из пункта главного меню **Таблица**. Точно так же можно и удалить лишние строки, щелкнув указателем мыши на подпункте **Удалить ячейки** (в том же пункте меню). Столбцы вставляются несколько сложнее, поэтому рекомендуется с ходу указывать верное их количество.

Форматирование таблицы

После того как таблица создана и заполнена, необходимо придать ей более привлекательный вид. Это можно сделать либо автоматически, либо вручную.

Для того чтобы отформатировать таблицу автоматически, необходимо:

- Поставить курсор (тот самый, который управляет вводом текста) на любое место таблицы).
- Щелкнуть указателем мыши на пункте главного меню **Таблица**.
- Выбрать подпункт **Автоформат** и тоже щелкнуть по нему указателем.
- В появившейся рабочей панели выбрать один из предлагаемых стандартных вариантов оформления таблицы (см. рис. 89).

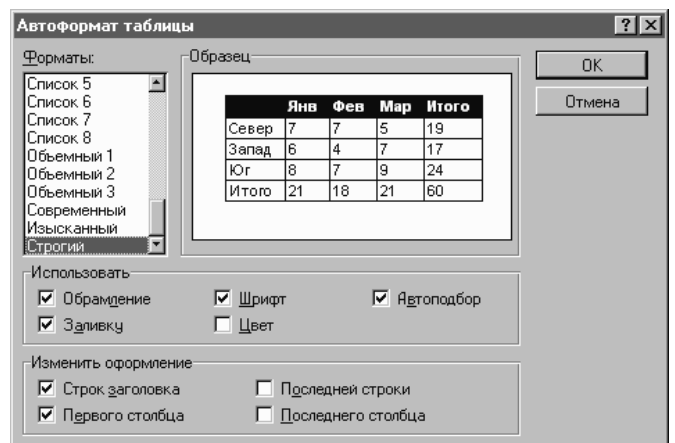


Рис. 89. Автоформатирование таблицы



- Подобрать параметры в рамочках **Использовать** и **Изменить оформление**, ставя или убирая галочки в маленьких окнах. При этом вид образца будет изменяться, и вам достаточно легко будет понять, что означает каждый из этих параметров.
- Нажать клавишу **ОК** для подтверждения преобразования созданной таблицы.

Если вдруг новый вид таблицы вам не понравился, всегда можно отменить автоформатирование, воспользовавшись кнопками откатки, рассмотренными выше.

Вставка графики в текстовый документ

В комплект текстового редактора входит собственный пакет, позволяющий рисовать различные простейшие графические объекты прямо в самом текстовом документе.

Мы, однако, рассмотрим случай, когда есть уже готовая картинка. Это можно сделать либо с помощью буфера обмена, либо вставив в документ содержимое файла, в котором сохранен рисунок.

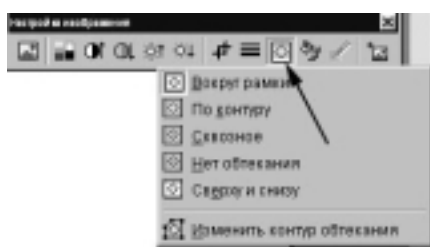
В последнем случае, разумеется, текстовый редактор должен понимать формат этого самого графического файла. Осуществляется же эта операция с помощью выбора пункта главного меню **Вставка** и затем — подпункта **Рисунок**. Далее просто необходимо с помощью файлового монитора найти необходимый файл и, предварительно просмотрев, подтвердить его выбор клавишей **Добавить**.

Рисунок в текстовом редакторе является в подавляющем большинстве случаев все же второстепенным, хотя и важным элементом. Поэтому очень важно, чтобы он естественно вписывался в текст документа.

Для этого его заключают в *кадр*. Кадрированный рисунок можно легко переместить на любое место страницы, уменьшить или увеличить его размер, а самое главное — организовать *обтекание* его текстом, как это сделано, например, с изображением панели рисования.

Для кадрирования рисунка необходимо:

- Один раз щелкнуть по нему левой клавишей мыши.
- Выбрать пункт меню **Отобразить панель настройки изображения**.
- Появится соответствующее окно, изображенное на рисунке:



- Нажмите кнопку, на которую указывает черная стрелка, щелкните мышкой по надписи **Изменить контур обтекания**, а затем
- Выберите один из вариантов **Вокруг рамки**, **По контуру** и т.д.

Не бойтесь и поэкспериментируйте, поймите, что же означают различные варианты расположения картинки относительно текста.

На этом заканчивается наше весьма краткое знакомство с одним из самых мощных текстовых редакторов Microsoft Word. Вряд ли здесь нам удалось отразить даже десятую долю его поистине обширнейших возможностей. Но и сказанного вполне хватает для того, чтобы понять и оценить такой незаменимый инструмент, каким является современный текстовый редактор.

Освоив самые необходимые его функции, вы уже способны подготовить в меру сложный документ, содержащий многоколонный текст, таблицы и графику. Надеемся, что в дальнейшем без особых сложностей вы сможете существенно расширить свои знания и умения, используя для изучения этого текстового редактора гораздо более подробные руководства.

И, уж конечно, вам скорее всего с лихвой хватит описанных возможностей, чтобы продолжить оформление своей поэтической странички, вставив в нее многоколонный текст, таблицы и графику.

Вопросы и задания

Закончите оформление своей поэтической странички, используя новые возможности, описанные в этом параграфе, и попросите учителя распечатать ваш труд на принтере.

§ 23. Публикация документов, подготовленных в Microsoft Word, в “Виртуальной школе”

Во время предыдущей практической работы вы вложили немало усилий в подготовку своей поэтической странички, подбирая лучшие стихи и красочные иллюстрации. И, разумеется, вам бы хотелось увидеть ее не только на листочке бумаги, но и как часть “Виртуальной школы”.

Существует два способа поместить документ в формате Microsoft Word в “Виртуальную школу” или Интернет. Но прежде давайте вспомним о контейнере.

Это, как вы знаете, пара тегов, придающих новые свойства объектам, расположенным внутри них. Все вы сразу же укажете эти новые свойства, например, для контейнеров `<CENTER> ... </CENTER>` и `<A ... > ... `.

Ну и, разумеется, вы помните, что

контейнеры должны располагаться строго один внутри другого, по принципу хорошо вам известной матрицы.

Самое интересное, что контейнером может быть не только пара HTML-тегов, но и несравненно более сложные объекты. Например, программа просмотра web-страничек. Под стать такому контейнеру и его содержимое. Им, например, может быть уже знакомый вам текстовый редактор Microsoft Word.

Чисто внешне это будет выглядеть так, как если бы внутрь окошка программы просмотра вставили окошко текстового редактора. При этом к пунктам главного меню программы Microsoft Internet Explorer добавятся пункты главного меню текстового редактора (см. *рис. 90*).



Рис. 90. Вид окна текстового редактора Microsoft Word в окне программы просмотра Microsoft Internet Explorer

Внутри контейнера программы просмотра может попасть и окно хорошо вам знакомого графического редактора CorelDraw! (см. *рис. 91*), и

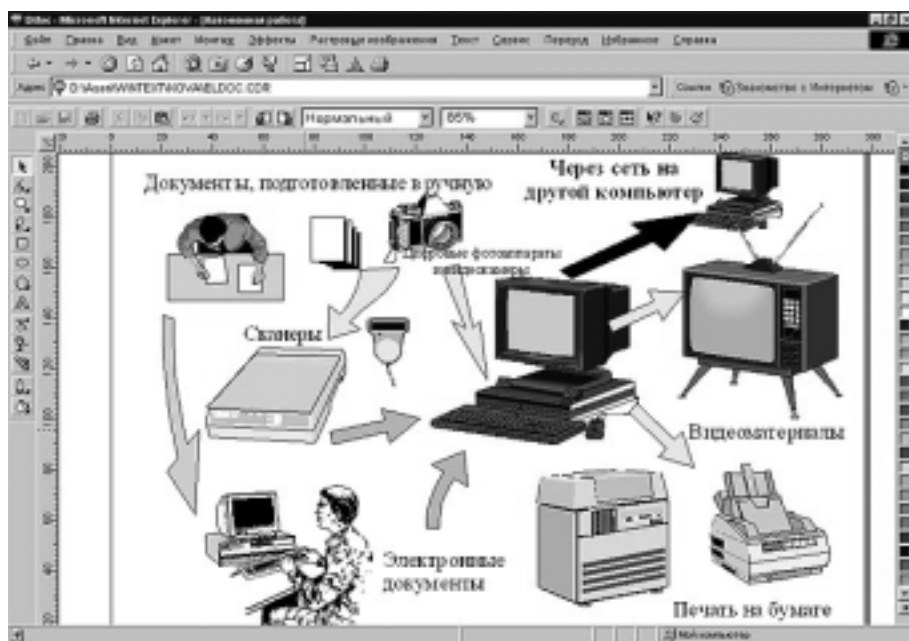


Рис. 91. Вид окна графического редактора CorelDraw! в окне программы просмотра Microsoft Internet Explorer

вообще окно любой программы, разработчики которой придерживались оговоренного стандарта. Возможно, на будущее вам пригодится его название:

для того чтобы программы могли использовать окна друг друга как контейнер, они должны поддерживать технологию ActiveX, разработанную фирмой Microsoft.

Конечно, тот же текстовый редактор, расположенный внутри контейнера другой программы, немного отличается по своим возможностям от варианта “независимого” запуска. И в первую очередь ограничена его возможность работы с файлами.

Так, в силу специфики технологии ActiveX, в которую мы не будем вдаваться, невозможно использовать тот же текстовый редактор в многооконном режиме. В случае попытки открыть второй файл в формате Microsoft Word этот текстовый редактор просто стартует, как еще одна независимая программа в своем собственном окне.

После столь обширного введения осталось только сказать, каким же образом можно воспользоваться программными контейнерами. И это — самое простое. Достаточно просто сделать ссылку не на HTML-файл, а на файл в формате Microsoft Word или CorelDraw!

Когда вы проделаете это в первый раз, обратите внимание на время, в течение которого откроется нужный вам документ. Оно гораздо больше времени открытия HTML-документа со сравнимым объемом информации, и это вполне понятно. Две довольно сложные и объемные программы, работающие одна внутри другой, как в контейнере, равносильны фактически много-

задачному запуску, когда системные ресурсы используются уже несколькими приложениями.

Иными словами, это примерно то же самое, что и работа с двумя окнами, в одном из которых, скажем, программа просмотра, а в другом — графический редактор. Вы по себе знаете, как это сказывается на быстродействии компьютера.

Тем не менее в “Виртуальной школе” довольно много документов именно в формате Microsoft Word и CorelDraw! Чем же они лучше привычных для нас HTML-файлов?

Во-первых, документы в формате Microsoft Word очень легко править. Фактически не требуется никаких особых навыков для ведения, скажем, листов успеваемости класса, информация в которых меняется еженедельно.

Во-вторых, документы в формате текстового редактора гораздо больше приспособлены к печати на принтере, чем HTML-файлы. Поэтому многие отчеты, аналитические записки, планы работы и т.п. хранятся в административном разделе “Виртуальной школы” тоже в виде файлов Microsoft Word.

В-третьих, сам по себе графический редактор CorelDraw! очень удобно использовать в качестве так называемой “электронной доски”, когда учитель свободно передвигает по экрану отдельные изображения и слова, объясняя материал. Заранее подготовленные “электронные доски” тоже необходимо сделать всеобщим достоянием, разместив ссылки на них в “Виртуальной школе”.

Конечно, такого сорта публикацию документов удобно использовать во “внутреннем Интернете”, или **Интранете**, когда информация внутри одной организации передается по высокоскоростной локальной сети. В Интернете ее обычно не используют. И связано это не только с тем, что на компьютере-получателе должна быть установлена еще и программа — содержимое контейнера, но и с довольно-таки большим размером файла с самым простеньким текстовым документом или картинкой.

Поэтому в текстовом редакторе Microsoft Word существует возможность преобразования документа непосредственно в HTML-формат.

Для того чтобы ею воспользоваться, необходимо выбрать пункт главного меню “Файл”, подпункт “Сохранить в формате HTML”. Конечно, не все элементы оформления вашего документа в формате Microsoft Word будут точно так же выглядеть и в формате HTML, но в целом и целом такое преобразование выдает удовлетворительное качество.

Подведем итоги:

- В современных программах широко используется принцип контейнера, когда какой-

либо объект помещается внутрь другого, обогащаясь свойствами внешнего объекта.

- Используя технологию ActiveX, можно создать “гибриды” текстового или графического редакторов и программы просмотра. Программа просмотра выступает в этом случае в качестве контейнера. Это позволяет включать в единую информационную среду “Виртуальной школы” не только HTML-файлы и картинки в соответствующих форматах, но и текстовые документы в формате, например, Microsoft Word или графические файлы в формате CorelDraw!
- В текстовом редакторе, кроме того, существует возможность и непосредственного перевода документа в HTML-формат.

Вопросы и задания

1. Напечатайте вашу поэтическую страничку двумя способами: используя технологию ActiveX и путем преобразования документа в HTML-формат.
2. Сравните работу программы просмотра в обоих этих случаях.
3. Оцените, насколько удачно создан HTML-документ в автоматическом режиме. Если это необходимо, подправьте его, добавив фон и другие элементы оформления.

Заключение

Уважаемые ученики седьмых классов!

В этом году вы сделали свой самый первый шаг в увлекательный мир современной информатики, освоили графические и текстовые редакторы, научились публиковать свои странички в “Виртуальной школе”.

Разумеется, все это — только ничтожная часть тех широчайших возможностей, которые предоставляет человечеству Единая Информационная Среда, называемая Интернет.

Вы наверняка уже видели, что электронные документы, в отличие от обычных, могут вести активный “диалог” с читателем, откликаясь на движения мышкой и нажатие различных кнопок на клавиатуре.

Это их свойство называется **интерактивностью**, и именно оно лежит в основе тех широчайших возможностей Сети, о которых мы говорили в § 8.

В этом году мы как раз и научимся создавать интерактивные web-странички, что позволит нам не только украсить свои странички в “Виртуальной школе”, но и глубже понять основные принципы работы компьютеров.

Надеемся, после летних каникул ваш интерес к информатике вспыхнет с новой силой, поэтому желаем вам как можно лучше отдохнуть.



Леонид Витальевич Канторович

(К 70-летию со дня рождения)

А.Н. Колмогоров, В.А. Залгаллер

Наука стала одной из производящих сил общества, научные исследования ведутся сейчас широко, в них участвуют тысячи специалистов. Особенно возрастает при этом роль ведущих ученых, обогащающих науку фундаментальными результатами и новыми направлениями. К их числу принадлежит академик Леонид Витальевич Канторович. Выдающийся математик и экономист, обогативший и математику, и экономическую науку результатами первостепенного значения, он является также одним из видных организаторов советской науки.

Л.В. Канторович родился 19 января 1912 г. в семье врача. Самостоятельно и очень рано Леонид Витальевич занялся математикой. В 14 лет он становится студентом Ленинградского университета, который оканчивает в 18 лет. Уже студентом второго-третьего курсов, активно участвуя в научных семинарах, он выполняет свои первые научные работы по дескриптивной теории множеств, привлекавшей внимание многих математиков. Затем он обращается к функциональному анализу, молодому тогда разделу математики. Здесь Л.В. Канторовичу принадлежат многие результаты. В их числе — создание теории полупорядоченных пространств, названных в его честь K -пространствами. С тех пор эти разделы математики заметно выросли. Но и сейчас в них существенно используются результаты Канторовича, а функциональный анализ и теорию K -пространств многие изучают по изданным на нескольких языках книгам, написанным им совместно с его учениками.

Умение постоянно ощущать обращенные к науке запросы окружающей жизни и глубоко понимать внутренние связи самой математики создали то опережающее проникновение в перспективы науки, которое позволяет Леониду Витальевичу выбирать самые развивающиеся разделы для приложения своих сил и столь успешно направлять усилия других специалистов.

От функционального анализа Л.В. Канторович переходит к вычислительной математике. В то время еще не было ни ЭВМ, ни вычислительных центров; методические возможности вычислительной математики также были скромными. Леонид Витальевич заметно уси-

лил их, причем как раз к тому моменту, когда появились ЭВМ и вычислительные центры. В своих работах, развивавших связь функционального анализа и прикладной математики, Леонид Витальевич исходил из убеждений, что разумное обобщение позволяет яснее увидеть суть дела и нередко получать более точный результат, чем при индивидуальном рассмотрении задачи, и что наличие хороших приближенных методов способно помочь устанавливать существование и свойства точных решений. За научные работы этого направления Л.В. Канторович удостоен Государственной премии 1949 г. Многие вычислители и сейчас учатся по книге “Приближенные методы анализа”, написанной им совместно с В.И. Крыловым.

Наряду с приближенными методами Леонид Витальевич занимался и непосредственной организацией вычислительных работ. При этом он творчески использовал еще беспрограммные релейные машины, изобрел некоторые усовершенствования для них, затем перешел к работам на первых ЭВМ, руководил разработками в области программирования, постановки на ЭВМ аналитических вычислений, микропрограммирования.

К экономическим проблемам Л.В. Канторовича привлекли также совершенно конкретные вопросы. Студентом он ездил на практику как экономист-статистик; как сотрудник Математического института при Ленинградском университете встречался с обращавшимися за консультациями практическими работниками: с

одним он беседовал о загрузке станков, с другим — о практике лесопиления, с третьим — о маршрутах транспортных перевозок. Математическое обобщение, осознание единства целого круга задач, не находивших должных методов решения в арсенале классической математики, привели Леонида Витальевича к созданию большого нового направления в математике и экономике. Это направление получило позже название линейного программирования.

Речь идет о поиске наилучшего решения в ситуациях, когда на конечное число переменных накладывается целый ряд линейных ограничений, а переменным надо придать значения, которые, не нарушая этих ограничений, придавали бы наибольшее значение некоторому показателю (тоже линейно зависящему от этих



С любезного разрешения составителей мы продолжаем публикацию материалов из книги “Очерки истории информатики в России”

переменных). Леонид Витальевич в 1938 г. выделил этот класс задач и предложил для них путь решения. В 1939 г. в издательстве Ленинградского университета вышла написанная Л.В. Канторовичем брошюра “Математические методы организации и планирования производства”, ставшая классической работой нового научного направления. Леонид Витальевич раскрыл широту применимости этих методов: выбор вариантов загрузки станков, рациональный раскрой материалов, оптимальные кормовые смеси, маршрутизация перевозок, множество других применений. Эта работа была оценена не сразу — началась Великая Отечественная война. С опозданием переоткрывались за рубежом результаты этого направления.

В 1959 г. в книге “Экономический расчет наилучшего использования ресурсов” Леонид Витальевич с максимальной полнотой раскрывает экономическую значимость этих методов. Здесь он выступает уже не только как математик, а одновременно как ученый-экономист, глубоко проникающий в природу экономических зависимостей, способный дать их анализ, ведущий к реальным, практическим и эффективным результатам. Он анализирует широкие вопросы выбора вариантов технологических процессов, промышленных перевозок, размещения посевов, обновления оборудования, эффективности вложений, формирования оптовых цен, общей структуры экономических показателей, способных служить эффективной основой хозяйственного расчета и ориентировать на принятие решений в народно-хозяйственных интересах.

Эти работы Леонида Витальевича получили общесоюзное, а затем и мировое признание. В 1965 г. Л.В. Канторович удостоен Ленинской премии (совместно с В.С. Немчиновым и В.В. Новожиловым). В 1975 г. ему (совместно с американским математиком Т.Купмансом) присуждена Нобелевская премия по экономике.

Сейчас линейное программирование изучают на всех экономических и математических факультетах, о нем сообщается в школьных учебниках. Каждый вычислительный центр включает эти методы в свое программное обеспечение, которое постоянно совершенствуется. Экономический анализ немислим без их применения. Расчетной обработке подвергаются огромные массивы данных. Так, в расчетах по оптимизации загрузки прокатных станов страны, проводившихся под руководством Леонида Витальевича, фигурировало более миллиона переменных.

Л.В. Канторович — профессор с 1934 г., доктор физико-математических наук с 1935 г., член-корреспондент АН СССР с 1958 г., академик с 1964 г.

В военные годы Леонид Витальевич в кадрах флота — он начальник кафедры, преподает в блокадном Ленинграде; в 50-е годы заведует отделом приближенных вычислений в Математическом институте им. В.А. Стеклова АН СССР; в 60-е годы участвует в создании академического центра в Новосибирске, является руководителем отдела и заместителем директора Института математики Сибирского отделения АН СССР; в 70-е годы — профессор Института управления народным хозяйством в Москве, участвует в переподготовке кадров и в работе комиссий АН СССР, обсуждающих направление и этапы совершенствования планирования и управления эко-



Москва, 25 апреля 1978 г. Л.В. Канторович выступает в клубе “Эврика” на встрече, посвященной А.А. Ляпунову

номикой. Сейчас Леонид Витальевич — член Государственного комитета Совета Министров СССР по науке и технике, заведующий отделом и руководитель научного направления во Всесоюзном научно-исследовательском институте системных исследований.

В своей деятельности Л.В. Канторович уделяет большое внимание вопросам высшего и среднего математического образования, является членом редакционного совета журнала “Квант”.

Заслуги Леонида Витальевича, кроме высоких научных премий и званий, отмечены двумя орденами Ленина, тремя орденами Трудового Красного Знамени, орденом “Знак Почета” и медалями.

Многие отечественные и иностранные научные общества избрали его своим почетным членом. Леонид Витальевич Канторович является иностранным членом Академии наук Венгрии, Югославии, Академии наук и искусств США (Бостон), членом-корреспондентом Академии наук Мексики, почетным доктором университетов Глазго, Варшавы, Гренобля, Ниццы, Мюнхена, Хельсинки, Парижа (Сорбонна), Кембриджа, Пенсильвании, Статистического института в Калькутте, входит в редакционные коллегии ряда советских и зарубежных журналов.

**Государственный координационный центр
информационных технологий
Минобразования России совместно с Московской
финансово-юридической академией**

приглашает на курсы повышения квалификации
и профессиональной переподготовки

**• Преподавателей и специалистов по информатике
Гос. удостоверение/гос. диплом**

По специальностям:

- «Менеджмент в образовательных учреждениях»
- «Информационные технологии в образовании»

**Участие в системе грантов на получение
вычислительной техники**

Обучение, проживание и питание в одном здании.

Недельные курсы проводятся в течение всего учебного года

Тел. (095) 127-26-53

Факс (095) 123-15-00

Адрес: 113447, Москва, ул. Большая Черемушкинская, д. 17А

E-mail: post@rui.ru

МИНИСТЕРСТВО ОБЩЕГО И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ РОССИИ,
ИНСТИТУТ ЮНЕСКО ПО ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ В ОБРАЗОВАНИИ,
ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ РОССИЙСКОЙ АКАДЕМИИ НАУК,
МОСКОВСКИЙ КОМИТЕТ ОБРАЗОВАНИЯ,
НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ "БИТ ПРО" И
ГАЗЕТА "ИНФОРМАТИКА"

приглашают принять участие в IX международной конференции
"ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ"
("ИТО-99")

*Конференция проводится с 9 по 12 ноября 1999 года
в Московском городском физико-математическом лицее № 1511 при МИФИ.*

Работа конференции будет проходить по следующим секциям:

I. Информатика: стандарты и содержание

- концепции, базовый курс и стандарты
- образовательный минимум
- опыт и методика преподавания
- подготовка и повышение квалификации преподавателей

II. Интеграция информационных технологий в образование

- естественно-математические предметы
- гуманитарные предметы
- изучение языков
- профессиональное образование
- медиатеки, виртуальные музеи
- технология разработки, экспертизы и оценки программных средств

III. Технологии открытого образования

- телекоммуникации
- дистанционное образование
- виртуальная школа и университет

IV. Информационные технологии в управлении образовательными структурами

- управление образовательными заведениями
- единое информационное образовательное пространство

V. Информационные технологии в образовании для людей со специальными потребностями

- создание информационно-коммуникационной среды для социальной адаптации и реабилитации инвалидов
- виртуальный мир для учебы и общения
- специальные программно-аппаратные средства

В программу проведения конференции войдут **"круглые столы"** и **мастерские** (workshop). Одновременно пройдет **выставка**, собирающая десятки российских производителей учебных компьютерных программ и издателей учебно-методической литературы.

Тезисы и регистрационные формы необходимо выслать **до 15 сентября**. Информацию об оплате, требования к оформлению тезисов, а также бланк регистрационной формы можно получить в оргкомитете или на сервере <http://ИТО.ВITpro.ru>.

Спонсоры — издательства **"Лаборатория Базовых Знаний"** (тел.: (095) 973-90-64; e-mail: lbz@aha.ru) и **"Открытые системы"** (тел.: (095) 253-90-20; e-mail: giglavy@lit.msu.ru). Проекционное оборудование предоставляется компаниями **"СМИстар-Холдинг"** (тел.: (095) 235-85-76, 235-87-26; e-mail: smistar@aha.ru) и **Activision** (тел.: (095) 212-25-39, 212-62-27, 969-74-00; e-mail: post@activi.dol.ru).

Ждем Вас на самой крупной конференции в СНГ по проблемам информатизации образования!

Координаты оргкомитета:

Телефоны: (095) 324-55-86, 324-97-69.

Факс: (095) 324-55-86.

Адрес: 115522, Москва, Пролетарский проспект, д. 6, корп. 3. НПП "БИТ про".

E-mail: Office@BITpro.ru, <http://www.BITpro.ru>

Выдающийся проект знаменитого англичанина

Из всех изобретателей прошлых столетий, внесших вклад в развитие вычислительной техники, наиболее близко к созданию компьютера в современном представлении подошел англичанин Чарльз Бэббидж [1].

Желание механизировать вычисления возникло у Бэббиджа в связи с недовольством, которое он испытывал, сталкиваясь с ошибками в математических таблицах, используемых в самых различных областях.

В 1822 г. Бэббидж построил пробную модель вычислительного устройства, назвав ее "Разностной машиной": работа модели основывалась на принципе, известном в математике как "метод конечных разностей". Данный метод позволяет вычислять значения многочленов, употребляя только операцию сложения, и не выполнять умножение и деление, которые значительно труднее поддаются автоматизации. При этом предусматривалось применение десятичной системы счисления (а не двоичной, как в современных компьютерах).

Однако Разностная машина имела довольно ограниченные возможности. Репутация Бэббиджа как первооткрывателя в области автоматических вычислений завоевана в основном благодаря другому, более совершенному устройству — Аналитической машине (к идее создания которой он пришел в 1834 г.), имеющей удивительно много общего с современными компьютерами.

Предполагалось, что это будет вычислительная машина для решения широкого круга задач, способная выполнять основные операции: сложение, вычитание, умножение, деление. Предусматривалось наличие в машине "склада" и "мельницы" (в современных компьютерах им соответствуют память и процессор). Причем планировалось, что работать она будет по программе, задаваемой с помощью перфокарт, а результаты можно будет выдавать на печать (и даже представлять их в графическом виде) или на перфокарты. Интересно, что в процессе работы над проектом Аналитической машины Бэббидж нашел подходы к созданию значительно менее громоздкого устройства — Разностной машины № 2.

Заслуги Чарльза Бэббиджа и его ученицы и помощницы Ады Лавлейс (графиня Лавлейс, урожденная Августа Ада Байрон, дочь поэта лорда Байрона), которую называ-

ют первой программисткой, трудно переоценить. В чем же суть их достижений?

Во-первых, это идея программного управления процессом вычислений. Во-вторых, предложение использовать перфокарты для ввода и вывода данных и для управления, а также для обмена и передачи чисел в самой машине. В-третьих, изобретение системы предварительного переноса для ускорения расчетов. В-четвертых, применение способа изменения хода вычислений, получившего в дальнейшем название условного перехода. В-пятых, введение понятия циклов операций и рабочих ячеек [2].

В материалах Бэббиджа и комментариях Лавлейс намечены такие понятия, как подпрограмма и библиотека подпрограмм, модификация команд и индексный регистр, которые стали употребляться только в 50-х годах нашего века. Сам термин *библиотека* был впервые введен Бэббиджем, а термины *рабочая ячейка* и *цикл* предложила Ада Лавлейс [3, 4].

В 1985 г. сотрудники Музея науки в Лондоне решили выяснить наконец, возможно ли на самом деле построить вычислительную машину Бэббиджа. После нескольких лет напряженной работы старания увенчались успехом. В ноябре 1991 г., незадолго до двухсотлетия со дня рождения знаменитого изобретателя, Разностная машина № 2 впервые произвела серьезные вычисления [1].

ЛИТЕРАТУРА

1. *Дорон Свейд*. Механический компьютер Чарльза Бэббиджа: возвращение через полтора столетия: Пер. с англ. // Информатика, № 6/96.
2. *Частиков А.П.* От калькулятора до суперЭВМ. // Вычислительная техника и ее применение, № 1/88.
3. *Гутер Р.С., Полунов Ю.Л.* Августа Ада Лавлейс и возникновение программирования. Кибернетика и логика. М.: Наука, 1978.
4. *Бородин А.П., Галло В.Ф.* Августа Ада Лавлейс — первый программист. // Информатика, № 11/95.

<p>Гл. редактор С.Л.Островский Зам. гл. редактора Е.Б.Докшицкая Редакция: Н.Л.Беленькая, Н.П.Медведева Дизайн и компьютерная верстка: Н.И.Пронская Корректоры: Е.Л.Володина, С.М.Подберезина</p>	<p>©ИНФОРМАТИКА 1999 выходит четыре раза в месяц При перепечатке ссылка на ИНФОРМАТИКУ обязательна, рукописи не возвращаются</p>	<p>121165, Киевская, 24 тел. 249 4896 Отдел рекламы тел. 249 9870</p>	<p>Учредитель: ООО "Чистые пруды" Регистрационный номер 012868 Отпечатано в типографии ОАО ПО "Пресса-1". 125865, ГСП, Москва, ул. Правды, 24. Тираж 5000 экз. Заказ №</p> <p>Internet: inf@1september.ru Fidonet: 2:5020/69.32 WWW: http://www.1september.ru</p>
<p>ИНДЕКС ПОДПИСКИ для индивидуальных подписчиков 32291 комплекта приложений 32744</p>			
<p>Тел. (095)249 3138, 249 3386. Факс (095)249 3184</p>			