Урок 1

Знакомство с графическим модулем и процедурами, устанавливающими текущий цвет рисунка, тип закраски и ее цвет, рисующими прямоугольник и трехмерный параллелепипед.

Uses Graph – подключение модуля.
Close Graph – останавливает работу графической системы.

Экран:

Координаты левого верхнего угла экрана – 0,0.
X увеличивается слева направо от 0 до 639.
Y увеличивается сверху вниз от 0 до 479.

Инициализация графической системы и перевод аппаратуры в графический режим.

Процедура
InitGraph(var граф_драйвер: integer;var граф_режим: integer;
путь_драйвер:string).

Процедура **SetColor(цвет:word)** Устанавливает текущий цвет рисунка.

0	черный
1	синий
2	зеленый
3	бирюзовый
4	красный
5	малиновый
6	коричневый
7	светло-серый
8	темно-серый
9	светло-голубой
10	светло-зеленый
11	светло-бирюзовый
12	светло-красный
13	светло-малиновый
14	желтый
15	белый

Процедура **SetFillStyle**(образец:word;цвет:word)Устанавливает тип закраски и ее цвет.

0	заполнение фоновым цветом
1	непрерывное заполнение
2	заполнить ---
3	заполнить ///
4	заполнить жирными ///
5	заполнить жирными \\\
6	заполнить \\\
7	заполнить редкой штриховкой
8	частой пересекающейся штриховкой
9	заполнить прерывистыми линиями
10	заполнить редкими

Процедура **Bar**(x1,y1,x2,y2:integer)

Рисует столбец, используя стандартный тип и цвет закраски, заданные с помощью процедуры **SetFillStyle**.

Процедура **Bar3d**(x1,y1,x2,y2:integer;глубина:word;вершина:boolean)

Используя текущий тип и цвет закраски, рисует трехмерный параллелепипед. Вершина может быть истина – TRUE, тогда рисуется верхняя грань и ложь – FALSE, тогда верхняя грань не рисуется.

Пример: Трехмерный параллелепипед.

```
Program gr;
Uses crt,graph;
Var gd,gm:integer;
Begin
     gd:=detect;
     InitGraph(gd,gm,'c:\tp\');
     SetFillStyle(8,5);
     SetColor(5);
     Bar3d(100,100,300,250,60,true);
     Repeat until keypressed;
     CloseGraph;
End.
```

Результат

Урок 2

Линии, окружности, прямоугольники, точки, заливка

Процедура **SetLineStyle**(тип_линии:word;образец:word;толщина:word) Устанавливает текущую толщину и тип линии; параметр образец используется для типа, определяемого пользователем, для остальных типов этот параметр=0.

Тип линии

0	непрерывная линия
1	линия из точек
2	линия из точек и тире
3	пунктирная линия
4	тип, определяемый пользователем

Толщина линии

1	нормальная толщина
3	жирная линия

Примеры программ

Закрашенные концентрические окружности

```
program krugi;
uses crt,graph;
var gd,gm:integer;
    x0,r,y0:integer;c:word;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  c:=1;
  x0:=320;y0:=240;r:=100;
  while r>0 do begin
    SetColor(c);
    SetFillStyle(1,c);
    Circle(x0,y0,r);
    FloodFill(x0,y0,c);
    r:=r-5;if c<15 then c:=c+1 else c:=1
              end;
  repeat until keypressed; closegraph
end.
```

Результат

Типы линий

```
program type_line;
uses crt,graph;
var gd,gm:integer;
    x1,y,x2,c,t:integer;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  x1:=0;x2:=GetmaxX;y:=0;t:=0;
  while y<GetmaxY do begin
    SetColor(15);SetLineStyle(t,0,1);
    line(x1,y,x2,y);
    x1:=x1+10;x2:=x2-10;y:=y+15;
    if t<3 then t:=t+1 else t:=1;
                    end;
  repeat until keypressed;
  closegraph
end.
```

Результат

Эллипсы

```
program ellipsy;
uses crt,graph;
var d,min,x,x0,y,y0,gd,gm:integer;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  setcolor(10);
  min:=5;x0:=320;y0:=240;y:=200;x:=min;d:=5;
  while y>=min do begin
    ellipse(x0,y0,0,359,x,y);
     x:=x+d;y:=y-d end;
  repeat until keypressed;
  closegraph
end.
```

Результат

Урок 3

Использование случайных чисел

Процедура **Randomize** – инициализирует генератор случайных чисел;

Функция **Random(n)** – возвращает случайное целое число в диапазоне от 0 до n;

Примеры программ:

Звездное небо

```
program zwezda;
uses crt,graph;
var gd,gm:integer;
    i,x,y,c:integer;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  randomize;
  for i:=1 to 1000 do begin
    x:=random(639);
    y:=random(479);
    c:=random(15);
    PutPixel(x,y,c);
                        end;
  repeat until keypressed;
  closegraph end.
```

Результат

Случайные точки в центре экрана

```
program krug;
uses crt,graph;
var gd,gm,i,r,c,x,y:integer;
    xc,yc:real;
begin
  gd:=detect;r:=50;
  initgraph(gd,gm,'c:\tp\');
  randomize;
  repeat
    x:=random(639);y:=random(479);
    c:=random(15);
    xc:=x-320;yc:=y-240;
    if xc*xc+yc*yc<=r*r then
    PutPixel(x,y,c);
  until keypressed;
  closegraph end.
```

Результат

Процедура **Line**(x1,y1,x2,y2:integer)

Рисует прямую линию из точки (x1,y1) в точку (x2,y2), толщина и тип которой задаются процедурой SetLineStyle, а цвет устанавливается процедурой SetColor.

Процедура **Circle**(x,y:integer;радиус:word)

Рисует окружность с заданным радиусом и центром, используя текущий цвет

Процедура **Ellipse**(x,y:integer;нач_угол,кон_угол:word;радX, радY:word)

Рисует эллиптическую дугу от начального до конечного угла, используя x,y в качестве центра, радX и радY – горизонтальная и вертикальная оси

Процедура **Rectangle**(x1,y1,x2,y2:integer)

Рисует прямоугольник, используя текущий цвет и тип строки. (x1,y1) – верхний левый угол прямоугольника, (x2,y2) – нижний правый. (0<=x1<x2<=GetMaxX и 0<=y1<y2<=GetMaxY)

Функция **GetMaxX** и Функция **GetMaxY**

возвращают для текущего графического режима и драйвера максимальное значение x (639)и y (479).

Процедура **PutPixel**(x,y:integer;цвет:word)

Строит точку в (x,y) цветом, заданным параметром цвет.

Процедура **FloodFill**(x,y,граница:word)

Заполняет ограниченную область, используя текущий заполнитель.

Урок 4

Рисование линий

Текущий указатель в графическом режиме аналогичен курсору, который используется в текстовом режиме. Различает их то, что текущий указатель не виден.

Процедура **LineRel**(dx,dy:integer)

Проводит прямую линию в точку, заданную относительным расстоянием (dx,dy) от текущего указателя;

Процедура **LineTo**(x,y:integer)

Рисует прямую линию из точки, в которой находится указатель в точку (x,y);

Процедура **MoveRel**(dx,dy:integer)

Перемещает текущий указатель на расстояние, заданное относительными координатами от его текущей позиции;

Процедура **MoveTo**(x,y:integer)

Перемещает текущий графический указатель в точку (x,y).

Примеры программ

Развернутые конверты

```
program konwert;
uses crt,graph;
var gd,gm:integer;
    x0,y0,r:integer;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  x0:=100;y0:=400;r:=100;
  MoveTo(x0,y0);
  LineTo(x0,y0-r);LineTo(x0+r,y0-r);
  LineTo(x0+r,y0);LineTo(x0,y0);
  LineTo(x0+r,y0-r);LineTo(x0+r div 2,y0-r*2);
  LineTo(x0,y0-r);LineTo(x0+r,y0);
  MoveRel(r,0);
  LineRel(0,-r);LineRel(r,0);
  LineRel(0,r);LineRel(-r,0);
  LineRel(r,-r);LineRel(-r div 2,-r);
  LineRel(-r div 2,r);LineRel(r,r);
  repeat until keypressed;
  closegraph end.
```

Результат

Ступени

```
program stupeni;
uses crt,graph;
var gd,gm:integer;
    x0,y0,r,i:integer;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  x0:=10;y0:=400;r:=20;
  MoveTo(x0,y0);
  for i:=1 to 10 do begin
    LineRel(0,-r);LineRel(r,0)
                  end;
  for i:=1 to 10 do begin
    LineRel(0,r);LineRel(r,0)
                  end;
  repeat until keypressed;
  closegraph
end.
```

Результат

Урок 5

Движение объекта

Примеры программ:

Движущийся квадрат

```
program dw;
uses crt,graph;
var x1,x2,y1,y2,gd,gm:integer;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  x1:=0;y1:=0;x2:=20;y2:=17;
  repeat
    setcolor(10);
    Rectangle(x1,y1,x2,y2);delay(10000);
    setcolor(0);
    Rectangle(x1,y1,x2,y2);
    x1:=x1+10;x2:=x2+10;y1:=y1+7;y2:=y2+7
  until (x2>=GetMaxX) or (y2>=GetMaxY);
  setcolor(10);
  Rectangle(x1,y1,x2,y2);
  repeat until keypressed;
  closegraph
end.
```

Результат

Движение человечка

```
program ch;
uses crt,graph;
var gd,gm:integer;
    x0,x,r,y0,y:integer;
procedure chelovek(xp,yp:integer);
begin
  Circle(xp,yp,r);
  Line(xp,yp+r,xp,yp+2*r);
  Line(xp,yp+2*r,xp+r,yp+3*r);
  Line(xp,yp+2*r,xp-r,yp+3*r);
  Line(xp,yp+2*r,xp,yp+3*r);
  Line(xp,yp+3*r,xp+r,yp+5*r);
  Line(xp,yp+3*r,xp-r,yp+5*r);
end;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  x0:=20;y0:=240;r:=20;
  x:=x0;y:=y0;
  while x<600 do begin
    SetColor(14);chelovek(x,y); Delay(7000);
    SetColor(0);chelovek(x,y);
    x:=x+20;     end;
    SetColor(14);chelovek(x,y);
  repeat until keypressed;
  closegraph end.
```

Результат

Движение с изменением цвета и типа закраски

```
program diagonali;

uses crt,graph;

var i,j,x1,x2,y1,y2,gd,gm:integer;

begin

gd:=detect;  initgraph(gd,gm,'c:\tp\');  rectangle(0,0,639,479);

repeat  x1:=1;y1:=1;x2:=100;y2:=50;  j:=1;i:=0;{движение вниз и вправо}
while x2<620 do begin  SetFillStyle(j,i);  bar(x1,y1,x2,y2);
{delay(15000);Setfillstyle(1,0);bar(x1,y1,x2,y2);}
x1:=x1+10;y1:=y1+8;x2:=x2+10;y2:=y2+8;

if j<11 then j:=j+1 else j:=1;  if i<15 then i:=i+1 else i:=1;
end;

x1:=1;y1:=478;x2:=100;y2:=429;  j:=1; i:=0;{движение вверх и вправо}
while x2<620 do begin  SetFillStyle(j,i);
bar(x1,y1,x2,y2);{delay(10000);Setfillstyle(1,0);bar(x1,y1,x2,y2);}
x1:=x1+10;y1:=y1-8;x2:=x2+10;y2:=y2-8;  if j<11 then j:=j+1 else j:=1;
if i<15 then i:=i+1 else i:=1;

end;

until keypressed;

closegraph;

end.
```

Результат

Урок 6

Графики функций

Началом координат будем считать точку с координатами (320, 240). Тогда оси координат:

line(0,240,639,240) – ось X;

line(320,0,320,479) – ось Y;

Построим график функции **Y=X**.

Пусть X изменяется от -200.

Начальное значение X=-200.

В цикле будем увеличивать значение аргумента на шаг (например на 5), вычислять значение функции и ставить точку с координатами X, Y с помощью процедуры PutPixel.

В параметрах процедуры перед значением Y будем ставить знак минус, так как 0 по оси Y вверху.

Все значения Y будем увеличивать на 240, а X – на 320, чтобы «передвинуть» график в нашу систему координат.

Для растягивания графика по осям координат (иначе его не будет видно) умножаем значения X и Y на коэффициент 10, то есть меняем масштаб графика.

Примеры программ:

График функции Y=X

```
program grafiki;
uses crt,graph;
var gd,gm:integer;
    i,x0,y0:integer;x,y:real;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  SetColor(7);
  rectangle(0,0,639,479);
  x0:=320;y0:=240;
  line(320,0,320,479);line(0,y0,639,y0);
  x:=-20;
  for i:=1 to 80 do begin
    x:=x+0.5;y:=x;PutPixel(round(x*10+x0),round(-y*10+y0),7)
                    end;
  repeat until keypressed;
  closegraph;
end.
```

Результат

Для графиков тригонометрических функции нужно задавать другие коэффициенты по осям координат, так в следующем примере эти коэффициенты 20 и 50. Аргумент тригонометрической функции – угол в радианах

Графики тригонометрических функций: sinX, cosX, tgX

```
program grafiki;
uses crt,graph;
var gd,gm:integer;
    i,n,x0,x,y0,y,y1,y2:integer;
    ug:real;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  SetColor(7);
  rectangle(0,0,639,479);
  x0:=0;y0:=240;ug:=0.1;
  line(0,0,0,479);line(0,240,639,240);
  for i:=1 to 1000 do begin
    x:=round(ug*20+x0);
    y:=round(-sin(ug)*50+y0);
    y1:=round(-cos(ug)*50+y0);
    y2:=round((-sin(ug)/cos(ug)+y0));
    PutPixel(x,y,4);PutPixel(x,y1,2);PutPixel(x,y2,14);
    ug:=ug+pi/100     end;
  repeat until keypressed;
  closegraph;
end.
```

Результат

Линия, соединяющая графики синуса и косинуса

```
program grafiki;
uses crt,graph;
var gd,gm:integer;
    i,n,x0,x,y0,y,y1,c:integer;
    ug:real;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  SetColor(7);
  rectangle(0,0,639,479);
  repeat
   x0:=10;y0:=240;ug:=0;c:=1;
   for i:=1 to 100 do begin
    x:=round(ug*30+x0);
    y:=round(-sin(ug)*100+y0);
    y1:=round(-cos(ug)*150+y0);
    setcolor(c);SetLineStyle(0,0,3);
    line(x,y,x+10,y1);delay(5000);
    {setcolor(0);
    line(x,y,x+10,y1);}{убрать фигурные скобки для изменения картинки}
    ug:=ug+pi/18;if c<15 then c:=c+1 else c:=1
                        end;
   until keypressed;
   closegraph;end.
```

Результат

Графики функции sinX*sinX/10 с разными амплитудами

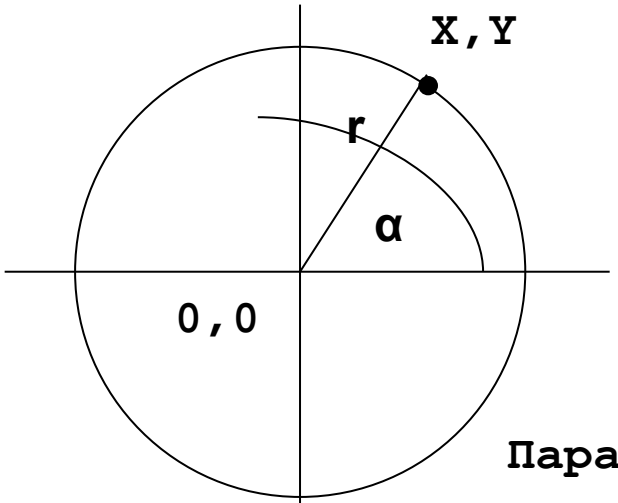
```
program grafiki;
uses crt,graph;
var gd,gm:integer;
    i,n,x0,x,y0,y,y1,y2:integer;
    ug:real;
begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  SetColor(7);rectangle(0,0,639,479);
  x0:=0;y0:=240;ug:=0.1;
  line(0,0,0,479);line(0,240,639,240);
  for i:=1 to 1500 do begin
    x:=round(ug*10+ug/9+x0);
    y:=round(-sin(ug)*sin(ug/10)*50+y0);
    y1:=round(-sin(ug)*sin(ug/10)*60+y0);
    y2:=round(-sin(ug)*sin(ug/10)*70+y0);
    PutPixel(x,y,4);PutPixel(x,y1,2);PutPixel(x,y2,14);
    ug:=ug+pi/60    end;
  repeat until keypressed;
  closegraph;
end.
```

Результат

Урок 7

Параметрическое задание кривых

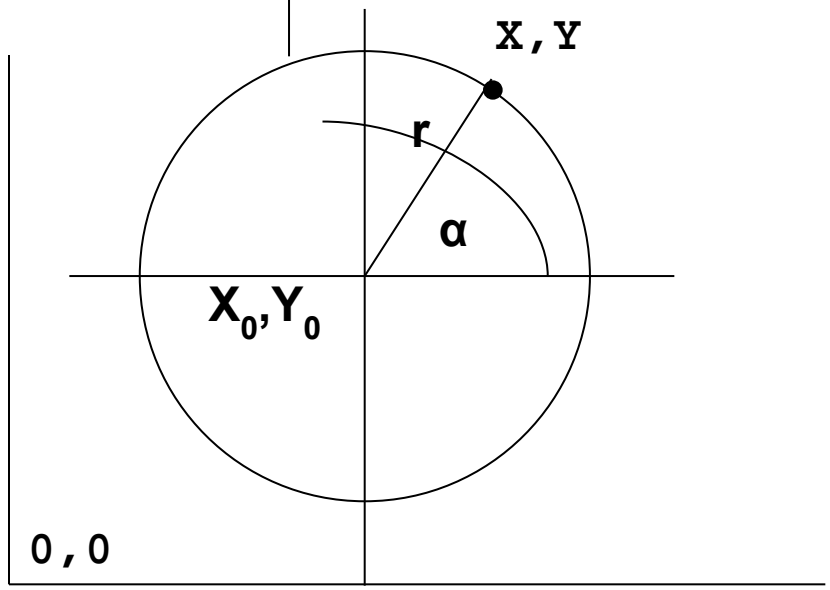
Параметрическое задание окружности с центром в начале координат



$$\sin\alpha = \frac{y}{r} \quad y = r\sin\alpha$$

$$\cos\alpha = \frac{x}{r} \quad x = r\cos\alpha$$

Параметрическое задание любой окружности



$$x = x_0 + r\cos\alpha$$

$$y = y_0 + r\sin\alpha$$

Примеры программ:

Окружность

```
program krug;
uses crt,graph;
var gd,gm:integer;
    x,y,f:real;
    i,x0,y0,r,c:integer;
begin
  gd:=detect;
  write('введите радиус окружности и цвет (1-15)');read(r,c);
  initgraph(gd,gm,'c:\tp\');
  x0:=320;y0:=240;f:=0;MoveTo(x0+r,y0);
  SetColor(c);
  for i:=1 to 100 do begin
    f:=f+2*pi/100;
    x:=x0+r*cos(f);y:=y0+r*sin(f);
    LineTo(round(x),round(y));
                              end;
  repeat until keypressed;
  closegraph
end.
```

Результат

“Убегающий многоугольник”

```
program ubeg_mn;
uses crt,graph;
var i,x0,y0,r,gd,gm,n,c:integer;
    f,x,y:real;
begin
  write('введите число сторон многоугольника_');read(n);
  gd:=detect;
  initgraph(gd,gm,'c:\tp\');
  c:=1;
  x0:=320;y0:=240;r:=150;f:=0;
  while r>0 do begin
    setcolor(c);
    x:=x0+r*cos(f);y:=y0+r*sin(f);
    moveto(round(x),round(y));
    for i:=1 to n do begin
      f:=f+2*pi/n;
      x:=x0+r*cos(f);y:=y0+r*sin(f);
      lineto(round(x),round(y));delay(40000);
                        end;
    f:=f+3;r:=r-3;if c<14 then c:=c+1 else c:=1;
               end;
  repeat until keypressed;
  closegraph
end.
```

Результат

Салфетка

```
program salfetka;
uses crt,graph;
var gd,gm:integer;
    f:real;
    i,j,x0,y0,n,c,r:integer;
    vr:array[1..100,1..2]of integer;
begin
  gd:=detect;
  write('введите число вершин и цвет (1-15)');read(n,c);
  initgraph(gd,gm,'c:\tp\');
  x0:=320;y0:=240;f:=0;r:=130;
  SetColor(c);
  for i:=1 to n do begin {заполнение массива вершин}
    vr[i,1]:=round(r*cos(f)+x0);
    vr[i,2]:=round(r*sin(f)+y0);
    f:=f+2*pi/n;   end;
  for i:=1 to n-1 do
   for j:=i+1 to n do
    Line(vr[i,1],vr[i,2],vr[j,1],vr[j,2]);
  repeat until keypressed;
  closegraph
end.
```

Результат