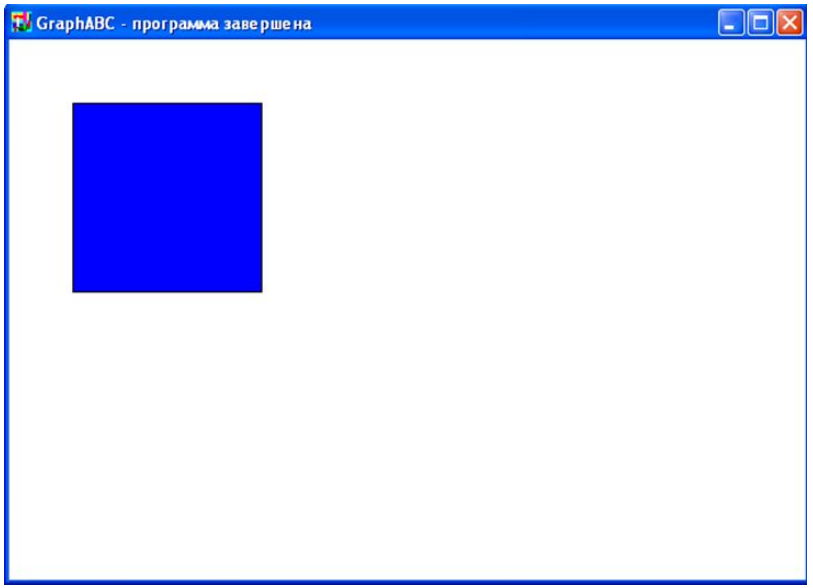
Графика в Pascal

Для того, чтобы использовать его графические средства нужно подключить **графический модуль.**

uses **GraphABC** функция подключения графического модуля записывается перед объявлением переменных

```
uses GraphABC;
var v,k: integer;
begin
```



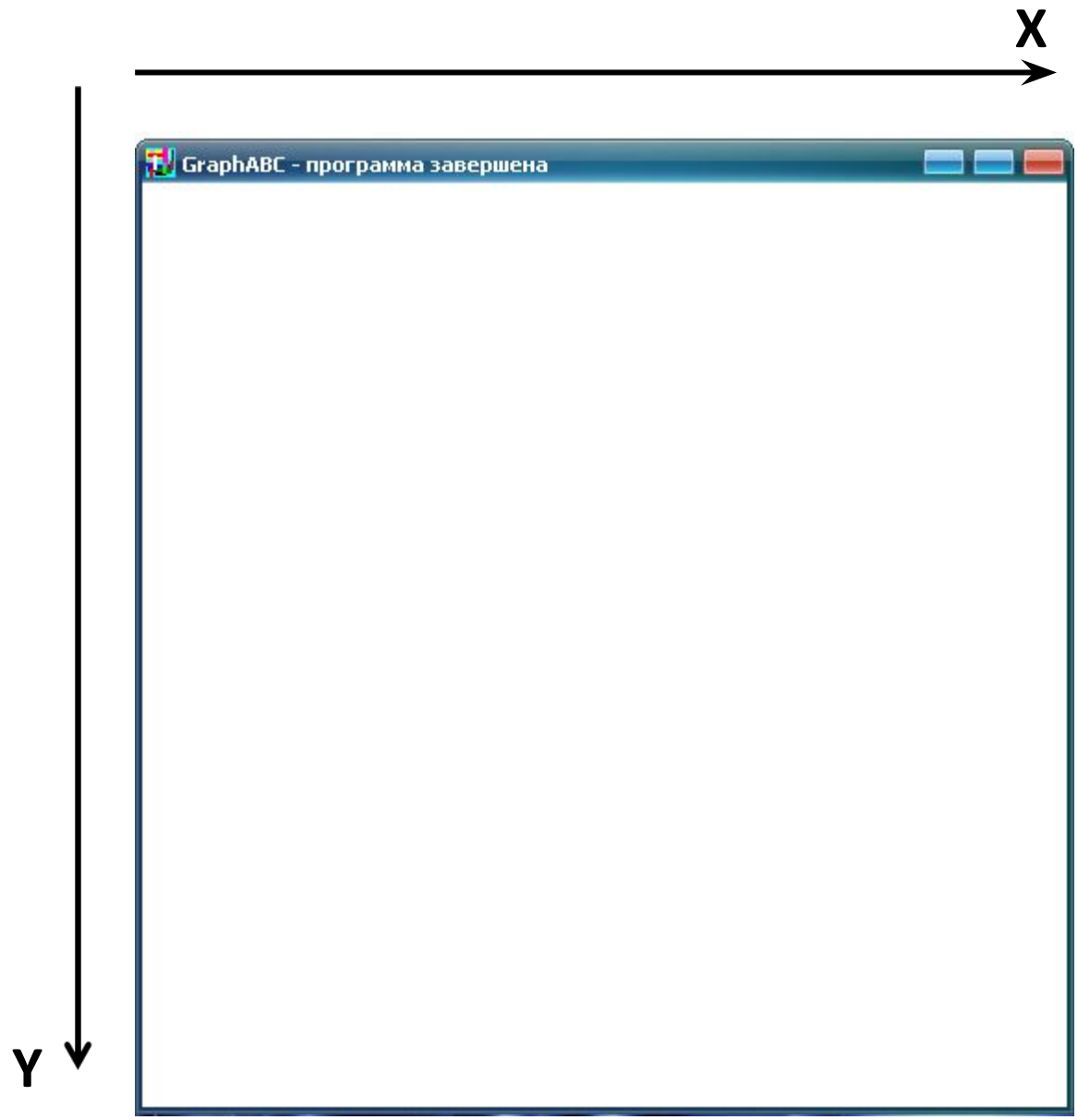
SetWindowPos(x,y); // Устанавливает позицию графического окна относительно монитора.

SetWindowWidth (500); //Устанавливает ширину графического окна

SetWindowHeight(100); //Устанавливает высоту графического окна.

SetWindowHeight(500);

SetWindowWidth(500);



Отрезок рисуется процедурой **Line.** Мы знаем, что отрезок прямой можно построить, если известно положение его двух крайних точек. Они-то и задаются в обращении к процедуре. Первая пара параметров - координаты одной точки (любой из двух), вторая пара - другой.

Line(100,50,250,150);

Прямоугольник рисуется процедурой **Rectangle**
где в параметрах указывают
Координаты правый верхний и левый нижний углы

Rectangle(150,150,50,200);

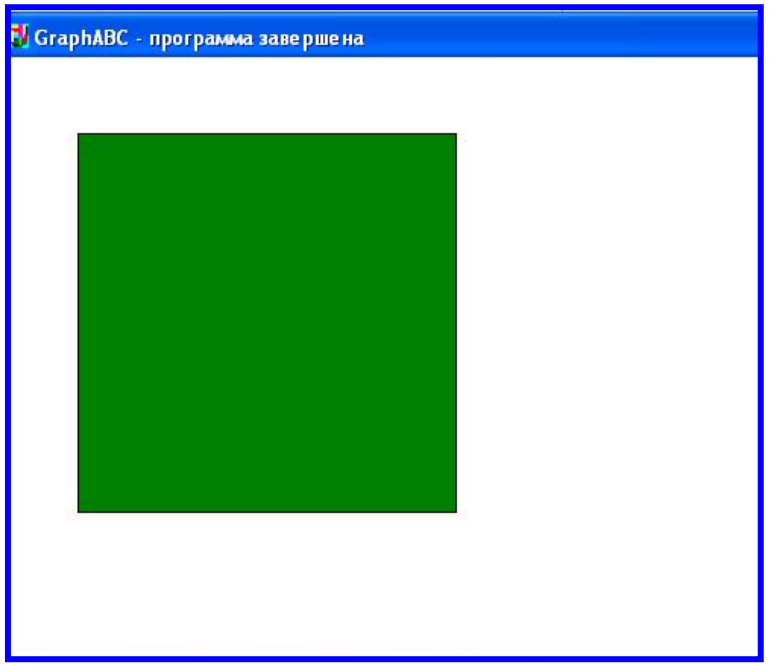
Окружность можно построить, если известно положение центра и радиус. Окружность рисуется процедурой **Circle**, первые два параметра которой - координаты центра, третий - радиус.

Circle(200,100,50);

Заливка кистью

SetBrushColor(color) - устанавливает цвет кисти.

Заливка кистью распространяется на замкнутый контур, описание которого следует за процедурой установки цвета кисти.




```
Program zalivka_kist;
uses GraphABC;
Begin
  SetBrushColor(clGreen);
  Rectangle(50,50,300,300);
end.
```

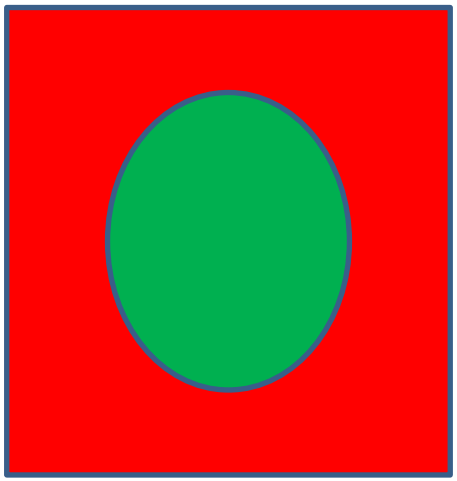
Используемые цвета

clBlack – черный	clAqua – бирюзовый
clPurple – фиолетовый	clOlive – оливковый
clWhite – белый	clFuchsia – сиреневый
clMaroon – темно-красный	clTeal – сине-зеленый
clRed – красный	clGray – темно-серый
clNavy – темно-синий	clLime – ярко-зеленый
clGreen – зеленый	clMoneyGreen – цвет зеленых денег
clBrown – коричневый	clLtGray – светло-серый
clBlue – синий	clDkGray – темно-серый
clSkyBlue – голубой	clMedGray – серый
clYellow – желтый	clSilver – серебряный
clCream – кремовый	

Random(16777215) – случайный цвет из всей палитры цветов Паскаля

```
USES      GraphABC;
const h= clRed;
d=clGreen;
BEGIN
  SetWindowWidth(500);    //Устанавливает высоту графического
окна

 SetWindowHeight(500); //Устанавливает ширину графического окна.

 SetWindowPos(0,0); // Устанавливает позицию графического окна.

  SetBrushColor(h );
   Rectangle(150,150,50,200);     //правый верхний и левый нижний
углы}

 SetBrushColor(d );
  Circle(200,100,50);                    {окружность}

Line(100,50,250,150);              {отрезок прямой}
ReadLn;

END.
```

- Составить программу рисующую красный квадрат в центре формы и две горизонтальные прямые, внутри квадрата зеленая окружность.



Очистка графического окна

ClearWindow; - **очищает графическое окно белым цветом.**

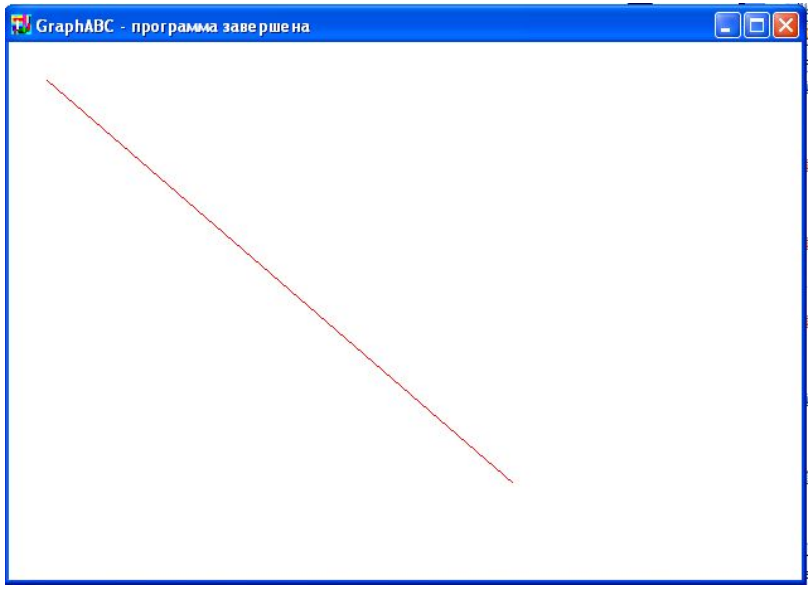
ClearWindow(color); - **очищает графическое окно указанным цветом.**

Цвет зеленых денег

```
uses GraphABC;
begin
ClearWindow;
ClearWindow(clMoneyGreen);
end.
```

Цвет линии

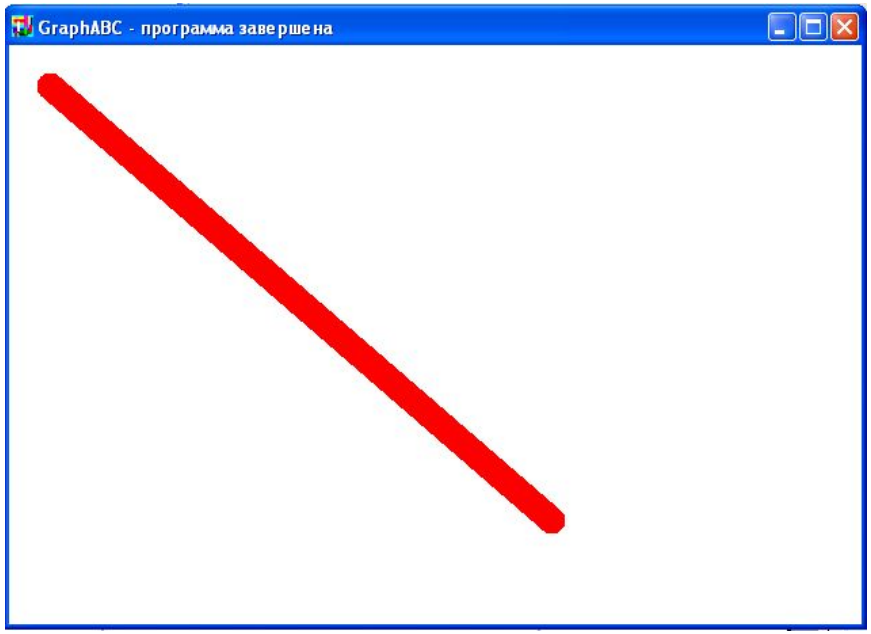
SetPenColor(color) - устанавливает цвет пера, задаваемый параметром color.



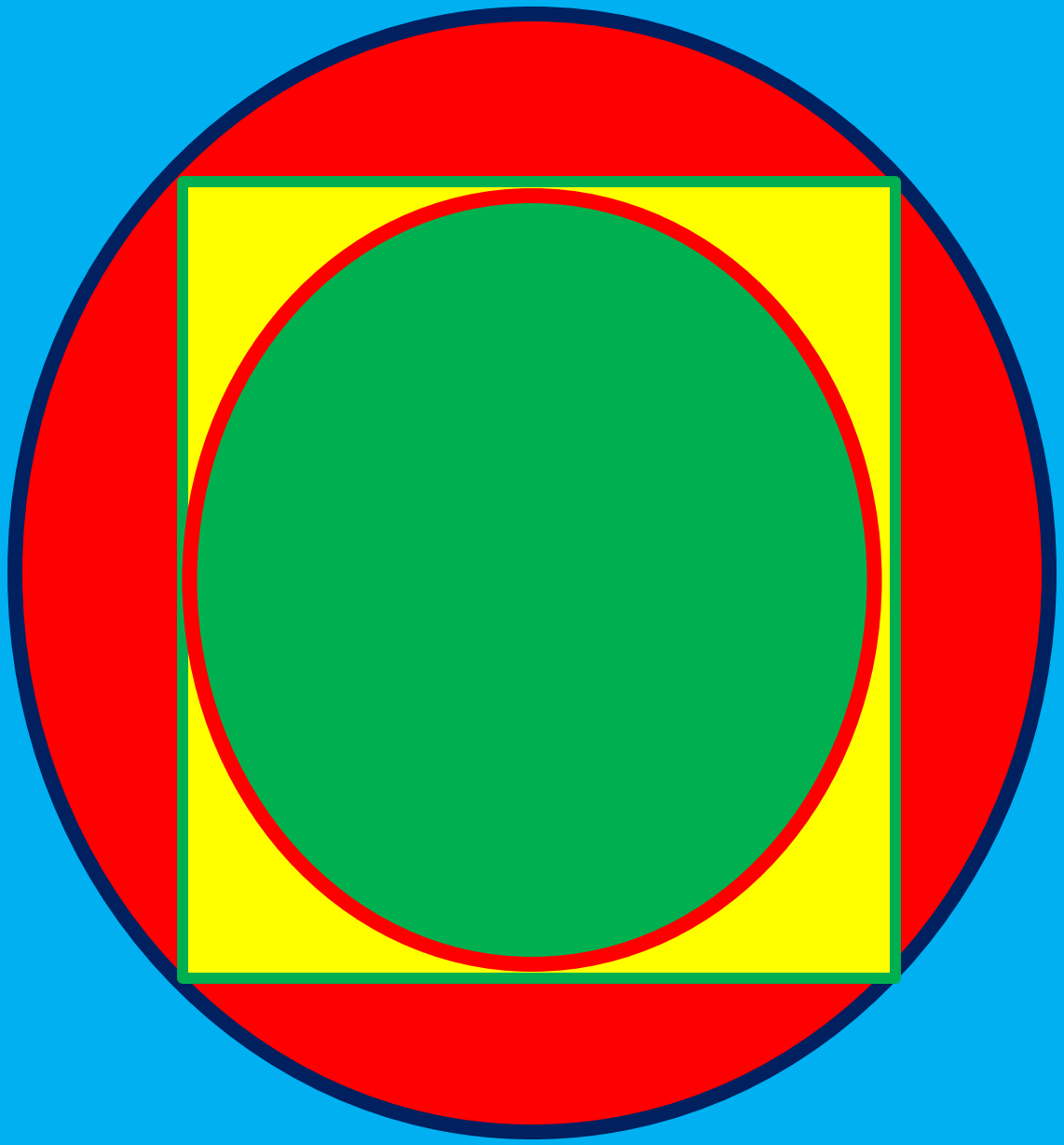

```
Program liniay;
uses GraphABC;
begin
  setpencolor(clred);
  line(30,30,400,350);
end.
```

Толщина линии

SetPenWidth(n) - устанавливает ширину (толщину) пера, равную n пикселям.

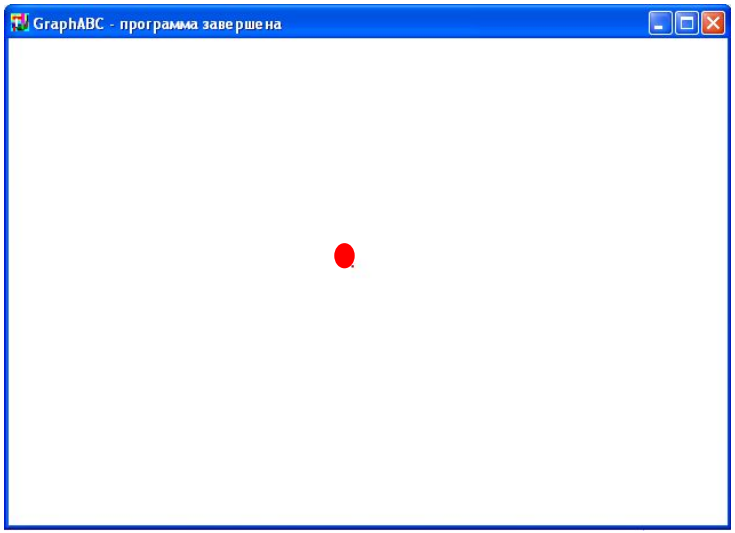



```
Program liniay;
uses GraphABC;
begin
  setpenwidth(20);
  setpencolor(clred);
  line(30,30,400,350);
end.
```



Точка

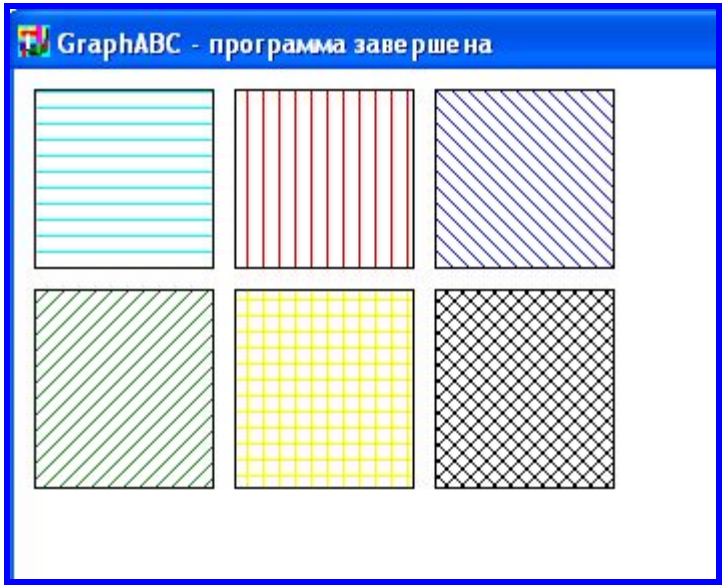
SetPixel(x,y,color) – **Закрашивает один пиксел с координатами (x,y) цветом color**




```
program tochka;
uses GraphABC;
begin
  SetPixel(300,200,clred);
end.
```

Заливка кистью

SetBrushStyle(номер от 0 до 7 или название) – устанавливает стиль кисти, задаваемый номером или символической константой.

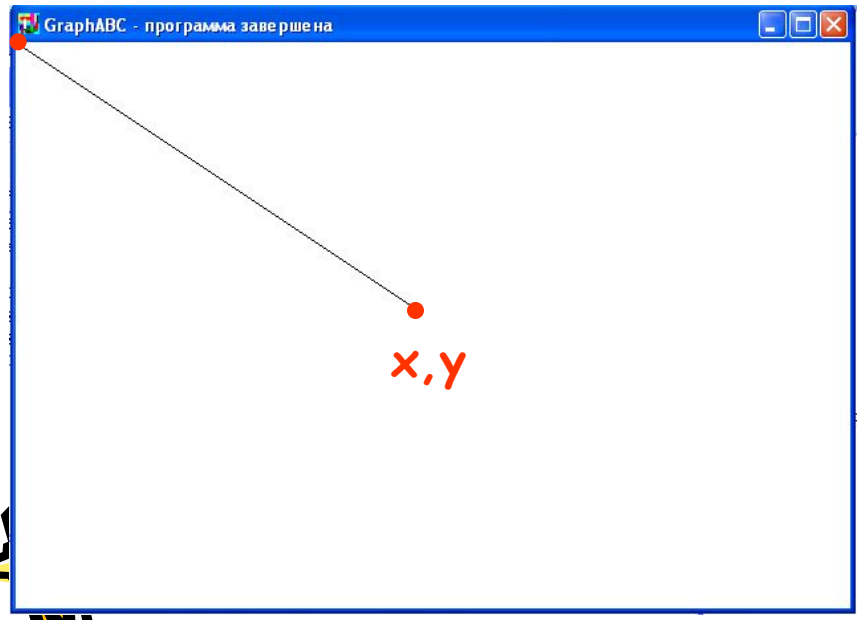


По умолчанию задается стиль 0 – сплошная заливка цветом.

```
Program p12_zalivka;
uses GraphABC;
Begin
 SetBrushColor(clAqua);
 SetBrushStyle(1);
 Rectangle(10,10,100,100);
 SetBrushColor(clRed);
 SetBrushStyle(2);
 Rectangle(110,10,200,100);
 SetBrushColor(clBlue);
 SetBrushStyle(3);
 Rectangle(210,10,300,100);
 SetBrushColor(clGreen);
 SetBrushStyle(4);
 Rectangle(10,110,100,210);
 SetBrushColor(clYellow);
 SetBrushStyle(5);
 Rectangle(110,110,200,210);
 SetBrushColor(clBlack);
 SetBrushStyle(6);
 Rectangle(210,110,300,210);
end.
```

Линии

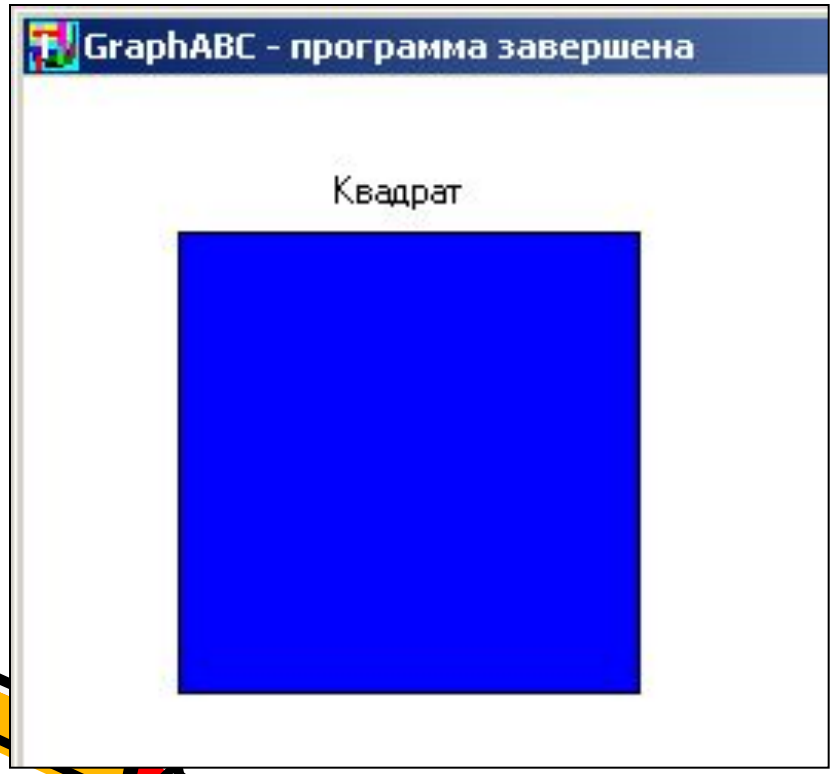
LineTo(x,y) - рисует отрезок от текущего положения пера до точки (x,y); координаты пера при этом также становятся равными (x,y).




```
Program liniay;
uses GraphABC;
begin
LineTo(300,200);
end.
```

Вывод текста в графическое окно

TextOut(x,y,'Текст'); – выводит строку текста в позицию (x,y) (точка (x,y) задает верхний левый угол прямоугольника, который будет содержать текст).




```
Program text;
uses GraphABC;
begin
TextOut(100,30,'Квадрат');
 Rectangle(50,50,200,200);
end.
```

Действия со шрифтом

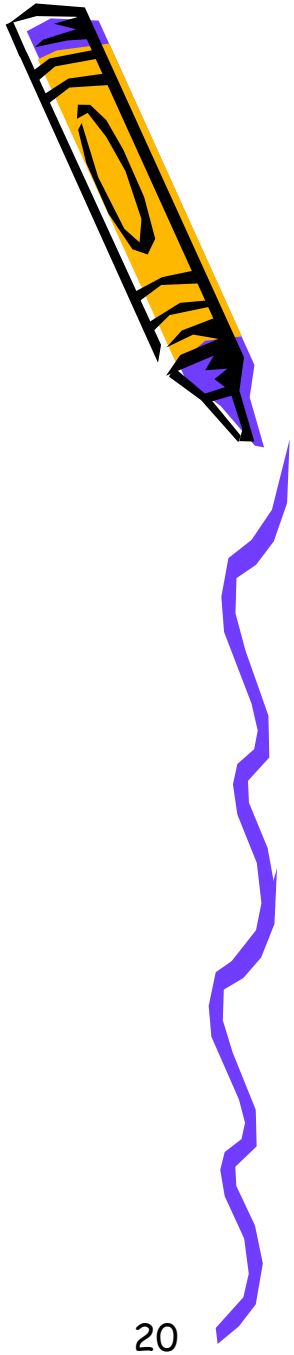
SetFontName('name')- устанавливает наименование шрифта.

SetFontColor(color) - устанавливает цвет шрифта.

SetFontSize(sz) - устанавливает размер шрифта в пунктах.

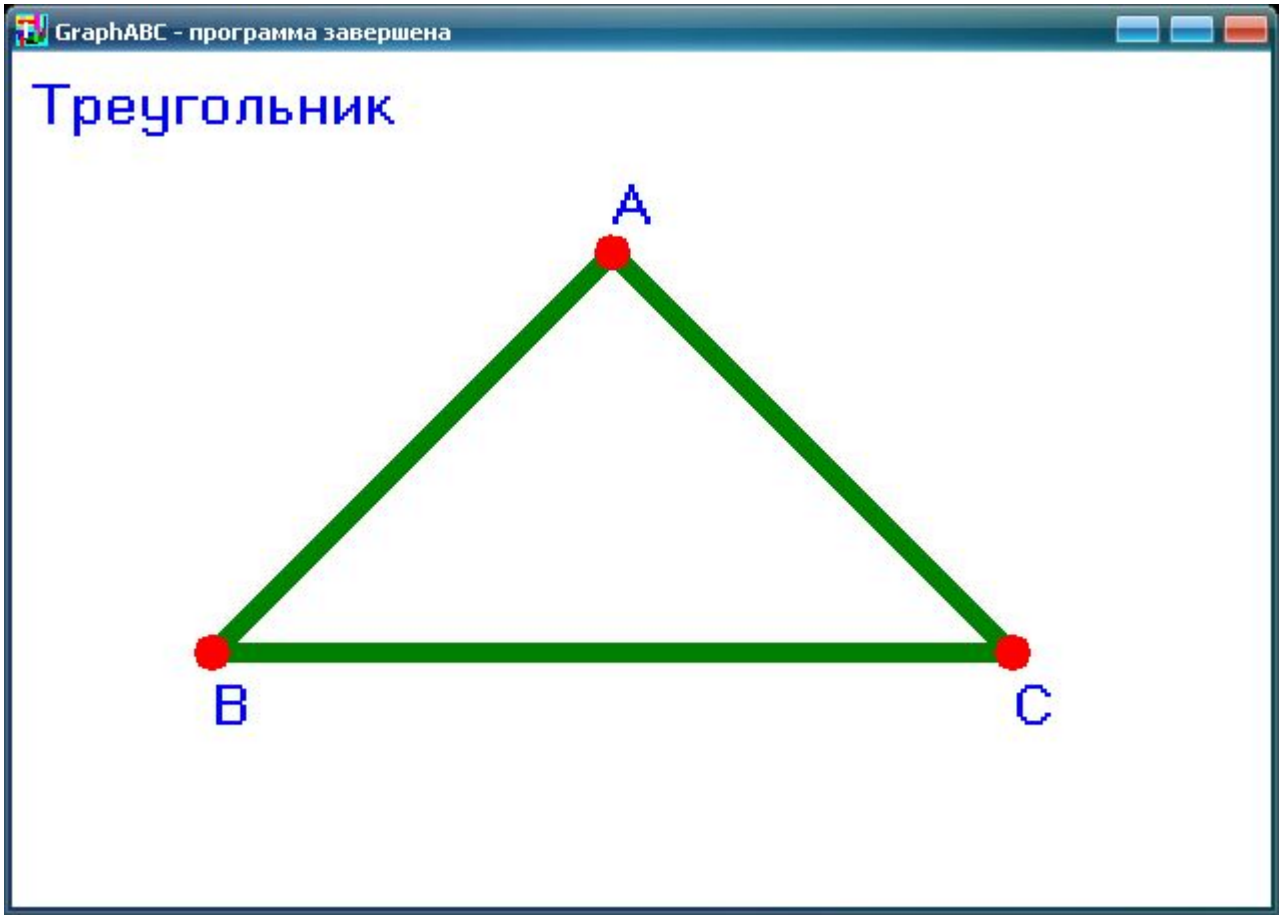
SetFontStyle(fs) - устанавливает стиль шрифта.

fsNormal – обычный;
fsBold – жирный;
fsItalic – наклонный;



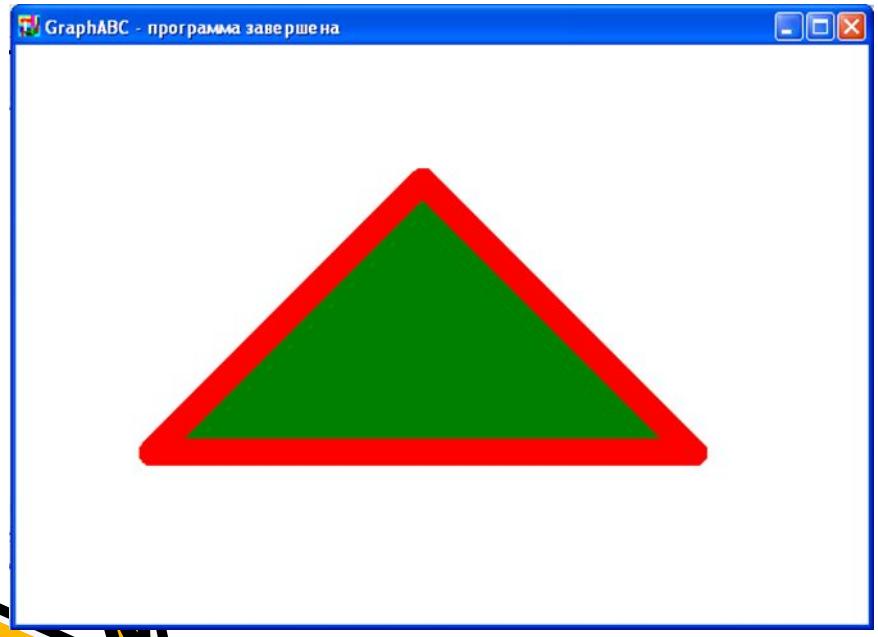
```
SetFontColor(clBlue);
SetFontSize(24);
SetFontStyle(fsBold);

TextOut(10,10,'Треугольник');
```

Создать программу
рисующую треугольник,
где выделены и названы вершины



Заливка цветом

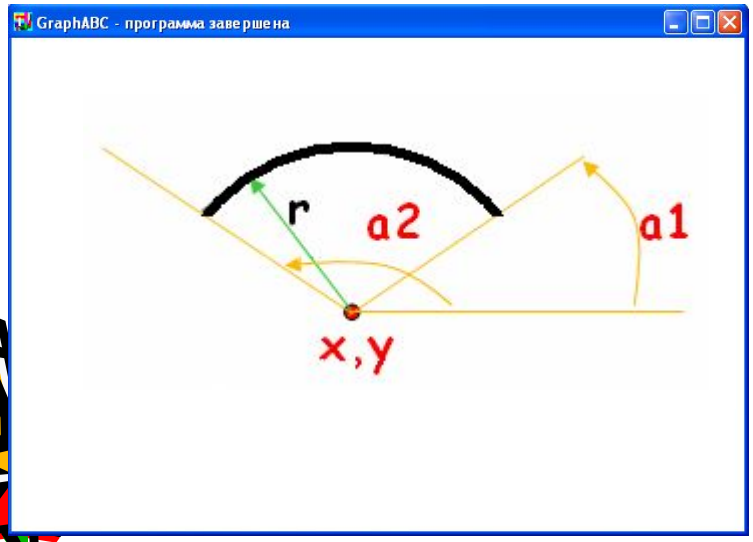
FloodFill(x,y,color) – заливает область одного цвета цветом color, начиная с точки (x,y).




```
Program treugolnik;
uses GraphABC;
begin
  setpenwidth(20);
  setpencolor(clred);
  line(300,100,500,300);
  lineto(100,300);
  lineto(300,100);
  floodfill(300,200,clgreen);
end.
```

Дуга окружности

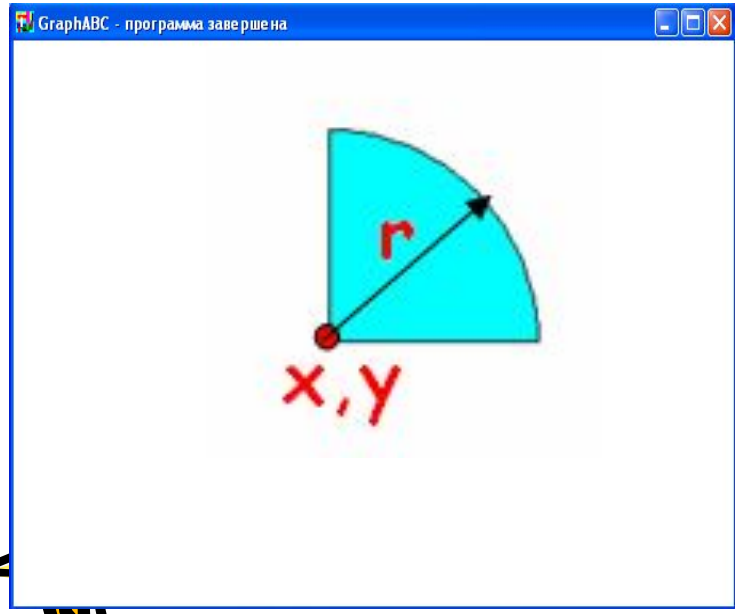
Arc(x,y,r,a1,a2) - Рисует дугу окружности с центром в точке (x,y) и радиусом r, заключенной между двумя лучами, образующими углы a1 и a2 с осью OX (a1 и a2 – вещественные, задаются в градусах и отсчитываются против часовой стрелки).




```
Program duga;
uses GraphABC;
Begin
SetPenWidth(10);
Arc(300,250,150,45,135);
end.
```

Сектор

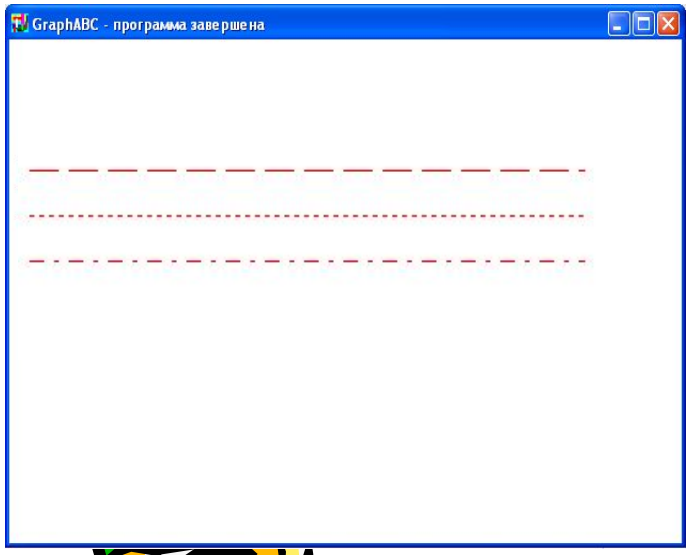
Pie(x,y,r,a1,a2) – рисует сектор окружности, ограниченный дугой (параметры процедуры имеют тот же смысл, что и в процедуре Arc).




```
Program sector;
uses GraphABC;
begin
Pie(300,200,100,0,90);
FloodFill(300+10,200-10,clAqua);
end.
```

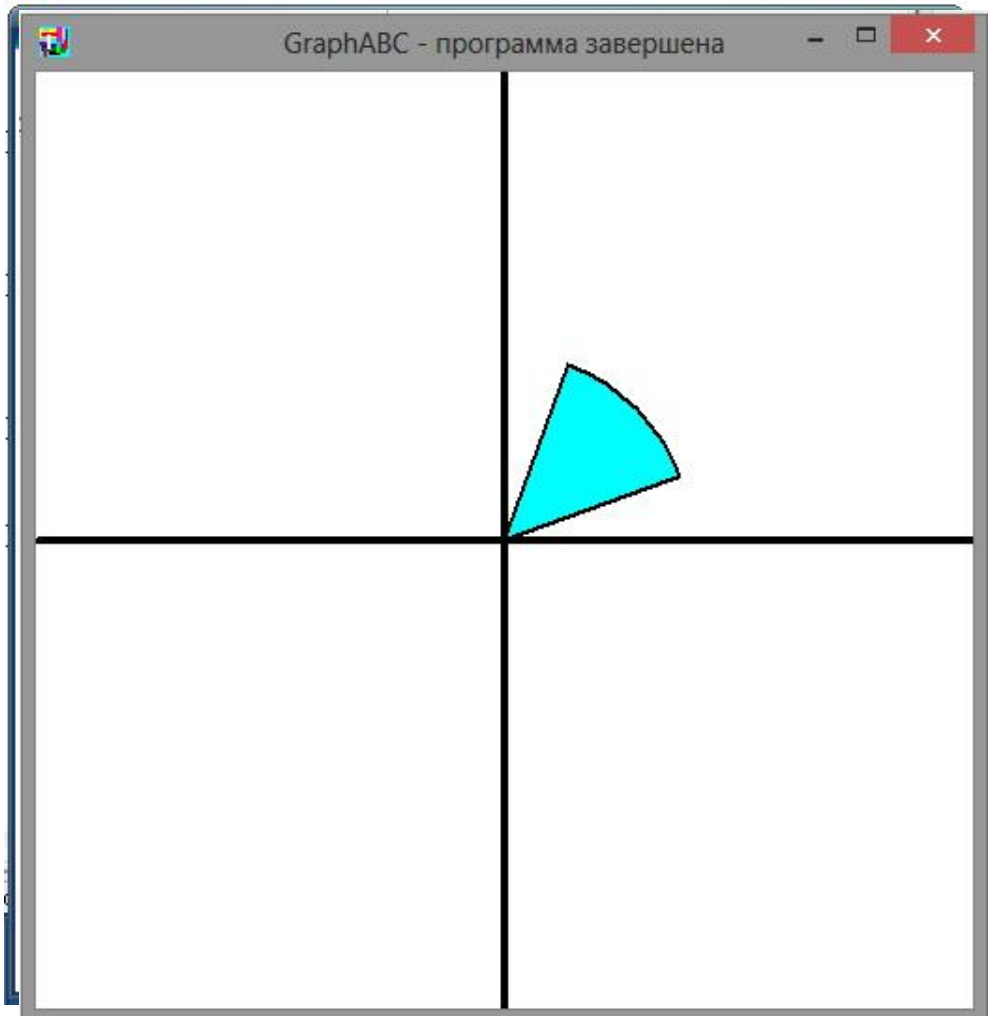
Пунктирная линия

SetPenStyle(<номер от 1 до 6>); – устанавливает стиль пера, задаваемый номером.




```
program prim;
uses GraphABC;
begin
  Setpencolor(clred);
  SetPenStyle(1);  {1 - длинный штрих}
  Line(10,100,350,100);
  SetPenStyle(2);  {2 - короткий штрих}
   Line(10,125,350,125);
   SetPenStyle(3);  {3 - штрих-пунктир}
   Line(10,150,350,150);
end.
```

Создать программу с использованием процедуры сектор и заливка цветом.



Random в Графике

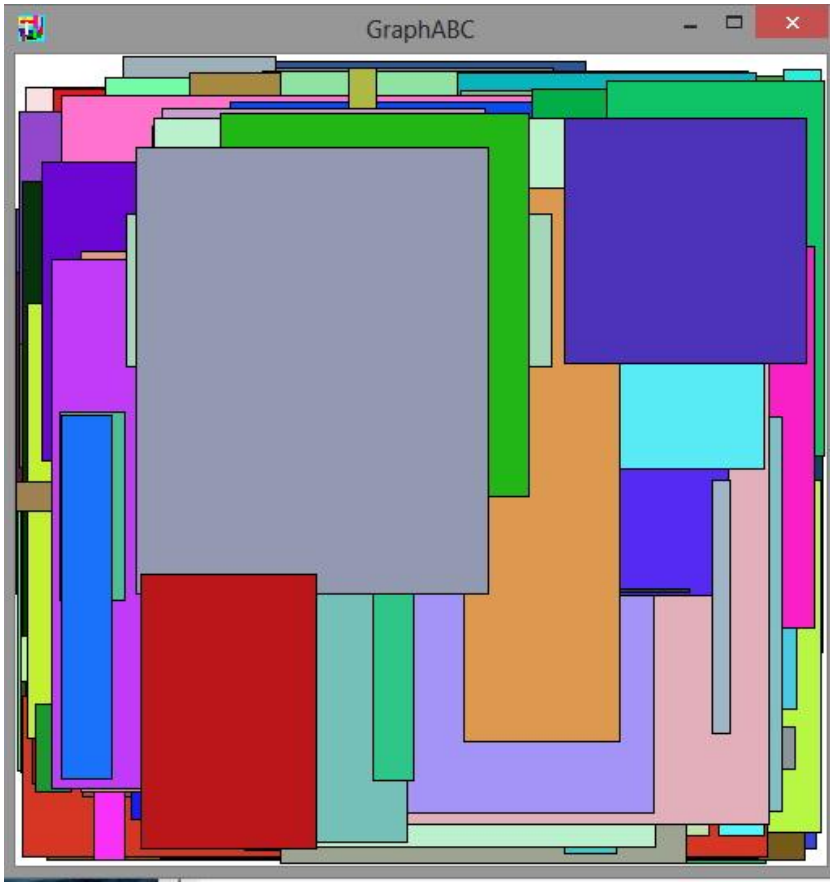
SetBrushColor(clRandom); - случайный выбор цвета

Rectangle(Random(WindowWidth),Random(WindowHeight),
Random(WindowWidth),
Random(WindowHeight)); случайные координаты для построения прямоугольника

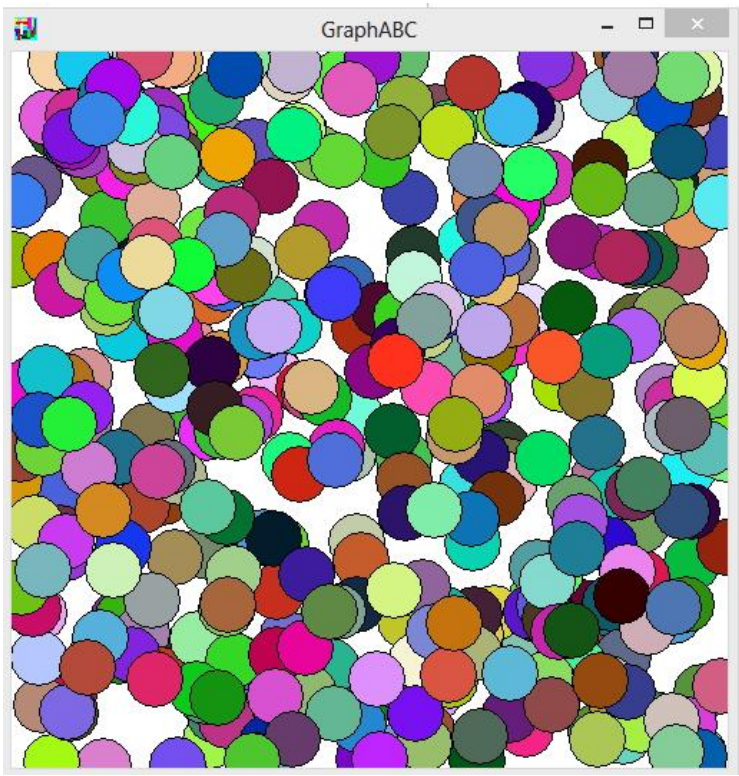
circle (Random(WindowWidth),Random(WindowHeight),
10); случайное построение окружности

sleep(1); **остановка программы миллисекунды**

Создать программу «Прямоугольники»

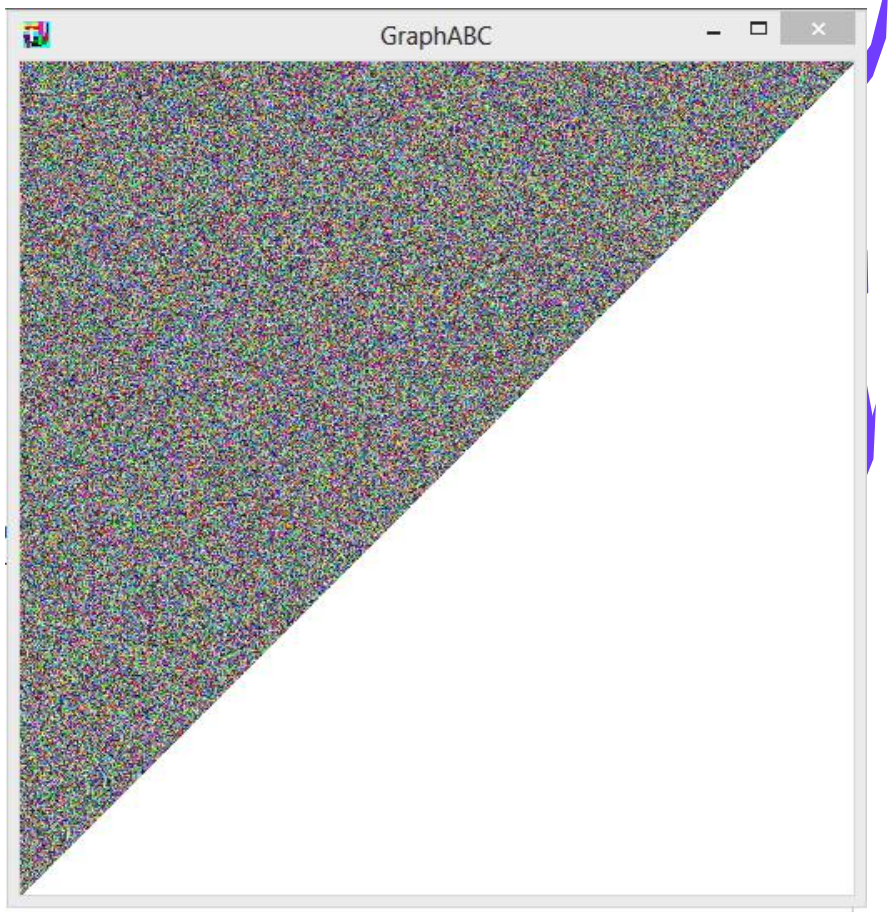


Создать программу «Конфетти»



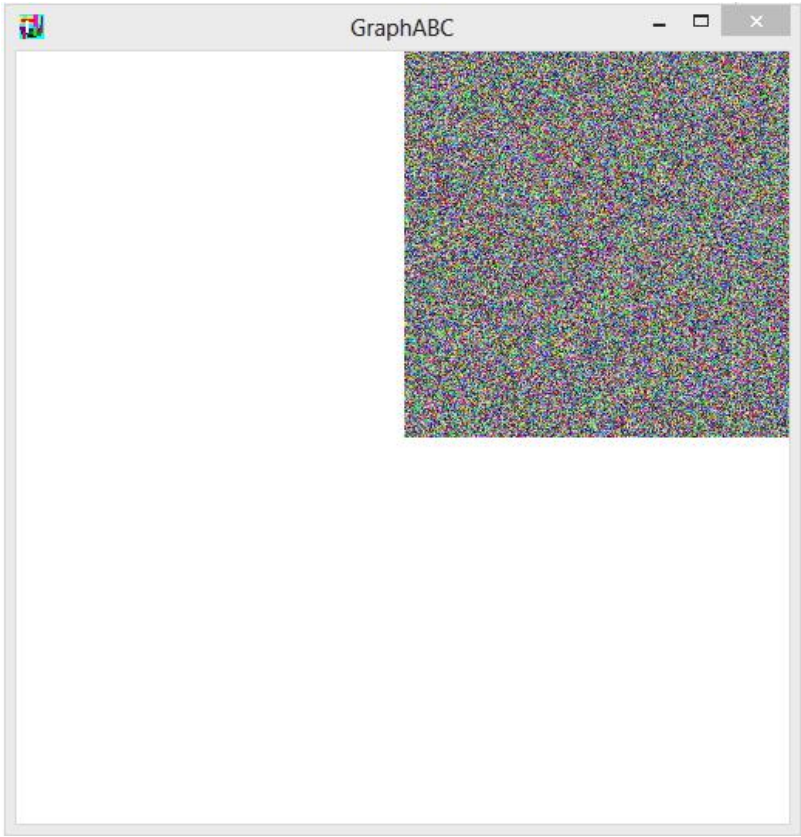
Создать программы заливки заданной области

```
begin
SetWindowHeight(500);
 SetWindowWidth(500);
 for j:=0 to
WindowHeight do
 for i:=0 to
WindowWidth-j do
    SetPixel(i,j ,clRandom);
 end.
```



Создать программы заливки заданной области

```
uses GraphABC;
var i,j: integer;
begin
SetWindowHeight(500);
 SetWindowWidth(500);
 for i:=0 to WindowHeight do
  for j:=0 to WindowWidth do
  If (i<250)and(j>250) then
    SetPixel(j,i,clRandom);
      Sleep(500);
     ClearWindow;
 end.
```



Создать программы заливки заданной области

```
uses GraphABC;
var i,j: integer;
begin
SetWindowHeight(500);
 SetWindowWidth(500);
 for i:=0 to WindowHeight do
  for j:=0 to WindowWidth do
   If
(i>=200)and(i<=300)and(j>=200)and
(j<=300) then
    SetPixel(j,i,clRandom);
      Sleep(50000);
     ClearWindow;
 end.
```

Пусть окружность с радиусом R1 размещена в обычной системе координат с центром в точке (0,0)

Движущаяся по окружности т. A определяет угол φ, образованный радиусом OA и осью OX.

Т.о. координаты т.A в любом месте на окружности имеют вид:

X=R1*cos φ

Y=R1*sin φ

Если центр окружности находится в центре экрана – в т. (300,300), то

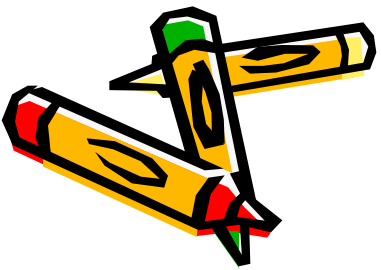
X=300+R1cos φ

Y=300-R1sin φ

Угол φ должен изменятся в цикле от 0 до 360

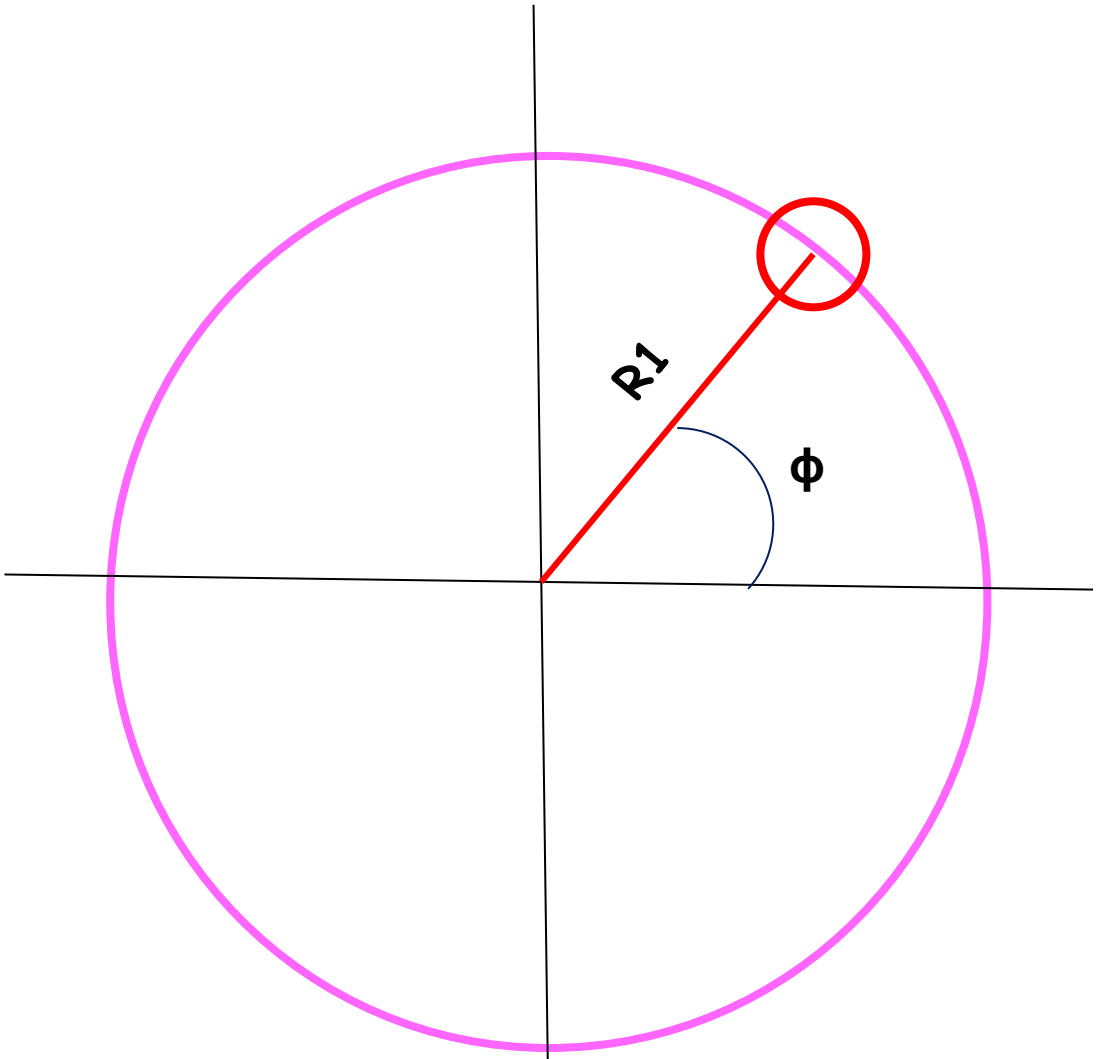
φ= **I * n** где **n** – число точек на окружности, **i** – счетчик цикла (от **0** до 360)

N = π/ угол от 0 до 360



circle(round(x), round(y) ,30);

Round (5.8)		**округление**




```
x:= 300+r*sin(i*pi/180);
    y:= 300+r*cos(i*pi/180);
    SetBrushColor(clred );
      circle(round(x),round(y),30 );
```

```
program Circle1;
uses graphabc;
var x:integer;
begin
  SetWindowWidth(600);
   SetWindowHeight(600);

for X:=25 to 600 do
    begin
         SetBrushColor(clred
        Circle(x,240,50);
        sleep(1);
        SetPenColor(clWhite)
        SetBrushColor(clWhit
);
        Circle(x,240,50);
end.
```

Подключаем модули CRT и Gfaph

Определяем начало и конец

Важно!
Координаты окружности при рисовании и стирании должны быть одинаковыми, иначе получим «трубу», а не движущуюся окружность

Закрашиваем объект цветом фона (стираем)

Допустим требуется построить график функции $y= x^2-3$ на отрезке [-3,3].

Договоримся располагать начало системы координат Oxy в середине экрана (т.е. в точке, определённой парой чисел 300, 300).

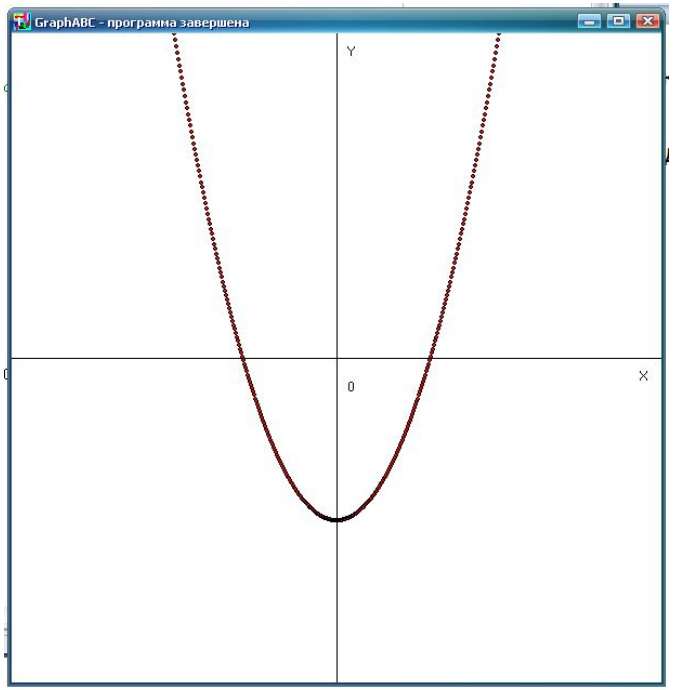
Необходимо условиться ещё и о количестве точек экрана, соответствующих единице измерения в системе координат Oxy, т.е. о масштабном множителе. Пусть его значение равно 50.

В этом случае положение точки графика с координатами (x,y) на экране определяется парой значений 300+50*x, 300-50*y.

I = от-150 до 150

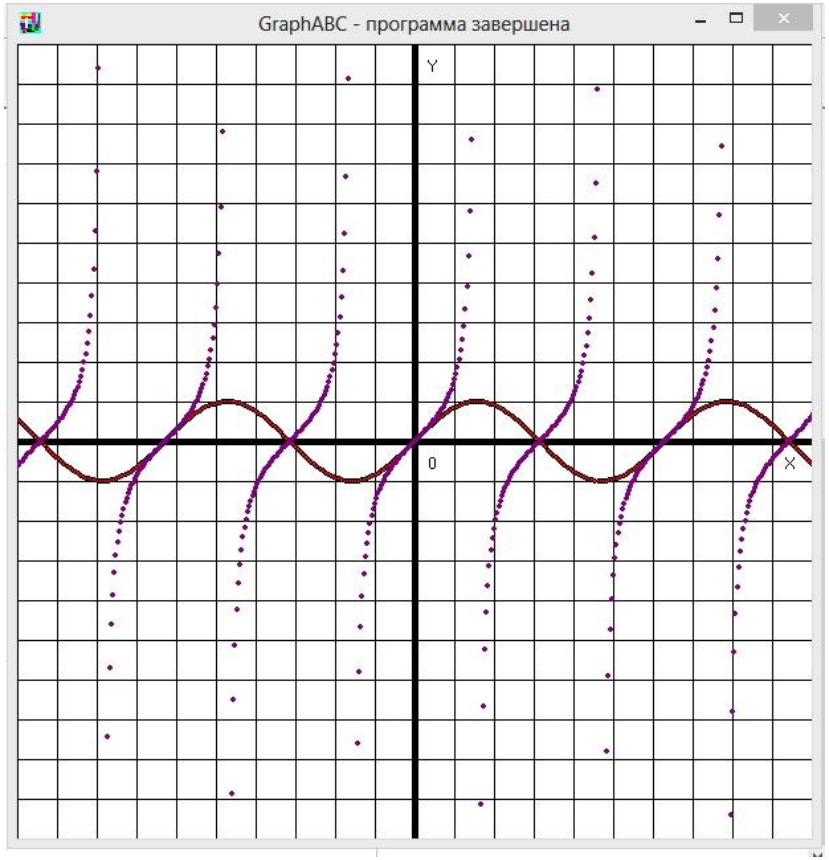
x:=0.02*i; выражаем X тогда y :=x*x-3;

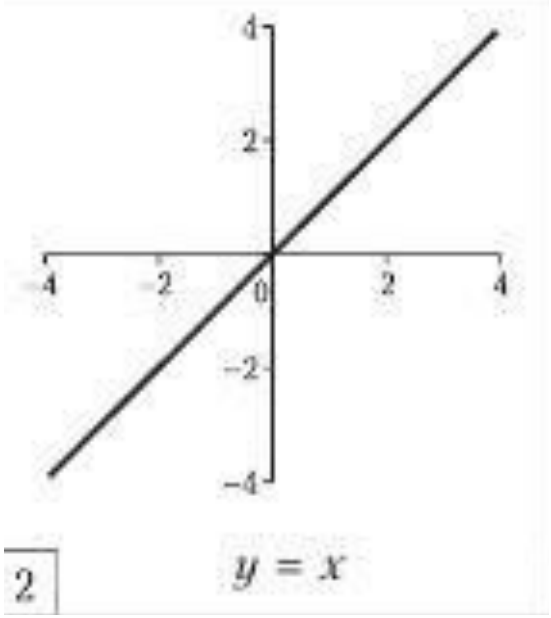
Round (5.8) - округление = 6



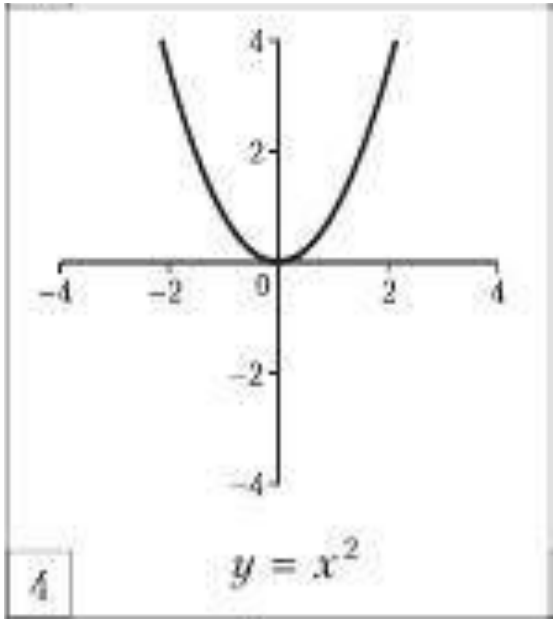
Cos(X), Sin(X) косинус (синус) числа X, где X - угол в радианах.

Функций тангенс и котангенс в Паскале нет. Для их вычисления используйте выражение **tg=sin(x)/cos(x)** или **ctg=cos(x)/sin(x)** для котангенса).

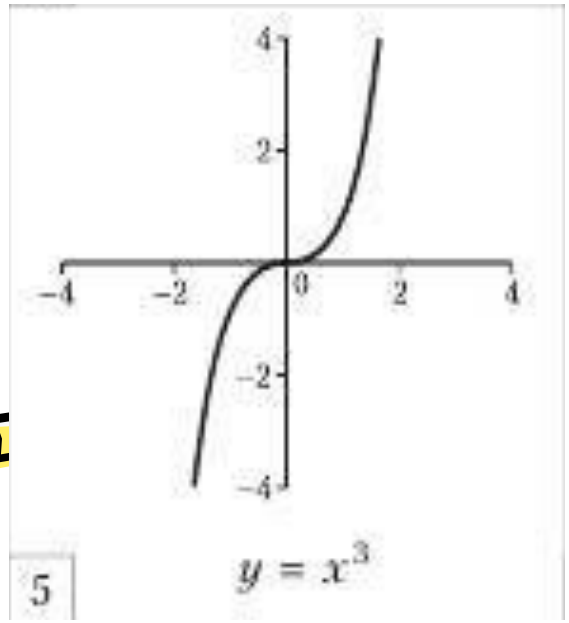




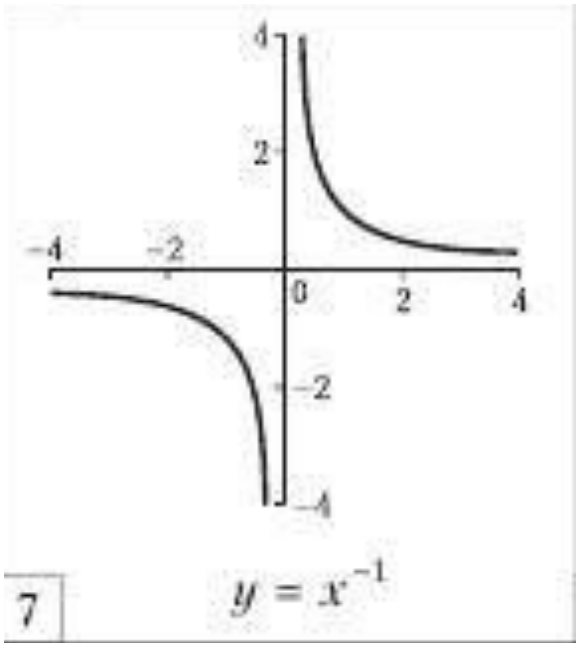
2 $y = x$



4 $y = x^2$



5 $y = x^3$



7 $y = x^{-1}$

Задание на урок:

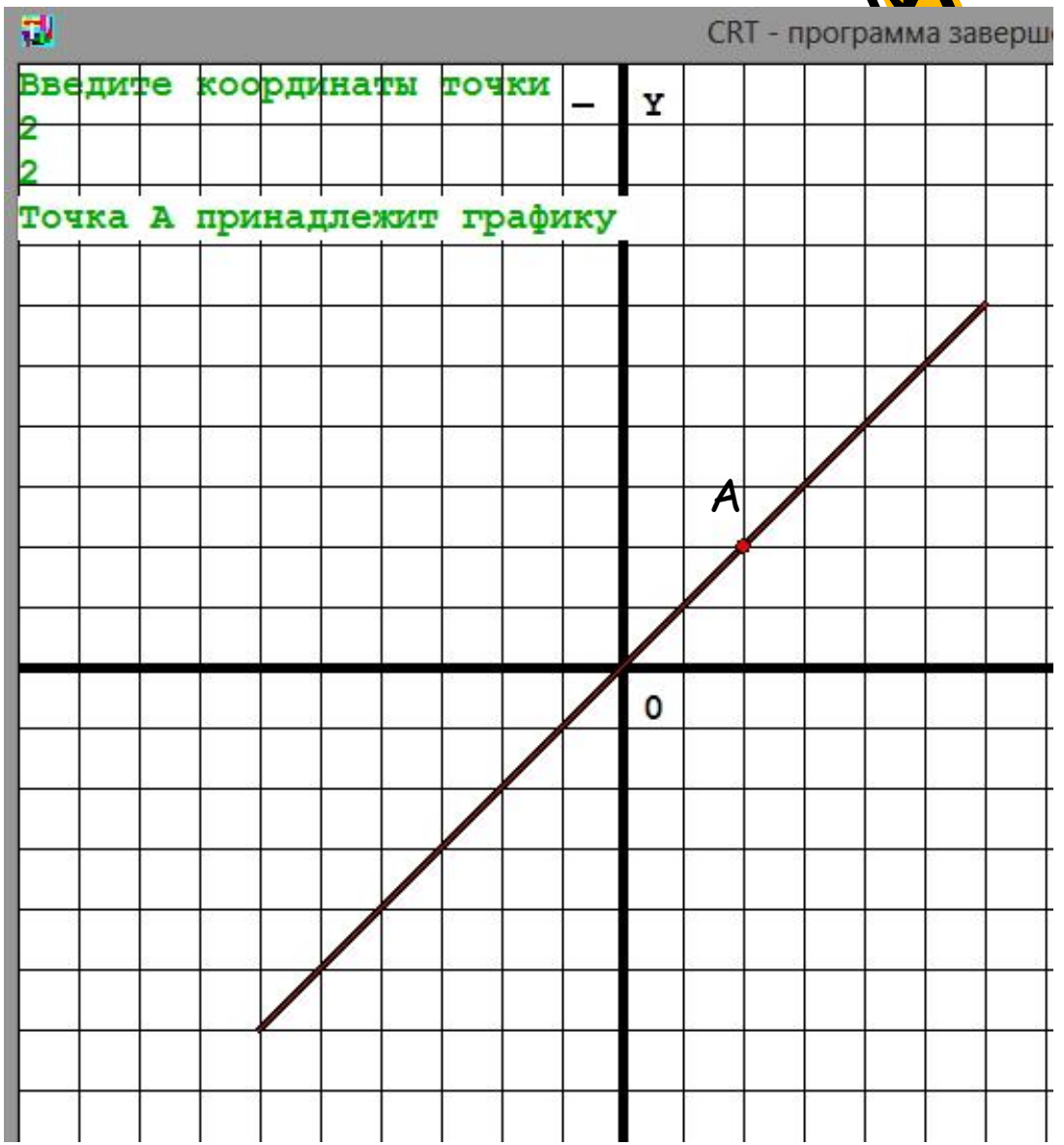
Нарисовать:

- Шарик, скатывающийся по ступенькам
- Луч, движущийся по кругу
- График функции x^2

Определение принадлежности точки графику функции

При работе с изображениями в среде PascalABC удобно все данные вводить и выводить в графическом окне. Совмещать работу с текстом и графикой в одном окне можно, подключив модули crt и GraphABC одновременно.

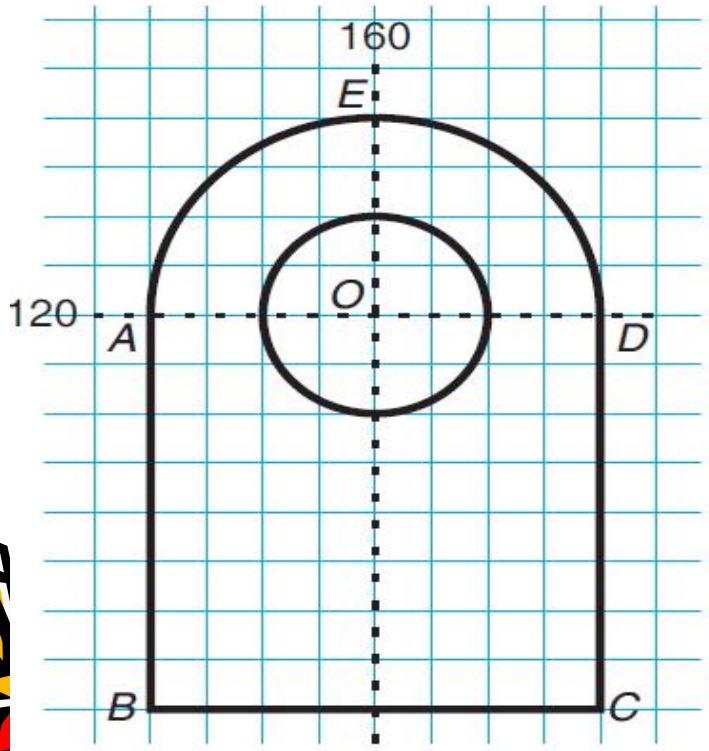
```
uses crt, GraphABC;
     var a,i,r,s:integer;
         x,y:real;
```

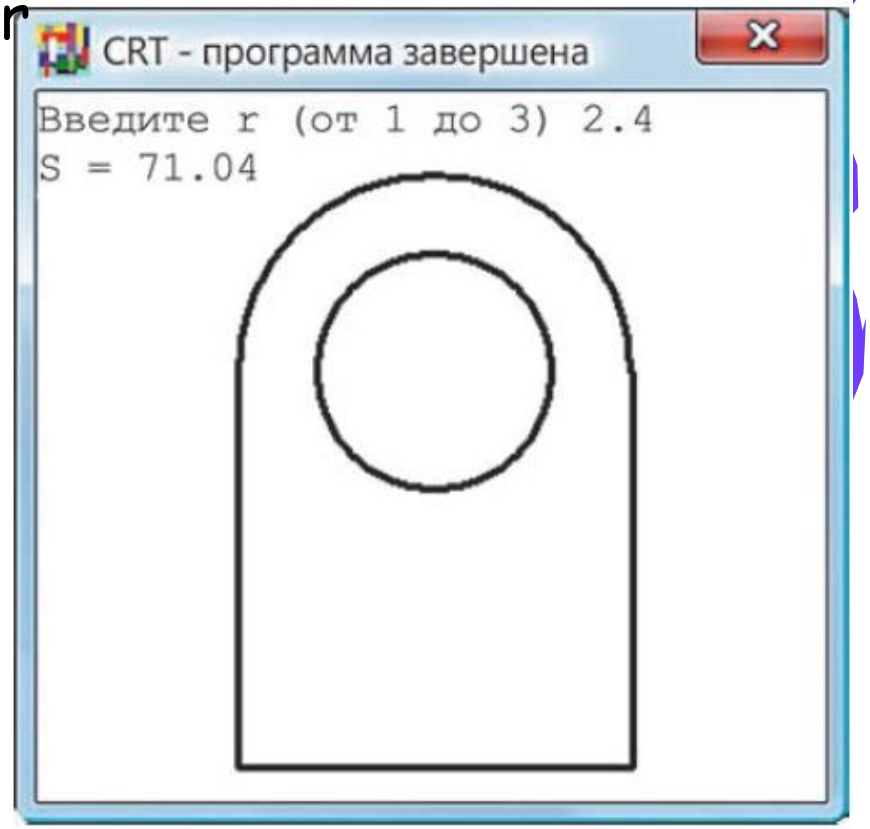


Составить программу, которая выполняет чертеж детали (рис. 2.1) и вычисляет ее площадь. Масштаб: 1 клетка — 20 пикселей. Радиус отверстия r в сантиметрах вводится с клавиатуры, AB = BC = 8 см, радиус полукруга =4см

Площадь детали складывается из площади квадрата *ABCD со стороной a и*
площади полукруга *AED диаметром a за вычетом площади круга радиусом r.*

```
S:=a*a+Pi*r1*r1/2-Pi*r*r
```

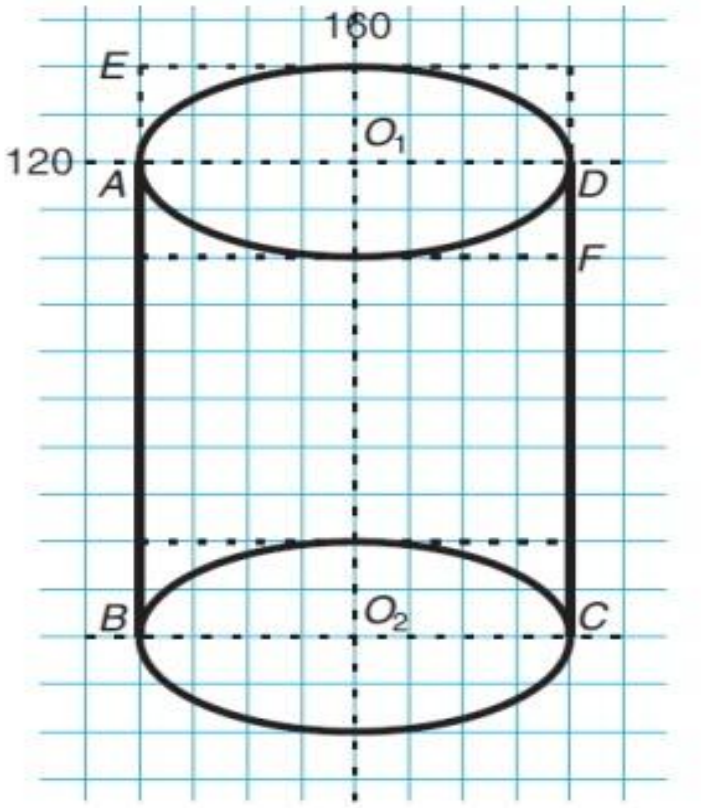




```
program Figura1;
uses crt, GraphABC; {Подключение модулей}
var r, a, S: real;
begin
SetWindowSize(320,320); {Размеры окна}
write('Введите r (от 1 до 3) ');
read(r); {Ввод радиуса}
SetPenWidth(3); {Толщина пера}
line(80,120, 80,280); {Рисование отрезков}
line(80,280, 240,280);
line(240,280, 240,120);
circle(160,120, trunc(r*20)); {Рисование окружности}
arc(160,120, 80, 0,180); {Рисование дуги}
a:=8; S:=a*a+Pi*r1*r1/2-Pi*r*r; {Вычисление площади}
write('S = ', S:2:2); {Форматный вывод}
end.
```

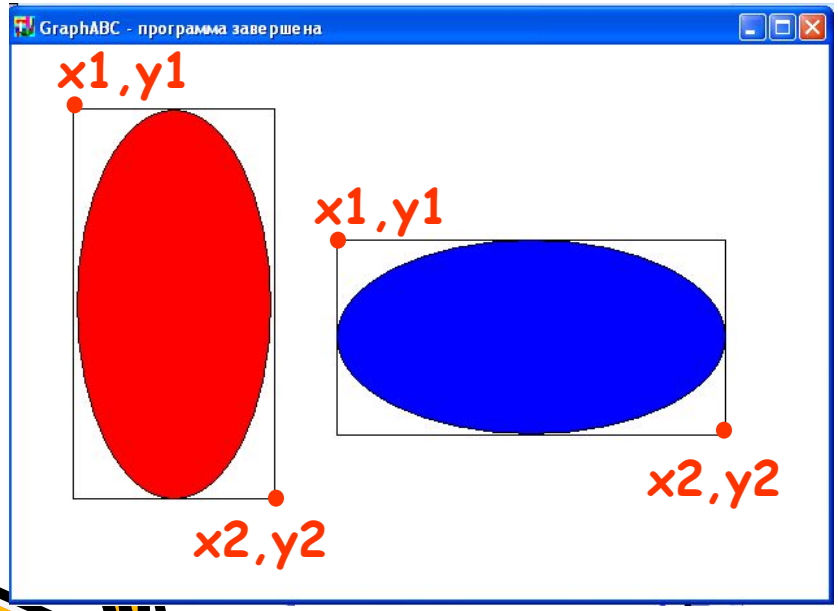
Составить программу, которая рисует цилиндр и вычисляет его объем. Масштаб: 1 клетка — 20 пикселей. Радиус цилиндра — 4 см. Высота цилиндра в сантиметрах вводится с клавиатуры

Вычислим объем цилиндра, как произведение площади основания (круга) и высоты $V = \pi r^2 h$.



Эллипс

Ellipse(x1,y1,x2,y2) – рисует эллипс, заданный своим описанным прямоугольником с координатами противоположных вершин (x1,y1) и (x2,y2).




```
Program oval;
uses GraphABC;
begin
  Ellipse(50,50,200,350);
  FloodFill(50+100,50+100,clred);
  Ellipse(250,150,550,300);
  FloodFill(250+100,150+100,clBlue);
end.
```

Объём цилиндра

```
uses crt, GraphABC;
var x, y, h, r: integer; V: real;
begin
SetWindowSize(320,320); {Размеры окна}
write('Введите высоту (1-8)  = ');
readln(h); {Ввод высоты}
write('Введите радиус (1-6)  = ');
 readln(r);
 V:=Pi*r*r*h; {Вычисление}
SetPenWidth(3); {Толщина пера}
SetBrushStyle(bsClear); {Стиль пера}
x:=160; y:=100; {Экранные координаты}
r:=20*r; h:=20*h; {Учет масштаба}
Ellipse(x-r,y-40, x+r,y+40); {Рисование эллипсов}
Ellipse(x-r,y+h-40, x+r,y+h+40);
line(x-r,y, x-r,y+h); {Рисование отрезков}
line(x+r,y, x+r,y+h);
writeln('Обьем равен = ',V);
end.
```

Создать программу « **Определение точек пересечения графиков двух функций»**

У=-2*х+4

У1=4*х-2