

# Графика

## модуль GraphABC

По материалам презентации  
<http://videouroki.net/filecom.php?fileid=98656904>

# 1. Подключение дополнительных библиотек

---

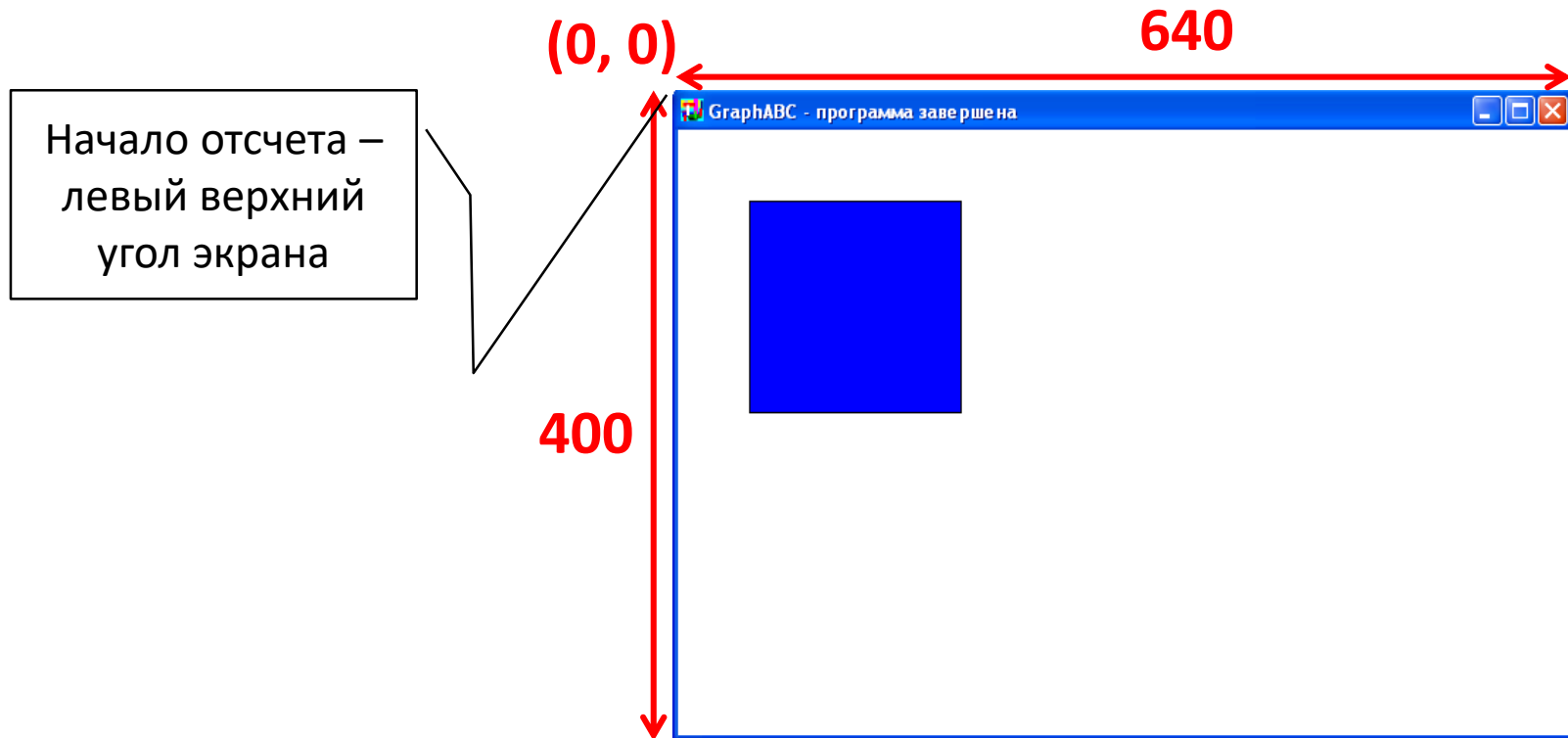
Для работы в графическом режиме необходимо подключение модуля **GraphABC**.

```
Program ... ;  
  Uses GraphABC ;  
  Var ... ;  
  Begin  
    ...  
  End.
```

## 2. Графический режим

---

Графический экран PascalABC (по умолчанию) содержит **640 точек по горизонтали** и **400 точек по вертикали**.



### 3. Определение размера экрана

---

Устанавливает ширину графического окна

**SetWindowWidth(w)**

Устанавливает высоту графического окна

**SetWindowHeight(h)**

## 4. Очистка графического окна

---

Очистка графического окна:

заливка белым цветом

**ClearWindow;**

заливка выбранным цветом

**ClearWindow(<color>);**

```
Program Z01;  
  Uses GraphABC;  
  Begin  
    ClearWindow;  
    ClearWindow(c1Green) ;  
  End.
```

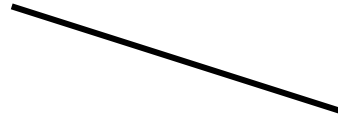
# 5. Графические примитивы

---

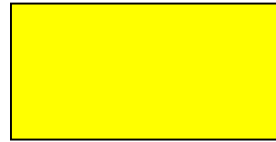
1.Точка



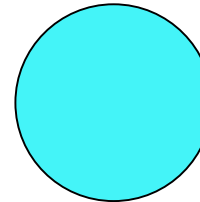
2.Линия



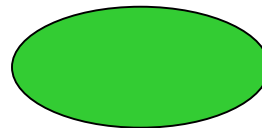
3.Прямоугольник



4.Окружность



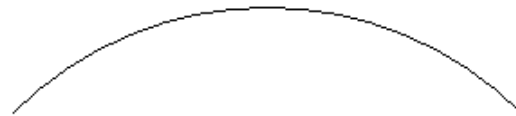
5.Эллипс



6.Сектор



7.Дуга



## 6. Цвета

---

**clBlack** – черный  
**clPurple** – фиолетовый  
**clWhite** – белый  
**clMaroon** – темно-красный  
**clRed** – красный  
**clNavy** – темно-синий  
**clGreen** – зеленый  
**clBrown** – коричневый  
**clBlue** – синий  
**clSkyBlue** – голубой

**clYellow** – желтый  
**clAqua** – бирюзовый  
**clOlive** – оливковый  
**clFuchsia** – сиреневый  
**clTeal** – сине-зеленый  
**clGray** – темно-серый  
**clLime** – ярко-зеленый  
**clLtGray** – светло-серый  
**clDkGray** – темно-серый  
**clMedGray** – серый  
**clSilver** – серебряный

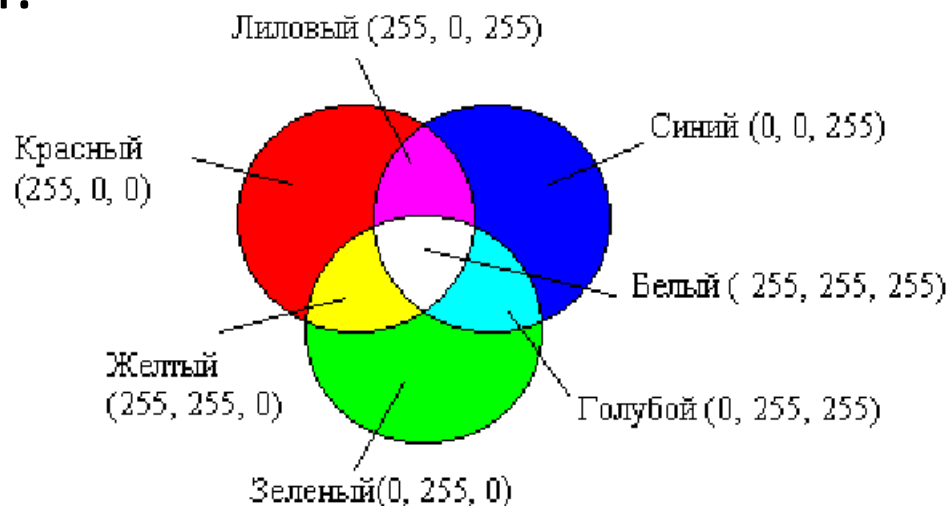
## 7. Задание цвета

---

Цвет можно задавать и с помощью функции **RGB(r,g,b)** где r, g и b – целые числа в диапазоне от 0 до 255 и определяют красную, зеленую и синюю составляющие.

RGB(255,255,255) – белый цвет.

RGB(0,0,0) – черный цвет.

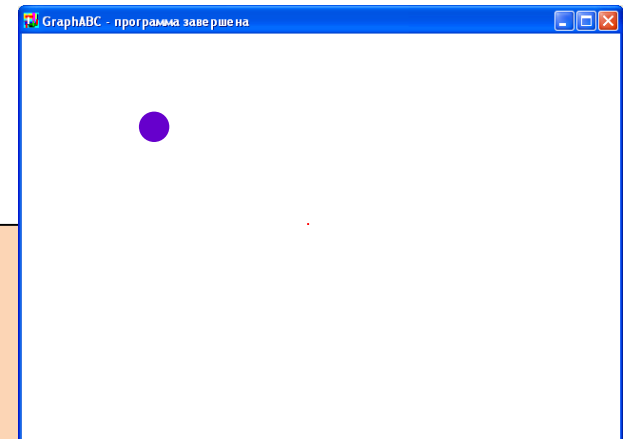




## 8. Точка

---

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



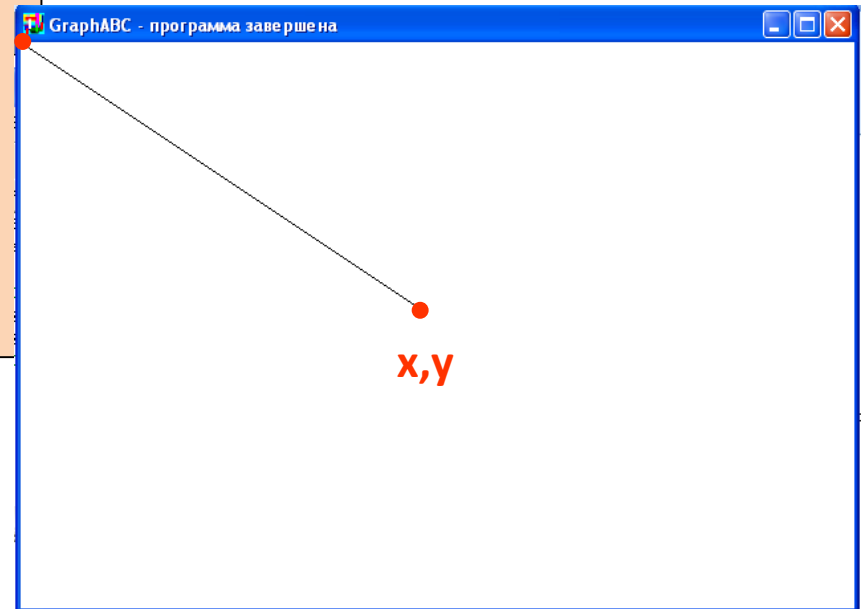
```
Program z02;  
Uses GraphABC;  
Begin  
  SetPixel(150,120, clBlue);  
  //SetPixel(150,120, RGB(0,0,255));  
End.
```

## 9. Линии

---

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

```
Program Z03;  
Uses GraphABC;  
Begin  
LineTo(300,200);  
End.
```

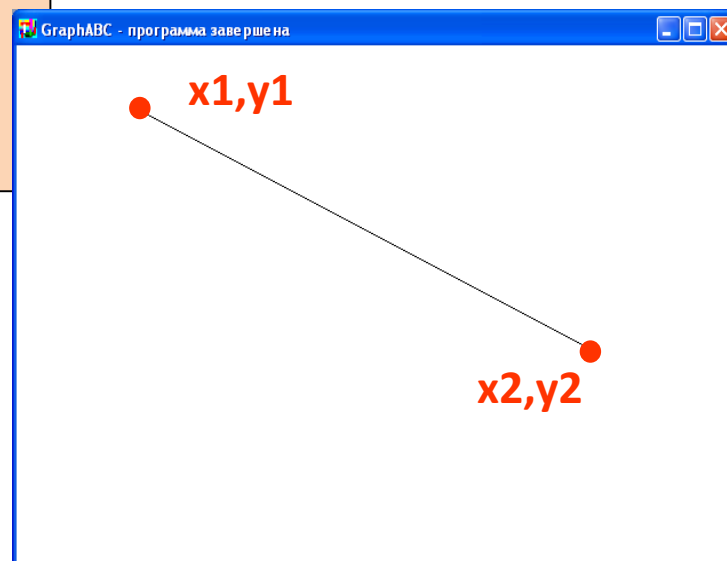


# 10. Линии

---

**Line(x1,y1,x2,y2)** - рисует отрезок с началом в точке (x1,y1) и концом в точке (x2,y2).

```
Program Z04;  
Uses GraphABC;  
Begin  
Line(100,50,500,250);  
End.
```

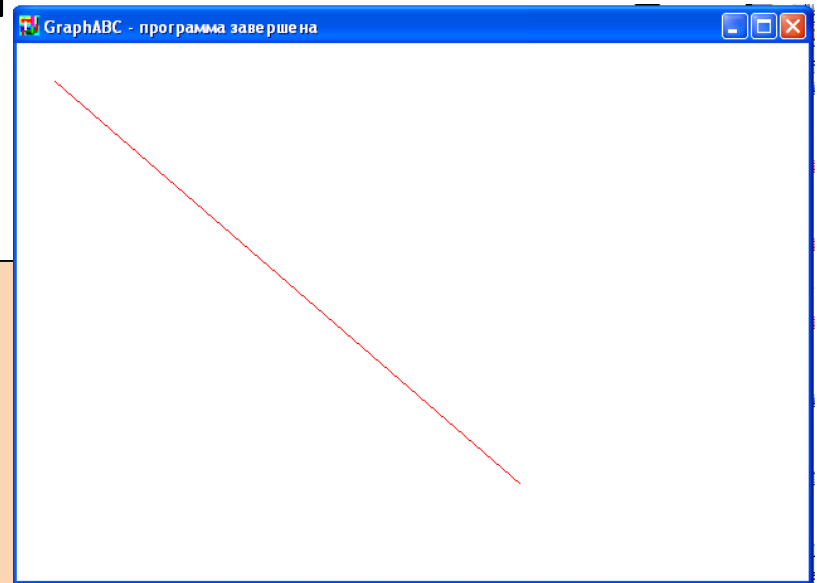


# 11. Цвет линии

---

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

```
Program Z05;  
Uses GraphABC;  
Begin  
SetPenColor(clred);  
//SetPenColor( RGB(255,0,0) );  
Line(30,30,400,350);  
End.
```

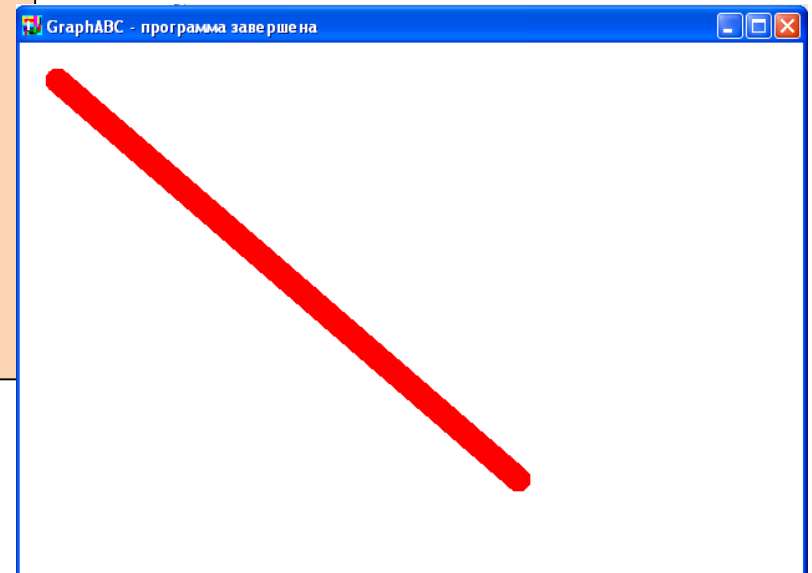


## 12. Толщина линии

---

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

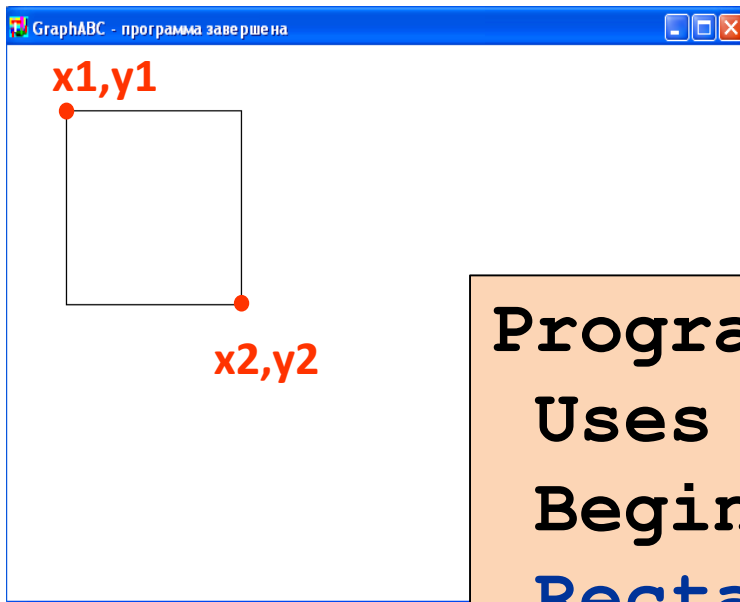
```
Program Z06;  
Uses GraphABC;  
Begin  
SetPenWidth(20);  
SetPenColor(clred);  
Line(30, 30, 400, 350);  
End.
```



# 13. Прямоугольник

---

**Rectangle(x1,y1,x2,y2)** - рисует прямоугольник, заданный координатами противоположных вершин (x1,y1) и (x2,y2).

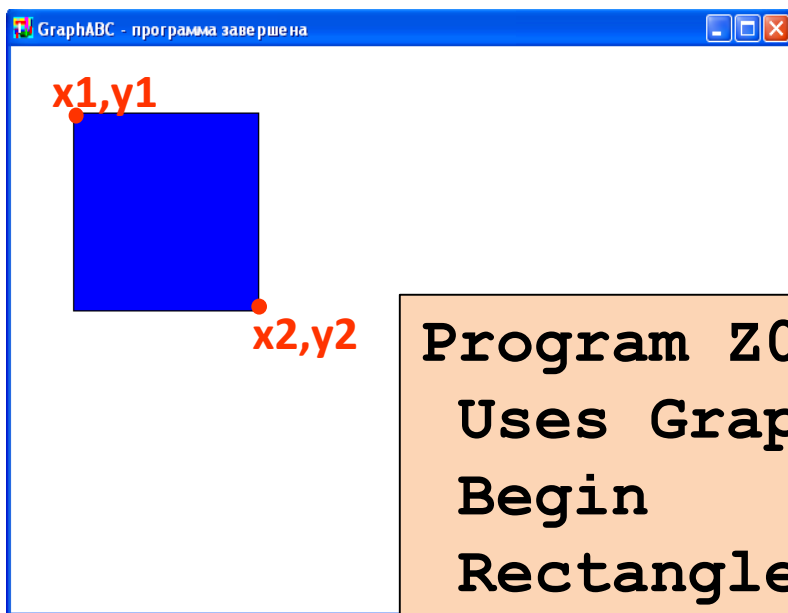


```
Program z07;  
Uses GraphABC;  
Begin  
Rectangle (50, 50, 200, 200);  
End.
```

# 14. Заливка цветом

---

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



```
Program Z08;  
Uses GraphABC;  
Begin  
Rectangle (50, 50, 200, 200) ;  
FloodFill (100, 100, clBlue) ;  
//FloodFill (100, 100, RGB (0, 0, 255) ) ;  
End.
```

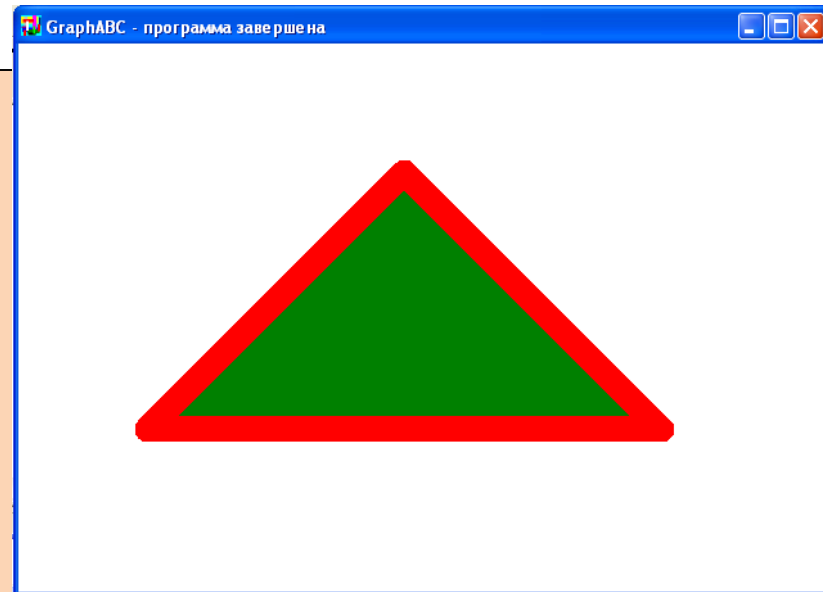
# 15. Треугольник

---

Рисуется процедурами

**Line(x1,y1,x2,y2); LineTo(x,y);**

```
Program Z09;  
Uses GraphABC;  
Begin  
  SetPenWidth(20);  
  SetPenColor(clRed);  
  Line(300,100,500,300);  
  Lineto(100,300);  
  Lineto(300,100);  
  FloodFill(300,200,clGreen);  
  //FloodFill(300,200,RGB(0,255,0));  
End.
```



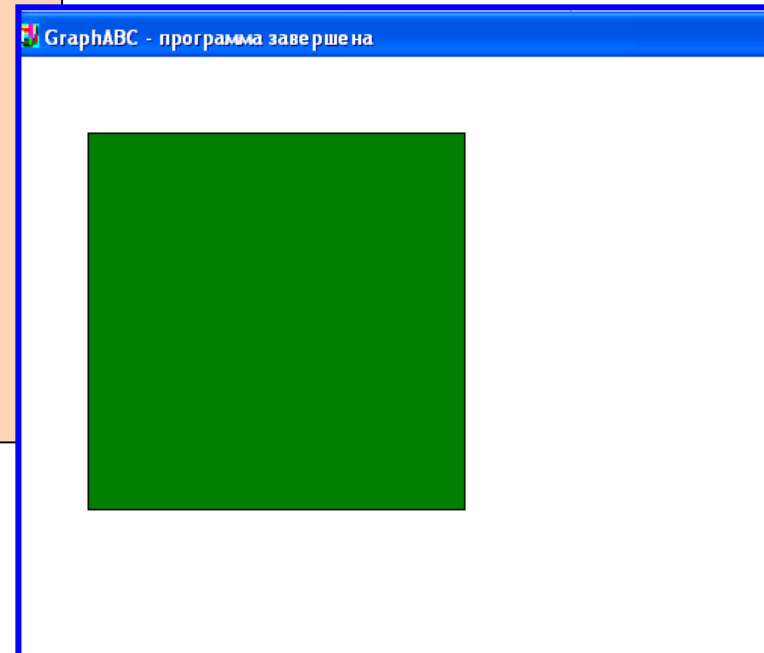


## 16. Заливка кистью

---

**SetBrushColor(color)** - устанавливает цвет кисти. Заливка кистью распространяется на замкнутый контур, описание которого следует за процедурой установки цвета кисти.

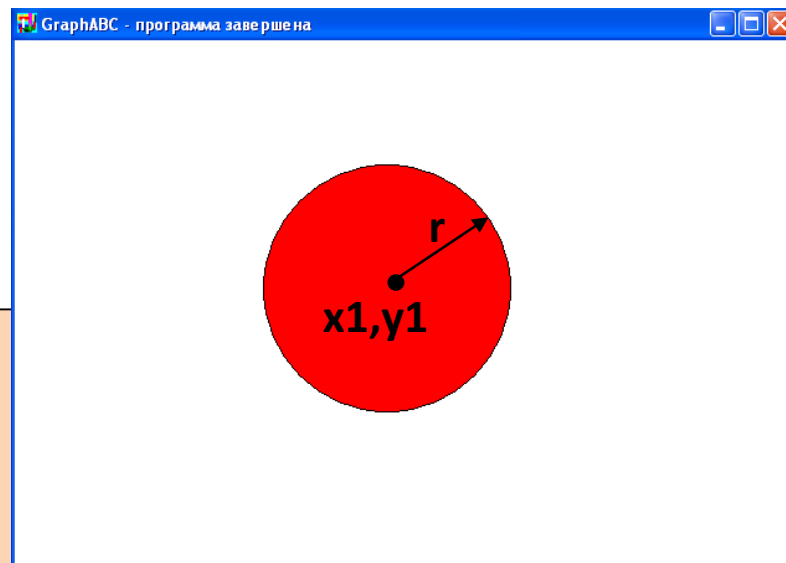
```
Program Z10;  
Uses GraphABC;  
Begin  
  SetBrushColor (clGreen) ;  
  Rectangle (50, 50, 300, 300) ;  
End.
```



# 17. Окружность

**Circle(x,y,r)** - рисует окружность с центром в точке (x,y) и радиусом r.

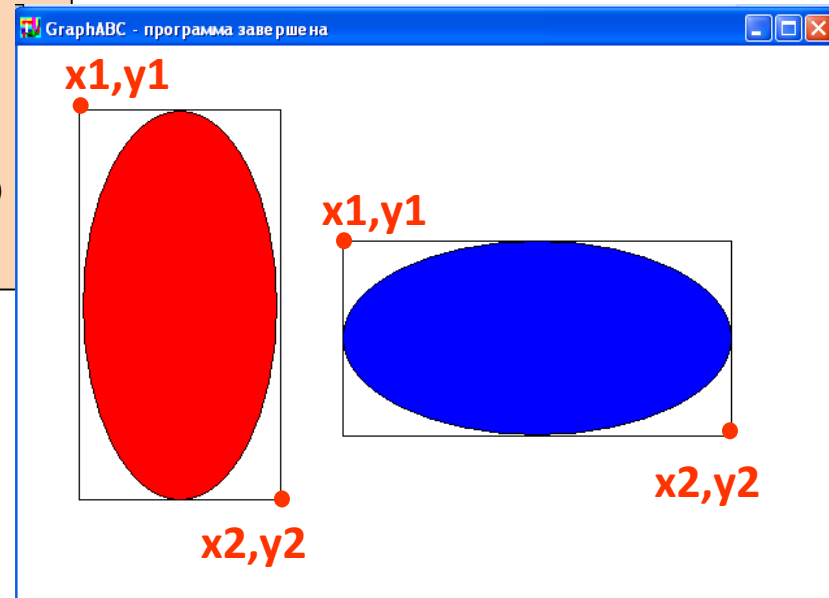
```
Program Z11;  
  Uses GraphABC;  
  Begin  
    Circle(500,200,100);  
    FloodFill(500,200,clRed);  
  End.
```



# 18. Эллипс

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

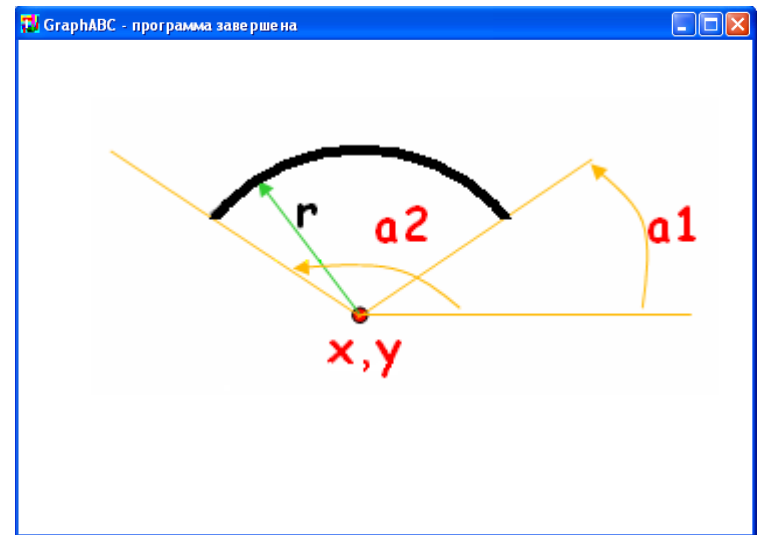
```
Program Z12;  
Uses GraphABC;  
Begin  
  Ellipse (50, 50, 200, 350) ;  
  FloodFill (50+100, 50+100, clred) ;  
  Ellipse (250, 150, 550, 300) ;  
  FloodFill (250+100, 150+100, clBlue)  
End.
```



# 19. Дуга окружности

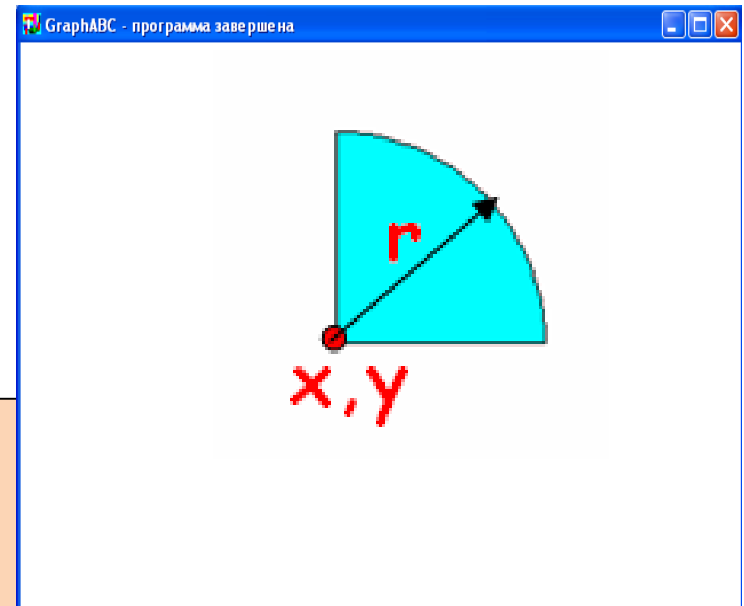
**Arc(x,y,r,a1,a2)** - Рисует дугу окружности с центром в точке (x,y) и радиусом r, заключенной между двумя лучами, образующими углы a1 и a2 с осью OX (a1 и a2 – вещественные, задаются в градусах и отсчитываются против часовой стрелки).

```
Program Z13;  
Uses GraphABC;  
Begin  
SetPenWidth(10);  
Arc(300,250,150,45,135);  
End.
```



## 20. Сектор

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



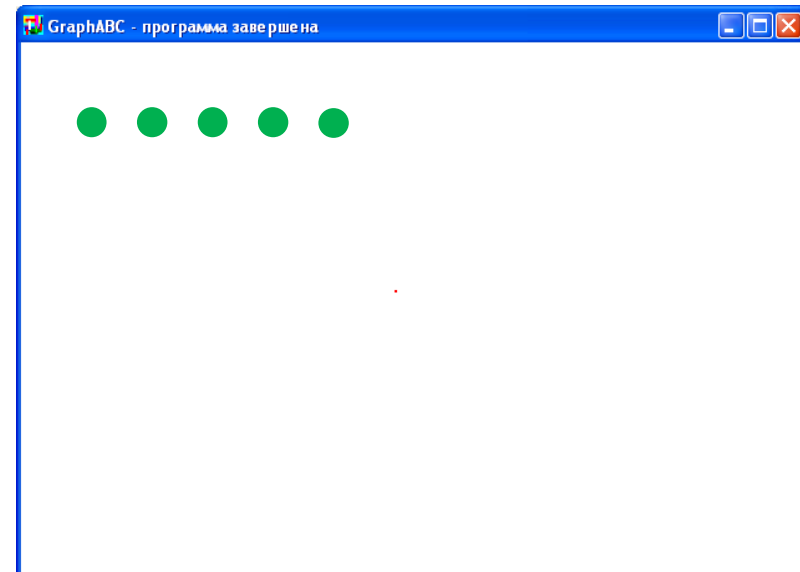
```
Program Z14;  
Uses GraphABC;  
Begin  
Pie(300,200,100,0,90);  
FloodFill(300+10,200-10,clAqua);  
End.
```

## 21. Пример программы

```
Program Z14;  
Uses GraphABC;  
Var I:Integer;  
Begin  
SetWindowWidth(700);  
SetWindowHeight(500);  
ClearWindow;  
For I:=1 to 5 do  
begin  
SetBrushColor(clGreen);  
Circle(I*40+10,30,10);  
Sleep(50);  
end;  
End.
```

5 окружностей,  
расположенных по  
горизонтали.

Окружности выводятся с  
задержкой (**Sleep(50)**).



## 22. Задания

---

1. Вывести на экран 10 окружностей разных цветов, расположенных по горизонтали.
2. Вывести на экран 10 окружностей разных цветов, расположенных по диагонали.
3. Вывести на экран 10 окружностей разных цветов, расположенных по вертикали.
4. Вывести на экран окружности по периметру прямоугольника (одна сторона прямоугольника состоит из 15 окружностей, другая – из 8 окружностей). При выводе окружностей использовать задержку.
5. Окружность «пробегает» по периметру прямоугольника.

## 23. Задания

---

1. Вывести на экран 6 прямоугольников размером 20x30 в одном ряду, а всего 8 рядов.
2. Вывести на экран следующие изображения:

