ЛЕКЦИИ 7

Тема. КОМПЬЮТЕРНОЕ ГРАФИЧЕСКОЕ МОДЕЛИРОВАНИЕ. МОДЕЛИ ДЕЛОВОЙ ГРАФИКИ

Модели художественной графики

Одним из основных достоинств компьютерного моделирования как метода познания является возможность исследователя представлять результаты моделирования в графическом виде, что позволяет сделать даже невидимые и абстрактные объекты «видимыми», например, увидеть строение молекулы, распад атомного ядра, лицо умершего человека, смоделированное по черепу, фоторобот, смоделированный по словесному описанию. Довольно часто программы, моделирующие объекты, процессы, явления в виде графических изображений, пишутся на основе их математической модели,но не всегда. Графическую модель можно построить вообще без использования математики, например, с помощью точек, отрезков прямых и кривых линий, плоских и объемных фигур и т.п. Получаемые модели являются красочными и наглядными, поэтомуэту главу начнем с рассмотрения графических моделей. В компьютерной графике различают художественную, деловую и научную графику, различие это чисто условное, в разных источниках оно определяется по-разному.

Графика, связанная с использованием художественных образов, называется **художественной**.

Одним из направлений художественной графики является моделирование различных узоров, при этом основная роль принадлежит зрительному восприятию результатов моделирования. Например, при пересечении двух семейств прямых под заданным углом получаются узоры, которые называются «муаровыми». Меняя параметры семейства линий, можно составить очень красивые «муаровые» узоры (Рис.32).

Моделирование «Муарового узора»

Для моделирования такого узора можно использовать, например, такую программу, составленную на языке программирования Паскаль.

```
program  Y3OR;
    uses Crt, Graph;
    var gD, gM, i, k, k1, k2, m, n : integer;
BEGIN
    gD:=Detect;  InitGraph(gD,gM,  '          ');
    randomize;  setbkcolor(15); k:=
    random(15)+1;  setcolor(k);
    k1:= 2; k2:= 4; {параметры узора}if
    k1 <= k2 then m :=k1 else m:=k2;
    n:=480 div m;
    for i:=1 to n do begin line(0, i*k1, 640, i*k2); line(0, i*k2, 640, i*k1); end;readkey;
    closegraph;
END.
```

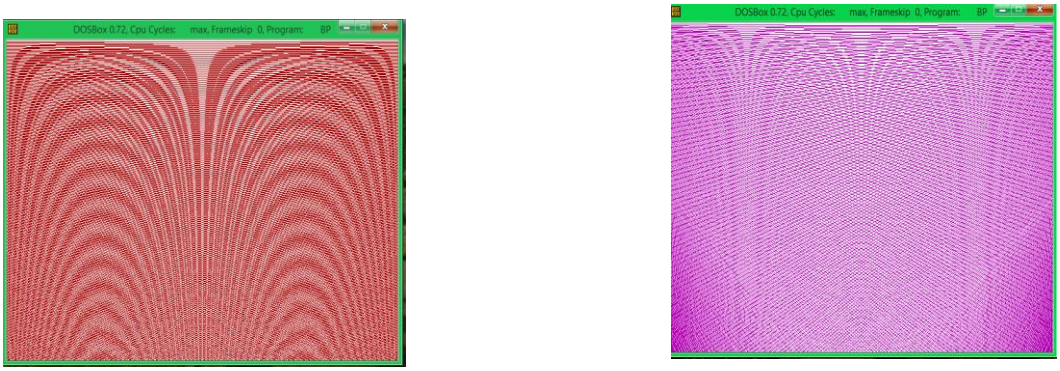


Рис. 32 - Графические модели – «муаровые узоры»

На Рис.32 представлены узоры, которые моделирует эта программа. При пересечении двух семейств прямых, которые рисуются операторами ***line*** с разными параметрами**,** наблюдается эффект «интерференции», воспринимаемый зрительно как узор.

Меняя параметры модели k1, k2, задающие расстояние между линиями соответственно в левом и правом семействе прямых, то есть, выполняя с моделью простейший компьютерный эксперимент, можно получить различные «муаровые» узоры. Например, при k1 =2, k2 =4, получится узор, представленный на Рис. 32 слева, при k1
=3, k2 =7, получится узор, представленный на Рис. 32 справа. Очевидно, узоры существенно отличаются друг от друга, цвет узора в программе задается случайным образом из диапазона [1-16] с помощью функции **random().** В данной графической модели «художественным образом» является отрезок прямой.

Моделирование узора – «Звезда»

Узоры можно рисовать и с помощью таких «художественных образов», как окружность, эллипс, ромб, угол. Рассмотрим, как можно использовать фигуру угол для рисования узора в виде звезды. Угол будем задавать координатами трех точек, (x1,y1), (x2,y2), (x3,y3) расположенных на двух окружностях разных радиусов, имеющих общий центр. При этом координаты точек будем рассчитывать по формулам (1), которые задают точки окружности с параметрами - радиусом r и углом α.

$$\begin{cases} x = x_0 + r \cdot \cos(\alpha) \\ y = y_0 + r \cdot \sin(\alpha) \end{cases} \qquad (1)$$

На Рис. 33 ниже фактически изображена графическая модель угла, на которой точка (x1,y1) изображена на окружности с радиусом r1, а точки (x2,y2) и (x3,y3) - на окружности с радиусом r2. Для точки (x2,y2) показан угол α3., который образует радиус-вектор этой точки с осью OX.

\

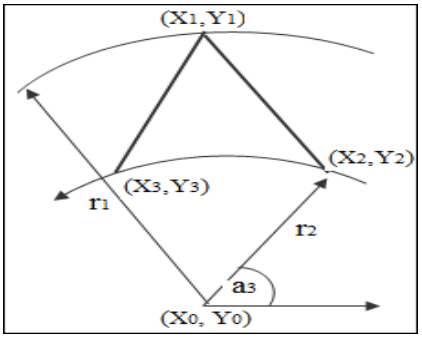


Рис. 33- Модель угла

Формула (1) является математической моделью точки с координатами (x,y), лежащей на окружности радиуса r с центром в точке (x0, y0), радиус вектор этой точки образует с горизонтальной осью угол α. Эта формула размещена в процедуре Ugol в программе Star, моделирующей узор – звезду. В данном случае, модель Star является уже компьютерной математической моделью фигуры звезда.

```
program  Star;
    uses Crt, Graph;
    var gD, gM, i, k, x0, y0, n: integer; r1, r2, a1, a2, a3: real;
```

```
procedure Ugol; { процедура, рисующая угол}
    var x1, y1, x2, y2, x3, y3 : integer;begin
        x1:= round (x0 + r1*cos (a1)); y1:= round (y0 - r1*sin (a1));x2:=
        round (x0 + r2*cos (a2)); y2:= round (y0 - r2*sin (a2));x3:= round
        (x0 + r2*cos (a3)); y3:= round (y0 - r2*sin (a3));line (x1, y1,x2, y2);
        line (x1, y1, x3, y3);
end;
BEGIN

    gD:=Detect;  InitGraph (gD, gM, '              ');
    randomize;
    setbkcolor(15);
    k:= random(15) +1;
    setcolor(k);
    x0:= 280; y0:= 250; {координаты центра окружностей}
    r1:= 180;      r2:= 60; {радиусы внешней и внутренней окружности} a1:
    = 0;  {начальный угол}
    n: = 5; {число лучей звезды}
    for i: =1  to n do
                        begin
                            a1:= a1+2*pi/n;  a2: =a1+pi/n;        a3: =a1-pi/n;
                            Ugol;
                            reakey; {нажмите любую клавишу}
                        end;
    setfilstyle (1,k);
```

Программа Star моделирует узор в виде звезды. Меняя в программе количество лучейзвезды n, радиусы окружностей r1 и r2, можно получить различные конфигурации звезд,изображенные на Рис.34

Рис. 34 - Графические модели – «звезды»

Моделирование узора – «Дерево»

Рассмотрим, как можно использовать эту же фигуру угол при моделировании другого узора, назовем его «Деревом».

Программа Derevo, моделирующая этот узор, представлена ниже. Она содержит ряд взаимно обратных процедур: Vpered (рисование линии от точки (x1,y1) до точки (x2,y2)) и Nazad (рисование линии от точки (x2,y2) до точки (x1,y1) – фактически возврат в точку (x1,y1)), PovVlevo (поворот влево на указанное число градусов) и PovVpravo (поворот вправо на указанное число градусов). В основе процедуры Vpered лежитматематическая модель (1)

```
program Derevo;
    uses Crt, Graph;
    var gM, gD, i : integer; x0, y0, a : real;procedure
Vpered (r: real);
    var x1, y1, x2, y2 : integer;
 begin
        x1:= round(x0); y1:= round (y0);
        x0:=x0 + r*sin(a); y0:=y0 - r*cos(a)
        x2:= round (x0); y2:= round (y0) ;
        line(x1, y1, x2, y2); delay(100);
end;
```

```
procedure  Nazad ( r: real); begin
                    Vpered ( -r);   end;
procedure  PovVpravo ( g: real);
       begin      a := a + g*PI / 180;       end;
procedure PovVlevo ( g: real);
        begin      PovVpravo ( - g);        end;
procedure  Vetka ( d: real;  t: integer);
       begin

          if t = 0 then exit ;
          PovVlevo (45); Vpered (d); Vetka (d/2, t – 1);
          Nazad (d);  PovVpravo (90);  Vpered (d); Vetka
          (d/2, t – 1);  Nazad (d);  PovVlevo (45);
       end;
BEGIN
       gD:=Detect;  InitGraph(gD, gM,  '                ');
       x0:= 320; y0:= 240;  {координаты центра ствола «дерева} a:=0;
       Vetka (100,10);
       readkey;         closegraph;
END.
```

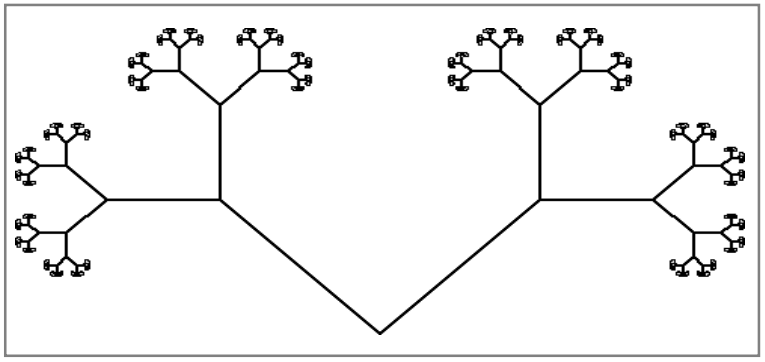


Рис.35 - Графическая модель «Дерево»

Основная процедура программы - Vetka, она рисует угол из двух веток, процедураимеет два параметра: d – длина ветки в начальном угле и t – количество углов - веток на дереве, эта процедура содержит в своем теле ряд других процедур и обращается сама к себе, то есть, является рекурсивной. При каждом обращении процедуры к себе длина стороны ветки, которою она рисует, уменьшается в два раза (d/2), а количество рисуемых углов уменьшается на единицу (t–1). Процедура заканчивает работу, когда t=0. На Рис. 35 представлен узор «Дерево», которое моделирует приведенная выше программа при d=100 и t=10.

Модели деловой графики

Деловой или иллюстративной называется графика, предназначенная для представления процессов, явлений, функционирования систем в виде различныхдиаграмм и графиков.

Эти диаграммы и графики являются, по сути, графическими моделями этих объектов**.** Если модели получены с помощью компьютера, то это будут компьютерные модели. В настоящее время существует множество программных средств, позволяющих строить такие модели — это MatCad, AutoCad, Excel, 3DMAX, различные среды программирования. В данном пособии в основном приведено построение иллюстративных моделей в интегрированных системах программирования, использующих язык Турбо Паскаль.

Моделирование динамики изменения числа студентов

Приведенная ниже программа Students моделирует процесс изменения числа студентов на физико-математических профилях Школы педагогики ДВФУ в течение пяти лет в виде ***модели - гистограммы***, представленной на Рис.35. Гистограмма имеет две оси с центром в точке (60,400), на горизонтальной оси отображаются учебные годы,на вертикальной - количество обучающихся

студентов, причем, отрезок длиной в 40 пикселей соответствует 10 студентам. Столбцы гистограммы рисуются с помощью процедуры ***bar***. Программа моделирует процесс изменения числа студентов, используя только входные данные из массивов, она не базируется на математической модели, хотя содержит математические вычисления для расчета размеров столбцов.

```
program  Students;
    uses  Crt, Graph;
    const  kol : array [1..5] of integer = (212, 198, 175, 140, 161);
             god: array[1..5] of string = ('2015/16', '2016/17', '2017/18', '2018/19', '2019/20');
    var  gM, gD, k, x0m, y0m, by, ay, ax : integer;  ky: real;   ks, kols : string [5];
BEGIN
     gD:= Detect;  InitGraph(gD, gM, '          ');
     setbkcolor(15); setcolor(1); setlinestyle (0,0,1);
     x0m:=60; y0m:=400;
     line(x0m, y0m, 600, y0m); line(x0m, 40, x0m, y0m);by:=40;
     ay:= 25; ky: = by/ay;
     for k :=1 to 8 do
          begin
               line(x0m - 3, y0m - k*by, x0m + 3, y0m - k*by);
               str (k*25,ks);  outtextxy(25 , y0m - k*by - 4, ks);
          end;
      settextstyle (0,0,2);  outtextxy(50,20, 'Динамика изменения числа студентов');
      settextstyle (0,0,1);  ax := 100;
      for k :=1 to 5 do
           begin
               setfillstyle(1, k);       str (kol[k], kols);
               bar(70 + (k - 1)*ax, y0m, 70 + k*ax, round (y0m-kol[k]*ky));outtextxy(110
              +(k-1)*ax, round(y0m -10 - kol[k]*ky), kols); outtextxy(90 + (k-1)*ax, y0m +
              10 , god[k]);
           end;
     readkey;
     closegraph;
END.
```

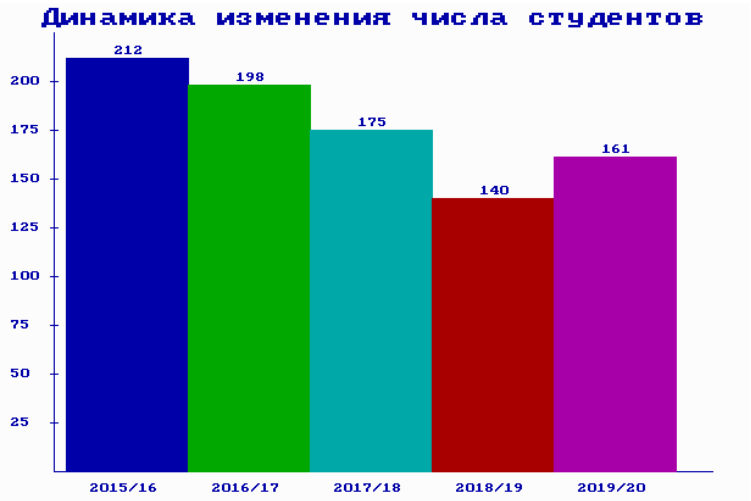


Рис. 36 - Модель – гистограмма

Моделирование распределения учащихся по секциям

Программа Sport, приведенная ниже, строит модель распределения учащихся спортивной школы по пяти секциям в виде ***круговой диаграммы,*** результат моделирования представлен на

Рис.36. В программе в цикле для каждой секции с помощью процедуры ***pieslice*** рисуется сектор круга с центральным углом, величина которого рассчитывается в теле цикла. С помощью процедуры ***bar*** рисуется прямоугольник того же цвета, что и сектор, справа от него выводится название секции, в самом прямоугольнике выводится число учащихся. Такое пояснение к аналогичной диаграмме в программе Excel называется Легендой. Эта модель является статической, таккак отображает процесс распределения учащихся в какой-то фиксированный момент времени, в ней для расчета координат точек (x, y), лежащих на окружности, использует математическая модель (1).

```
program  Sport;
      uses  Crt, Graph;
      const kol : array [1..5] of integer = (70, 120, 230, 150, 172);sek:
               array [1..5] of string  =('футбол', 'гимнастика',
                                                  'теннис','хоккей','самбо');
      var gM, gD, a, a1,a2, k, x0m, y0m, x, y, S, R1,R2 : integer;  d: real;  st, kols : string ;
BEGIN
       gD:= Detect;  InitGraph(gD, gM, '          ');
       setbkcolor(15); setcolor(1); settextstyle (0,0,2); outtextxy(60,
       70, ' Распределение учащихся по секциям');settextstyle (0,0,1);
       setlinestyle (0,0,1);
       a :=0;  S:=0;  x0:=170; y0:= 240; R1:=120; R2:=90;
       for  k := 1 to  5 do S:=S + kol[k];
       for k := 1 to 5 do
           begin
                setfillstyle (1, k+1);
                d:= kol[k] /S;
                 a1:= a + round (d*360);
                 pieslice(x0, y0, a, a1, R1); a2
                 := a + round (d*360/2);
                 x := x0m + round (R2*cos (a2*pi / 180));y :=
                 y0m - round (R2*sin (a2*pi / 180));
                 str(d*100 :2 : 0, kols);  outtextxy(x, y, kols + ' % ');
                 bar( 350, 80 + k*50, 400, 80 + 50*(k+1));
                 str(kol[k], kols);
                  outtextxy(365, 95 + k*50, kols);
                 outtextxy(410, 95 + k*50, sek[k]);
                 readkey;
                   a := a1;
           end;
        readkey;
        closegraph;
END.
```

Рис. 37 - Модель – круговая диаграмма

Построение моделей - графиков

Очень часто в деловой графике результаты моделирования строятся в виде графиков, демонстрирующих зависимость величины от какой-то величины. Рассмотрим программу, которая моделирует зависимость между двумя какими-то абстрактными величинами X и У в виде графика в декартовой системе координат, построенной на экране компьютера. Так как начало координат в построенной на экране и машинной системе координат обычно не совпадают, то в программе нужно преобразовывать координаты точек из построенной системы ХОУ в компьютерную (машинную) систему координат ХОУm. Рассмотрим, как можно сделать такое преобразование. Пусть:

(x,y) – координаты точки в системе координат ХОУ;

(x_m, y_m) – координаты этой точки в системе координат ХОУm;

($x0_m$, $y0_m$) – машинные координаты начала координат ХОУ;

b_X, b_Y– количество пикселей в одном делении на осях ОХ и ОУ; a_X,

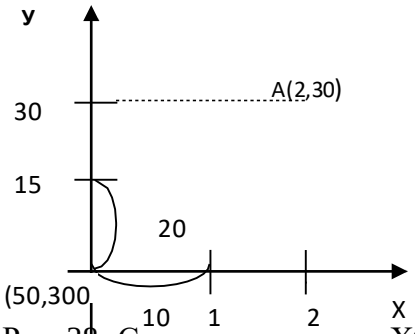
a_Y - цена деления на осях ОХ и ОУ соответственно;

$k_X = b_X/a_X$, $k_Y = b_Y/a_Y$ - коэффициенты перевода,

тогда формулы перевода координат из построенной декартовой системы координат ХОУ в компьютерную (машинную) систему имеют вид:

$$x_m = x0_m + x \cdot k_X, \quad k_X = b_X / a_X$$
$$y_m = y0_m - y \cdot k_Y, \quad k_Y = b_Y / a_X$$

Поясним такое преобразование на конкретном примере. Пусть на экране (Рис.38) построена система координат ХОУ с центром в точке экрана (50, 300), то есть, $x0_m$ = 50 пикселей, $y0_m$ = 300 пикселей – это машинные координаты центра системы ХОУ. Точка $A_в$ этой системе, как следует из Рис.37, имеет координаты (2,30).



b_X = 10 (количество пикселей в 1-ом делении оси ОХ); a_X = 1 (цена деления на оси ОХ)

b_Y = 20 (количество пикселей в 1-ом делении оси ОУ) a_Y = 15 (цена деления на оси ОУ)

k_X = 10/1 = 10 (коэффициент перевода по оси ОХ) k_Y = 20/15 = 4/3 (коэффициент перевода по оси ОУ.)

Теперь воспользуемся формулами, указанными выше.

Рис. 38- Система координат ХОУ

$$\begin{cases} X_m = 50 + 2 \cdot 10 = 50 + 20 = 70; \\ У_m = 300 - 30 \cdot 4 / 3 = 300 - 40 = 260; \end{cases}$$

Итак, машинные координаты точки А на экране - (70,260).

Ниже приведен текст программы, моделирующей зависимость между двумя произвольными величинами X и У в виде графика функции $y = x^2$ на заданном промежутке [a,b].

```
program Graphic;
    uses Crt, Graph;
    var gM, gD, i, bx, by, x0m, y0m, xm, ym tx1, tx2, ty1, ty2: integer;
        a, b, x, y, ax, ay, kx , ky: real; s1, s2 : string ;
BEGIN
    clrscr;
```

```
setbkcolor (15); setcolor(1);
write ('Введите машинные координаты начала координат ХОУ ');
readln (x0m, y0m);
write ('Введите число пикселей в одном делении на осях ОХ и ОУ ');
readln (bx, by);
write ('Введите цену деления на осях ОХ и ОУ '); readln (ax, ay);
write ('Введите концы отрезка [a,b] '); readln (a, b);D: =
Detect; InitGraph(gD, gM, ' ');
line (10, y0m, 630, y0m); line (615, y0m - 5, 620, y0m);
line (615, y0m + 5, 620, y0m); outtextxy (630, y0m, 'X');
line (x0m, 15, x0m, 460); line (x0m - 10, 25, x0m, 15);
line (x0m + 10, 25, x0m, 15); outtextxy (x0m, 5, 'Y');
tx1 := round((x0m – 10) / bx); tx2 := round((620 - x0m) / bx);
for i := 1 to tx1 do line(x0m –i*bx, y0m – 5, x0m – i*bx, y0m + 5); for i :=
1 to tx2 do line (x0m + i*bx,y0m – 5, x0m + i*bx, y0m + 5);ty1 :=
round((y0k – 15) / by); ty2 := round((460 - y0k) / by);
for i := 1 to ty1 do line (x0k – 5,y0k – i*by,x0k + 5, y0k – i*by); for i :=
1 to ty2 do line (x0k – 5, y0k + i*by, x0k + 5,y0k + i*by);str (ax : 2 : 1,
s1); outtextxy (x0m + bx, y0m + 8, s1);
str (ay : 2 : 1, s2); outtextxy (x0m - 10, y0m - by, s2);kx : =
 bx / ax ; ky : = by / ay ; x : = a;
 repeat
     y : = sqr(x);
     xm : = round(x0m +kx*x); ym : = round(y0m -ky*y);
     circle( xm, ym, 1);x
     : = x + 0.01;
 until x > = b;
 readkey; closegraph;
```

END.

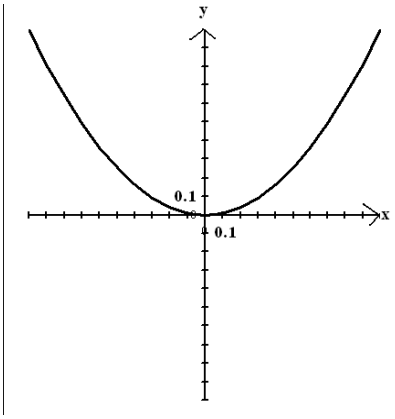


Рис.39 - Модель–график

Если в программу ввести x0m = 320, y0m = 240. bx =10, by = 10, ax = 0.1, ay = 0.1. a = -1, b = 1, то программа построит график, представленный на Рис.38. Эта программа строит график ***любой*** функции на ***любом*** промежутке, пользователю программы нужно только правильно выбрать начало координат строящейся системы координат, цену деления на осях координат и масштаб. Если это сделать неправильно, то график может получиться ненаглядным - занимать очень незначительную часть экрана, или Рис.39 не отображать всех значений функции на промежутке.