

# PASCAL ABC

## Модуль GraphABC

# ПОДКЛЮЧЕНИЕ ДОПОЛНИТЕЛЬНЫХ БИБЛИОТЕК

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

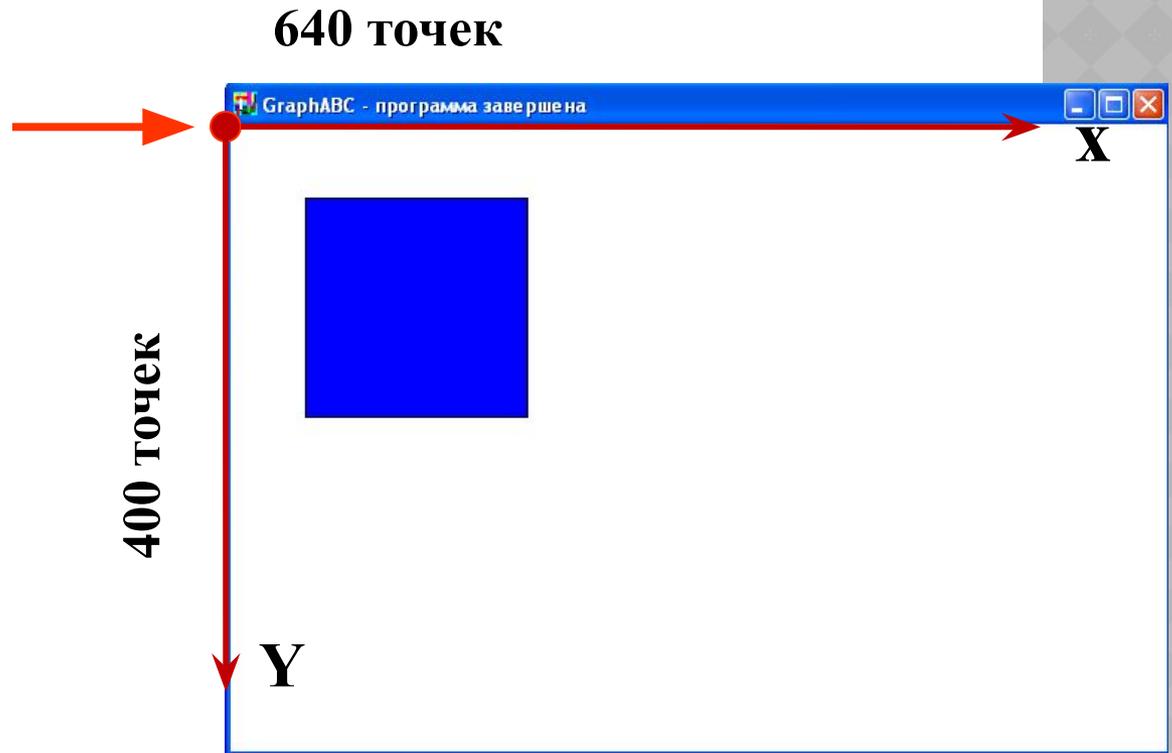
Первой инструкцией программы должна быть инструкция

**uses GraphABC;**

# Графический режим

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

**Начало отсчета** —  
**левый верхний**  
**угол экрана**

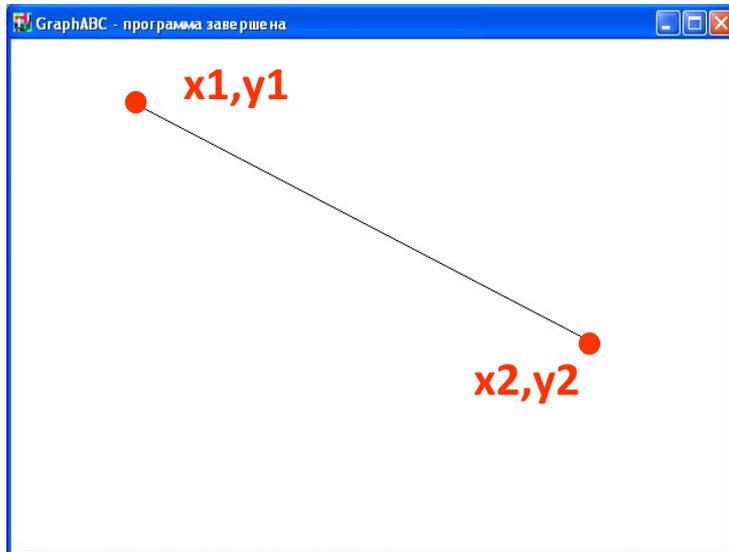


# Управление экраном

**SetWindowSize(x,y)** - Устанавливает ширину и высоту графического окна;

# Линии

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

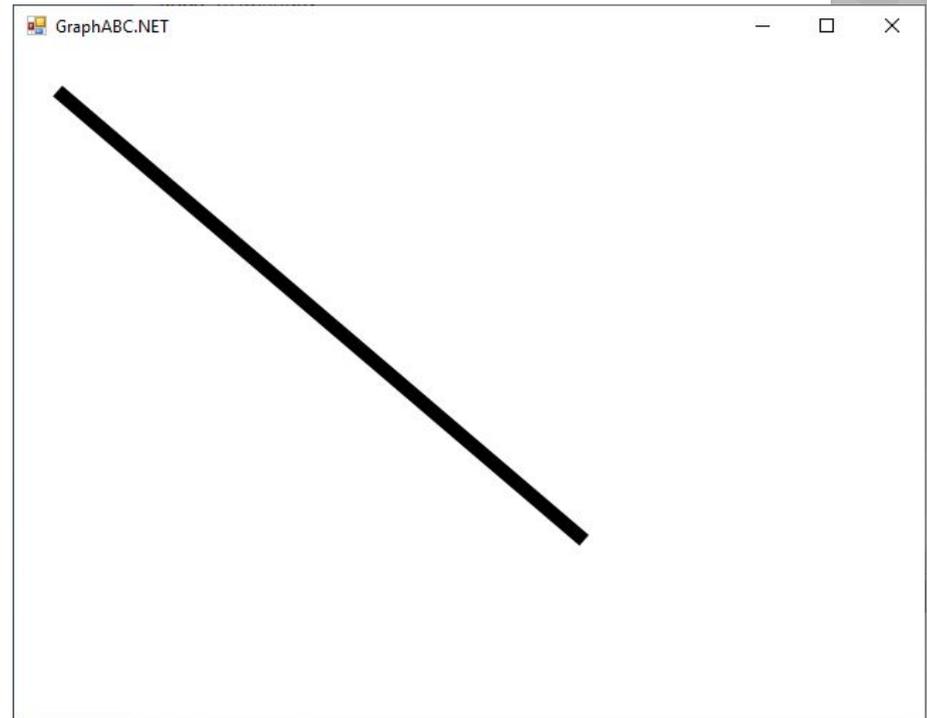


Program liniay;  
uses GraphABC;  
begin  
**line(100,50,500,250);**  
end.

# Толщина линии (пера)

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

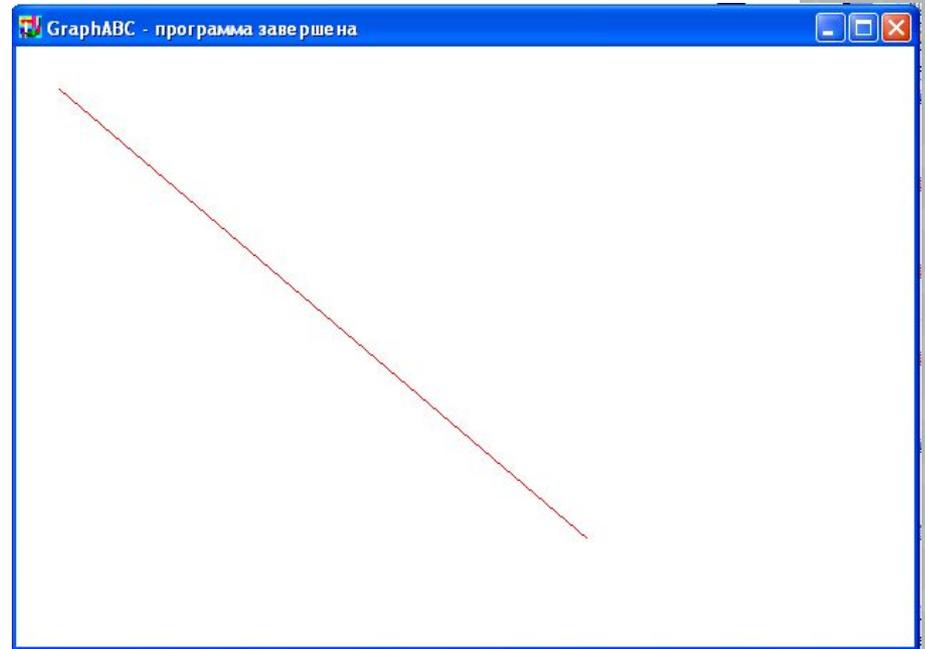
```
Program liniay;  
uses GraphABC;  
begin  
    setpenwidth(10);  
    line(30,30,400,350);  
end.
```



# Цвет линии (цвет пера)

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

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



## Цвет линии (способ 2)

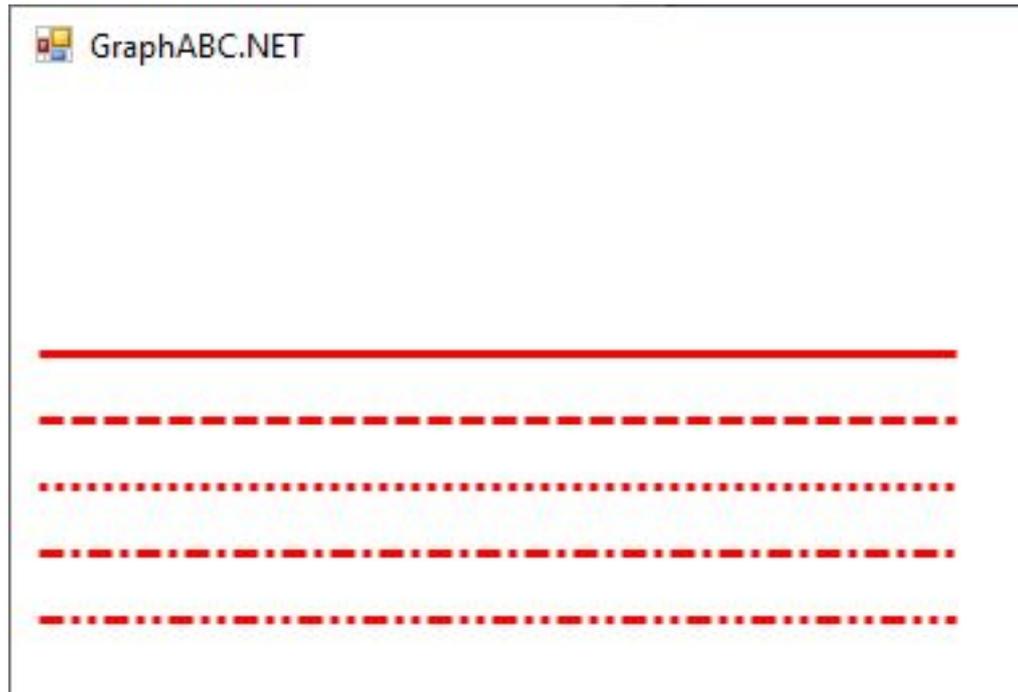
**Line(x1,y1,x2,y2, color)** - рисует отрезок от точки (x1,y1) до точки (x2,y2) цветом **color**.



```
program liniay2;  
uses GraphABC;  
begin  
    setpenwidth(10);  
    line(30,30,400,350,clred);  
end.
```

# Стиль линии

**SetPenStyle(стиль);** - устанавливает стиль пера (сплошной, пунктир и т.п.)



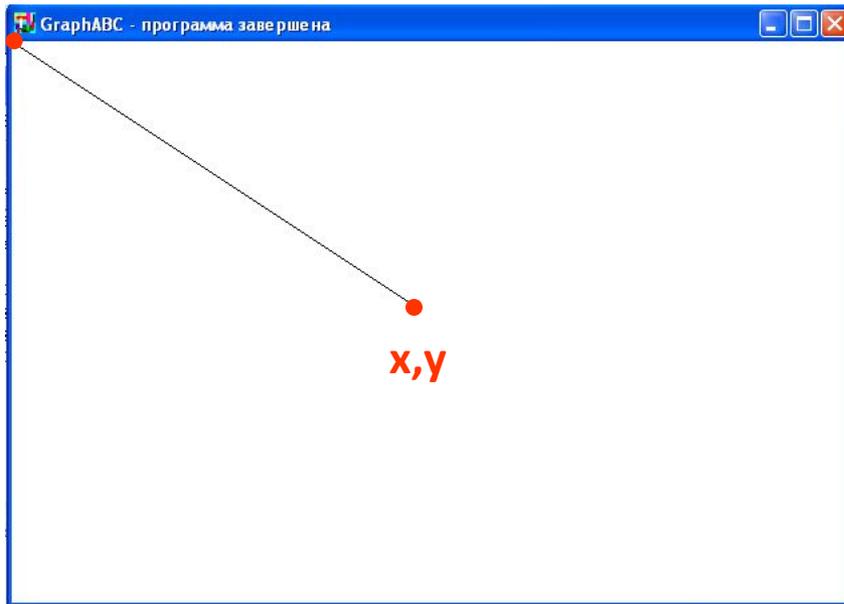
Текущее перо используется для рисования графических примитивов до его следующего изменения

# Стиль линии (пример программы)

```
program prim2;
uses GraphABC;
begin
Setpencolor(clred);
setpenwidth(3);
SetPenStyle(psSolid); //сплошное перо
Line(10,100,350,100);
SetPenStyle(psDash); //штриховое перо
Line(10,125,350,125);
SetPenStyle(psDot); //пунктирное перо
Line(10,150,350,150);
SetPenStyle(psDashDot); //штрихпунктирное перо
Line(10,175,350,175);
SetPenStyle(psDashDotDot); //альтернативное штрихпунктирное перо
Line(10,200,350,200);
end.
```

# Линии

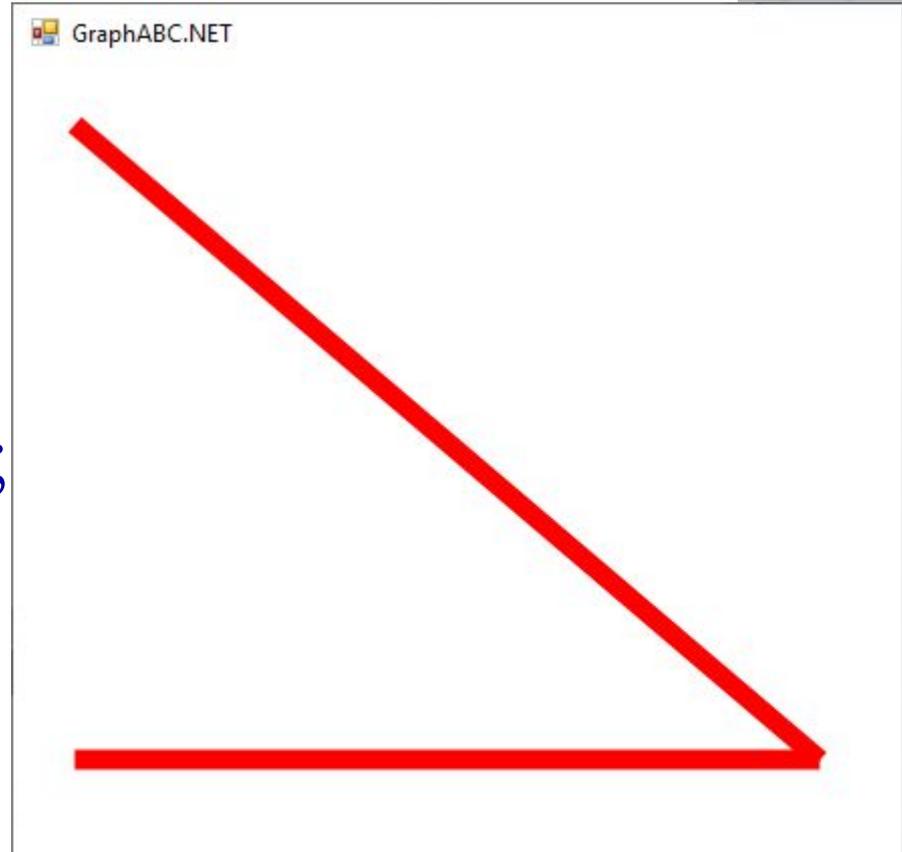
**LineTo(x,y)** - рисует отрезок от текущего положения пера (0,0) до точки (x,y); координаты пера при этом также становятся равными (x,y).



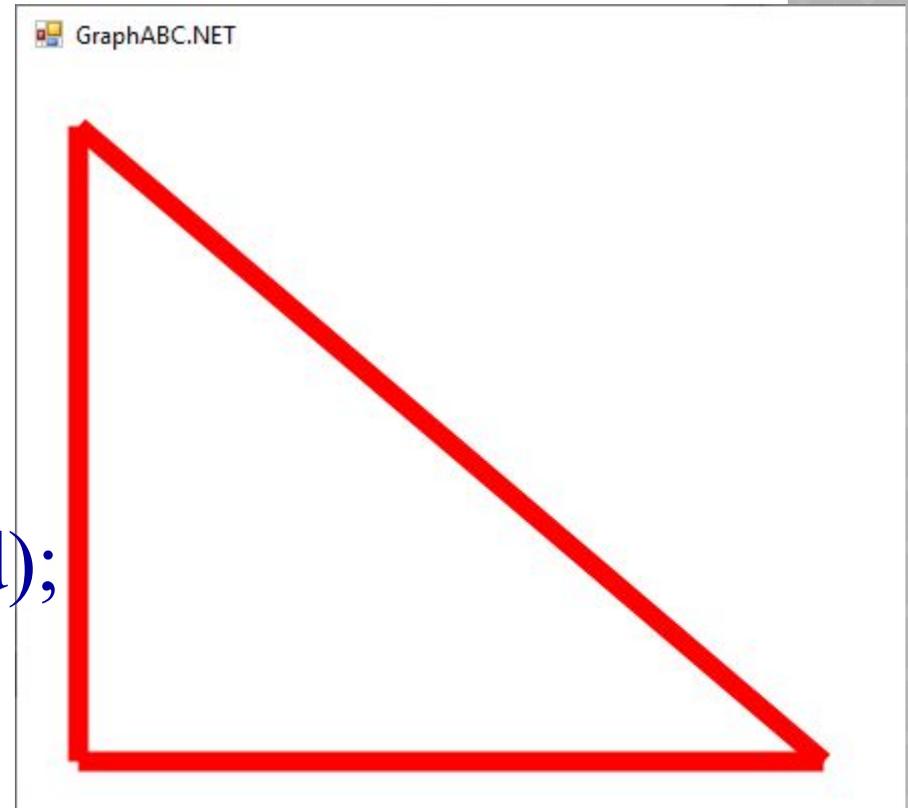
```
Program liniay;  
uses GraphABC;  
begin  
LineTo(300,200);  
end.
```

**MoveTo(x,y)** - устанавливает текущую позицию рисования в точку (x,y).

```
Program liniay3;  
uses GraphABC;  
begin  
    setpenwidth(10);  
    line(30,30,400,350,clred);  
    MoveTo(400,350);  
    lineto(30,350,clred);  
end.
```



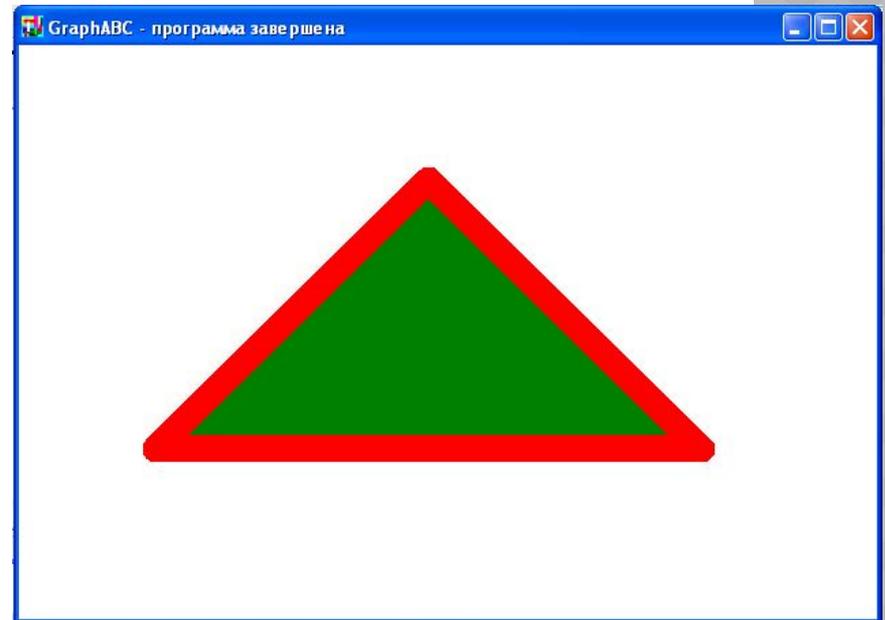
```
Program liniay3;  
uses GraphABC;  
begin  
    setpenwidth(10);  
    line(30,30,400,350,clred);  
    MoveTo(400,350);  
    lineto(30,350,clred);  
    lineto(30,30,clred);  
end.
```



# Треугольник

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

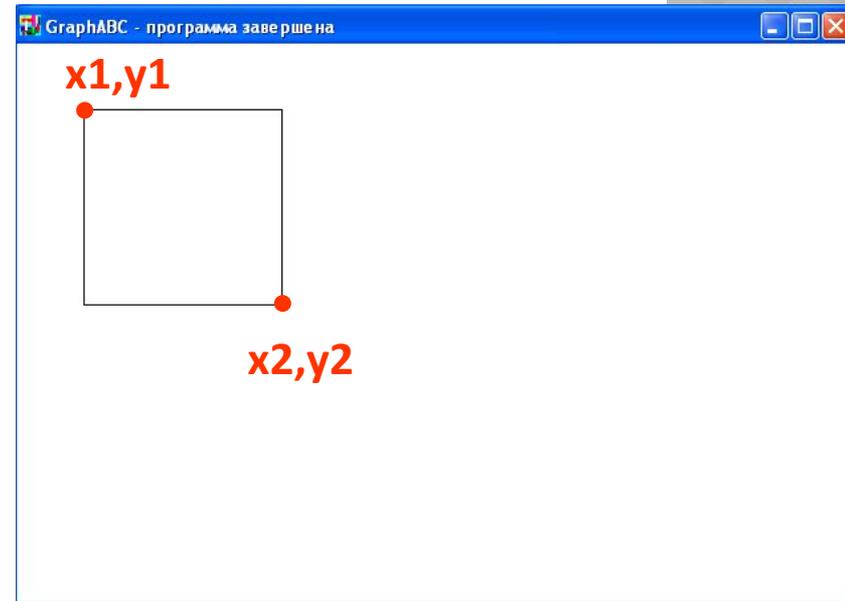
```
Program treugolnik;  
uses GraphABC;  
begin  
  setpenwidth(20);  
  setpencolor(clred);  
  line(300,100,500,300);  
  moveto(500,300);  
  lineto(100,300);  
  lineto(300,100);  
  floodfill(300,200,clgreen);  
end.
```



# Прямоугольник

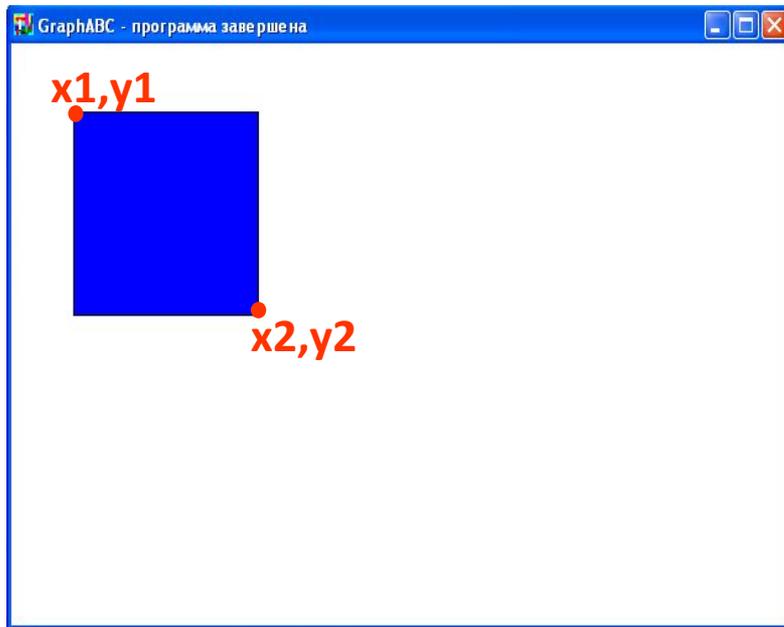
**Rectangle(x1,y1,x2,y2)** - рисует  
прямоугольник, заданный координатами  
противоположных вершин (x1,y1) и (x2,y2).

```
Program pryamougolnik;  
uses GraphABC;  
Begin  
Rectangle(50,50,200,200);  
end.
```



# Заливка цветом

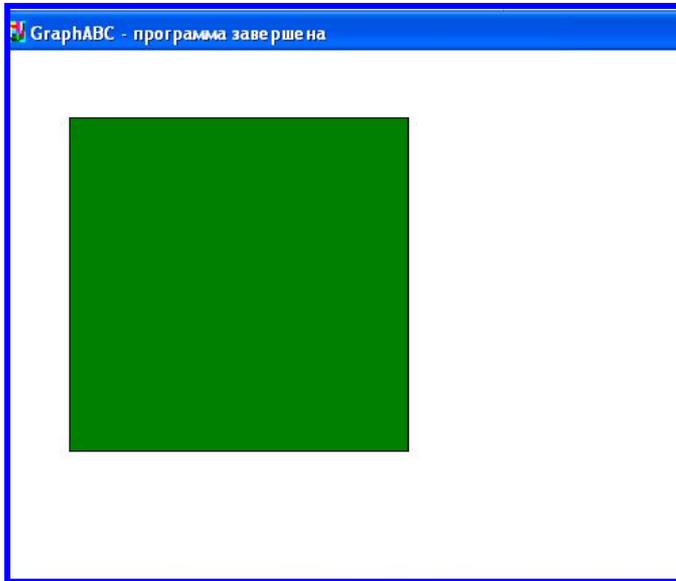
**FloodFill(x,y,color)** - заливает область одного цвета цветом color, начиная с точки (x,y).



```
Program pryamougolnik;  
uses GraphABC;  
begin  
    Rectangle(50,50,200,200);  
    FloodFill(100,100,clBlue);  
end.
```

# 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.
```

# Стили кисти

**SetBrushStyle(название стиля)** - устанавливает стиль кисти, задаваемый перечислимым типом.

**bsSolid** - сплошная кисть (по умолчанию)

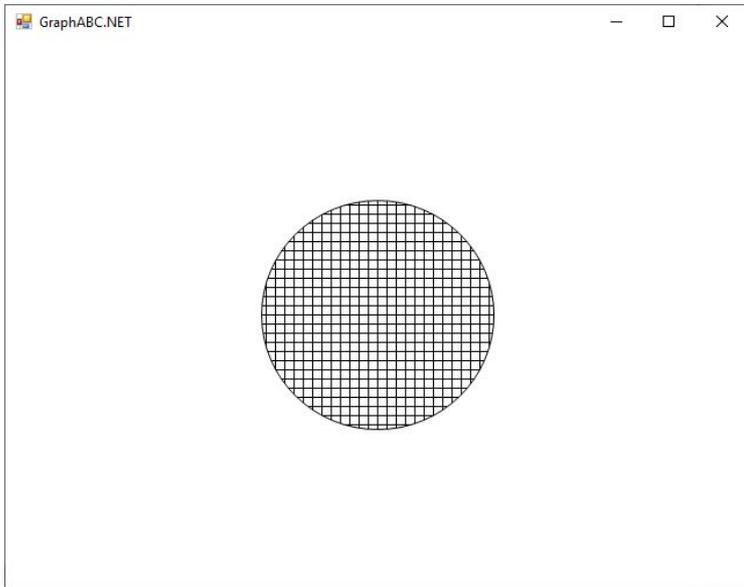
**bsClear** - прозрачная кисть

**bsHatch** - штриховая кисть

**bsGradient** - Градиентная кисть

Для всех кистей используется свойство **Color**. Для штриховой кисти дополнительно можно устанавливать свойства **Hatch** и **HatchBackgroundColor**, для градиентной - свойство **GradientSecondColor**.

# Стили кисти (штриховка)



```
Program zalivka_chtrix;  
uses GraphABC;  
Begin  
SetBrushStyle(bsHatch);  
Circle(320,240,100);  
end.
```

# Штриховка кистью

**SetBrushHatch(название штриха)** - устанавливает стиль штриховки, задаваемый символической константой.

```
Program p12_zalivka;  
uses GraphABC;  
Begin  
SetBrushStyle(bsHatch);  
SetBrushColor(clAquamarine);  
SetBrushHatch(bhHorizontal);  
Rectangle(10,10,100,100);  
SetBrushColor(clRed);  
SetBrushHatch(bhVertical);  
Rectangle(110,10,200,100);  
SetBrushColor(clBlue);  
SetBrushHatch(bhForwardDiagonal);  
Rectangle(210,10,300,100);  
SetBrushColor(clGreen);  
SetBrushHatch(bhBackwardDiagonal);  
Rectangle(10,110,100,210);  
SetBrushColor(clYellow);  
SetBrushHatch(bhCross);  
Rectangle(110,110,200,210);  
SetBrushColor(clBlack);  
SetBrushHatch(bhDiagonalCross);  
Rectangle(210,110,300,210);  
end.
```



# Домашнее задание

Написать программу используя команды, рассматриваемые на уроке (**Line(x1,y1,x2,y2); MoveTo(x,y); LineTo(x,y); Rectangle(x1,y1,x2,y2); FloodFill(x,y,color);**).

Не забудьте установить толщину контура

