

**П А С С А Л**

# Графика в Pascal ABC.NET



19.09.2016

# Подключение дополнительных библиотек

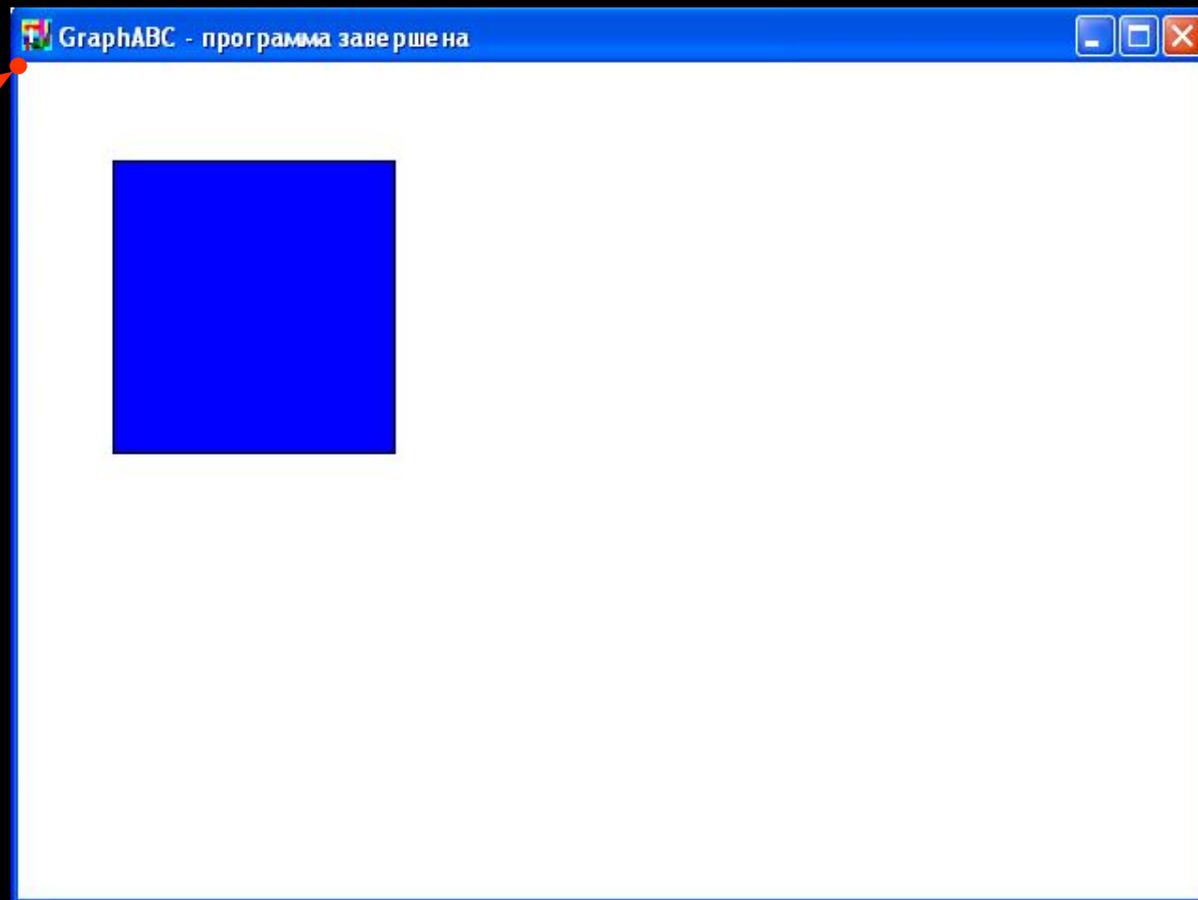
- Для работы в графическом режиме необходимо подключение модуля GraphABC.
- Первой инструкцией программы должна быть инструкция

```
uses GraphABC;
```

# Графический режим

- Графический экран

Центр координат



# Управление окном

```
SetWindowSize(w,h);
```

устанавливает размеры графического окна

```
SetWindowWidth(w);
```

устанавливает ширину графического окна

```
SetWindowHeight(h);
```

устанавливает высоту графического окна

```
SetWindowTitle('Заголовок');
```

меняет заголовок окна

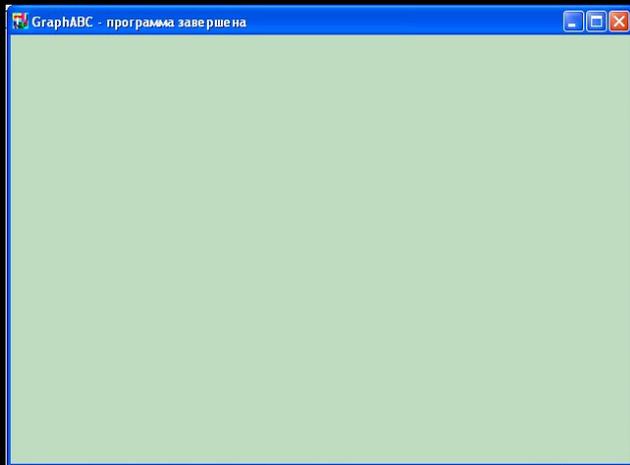
# Очистка графического окна

`ClearWindow;`

очищает графическое окно белым цветом

`ClearWindow(color);`

очищает графическое окно указанным цветом.



```
uses GraphABC;  
begin  
    ClearWindow;  
    ClearWindow (c1MoneyGreen);  
end.
```

Цвет зеленых денег

# Графические примитивы

Точка



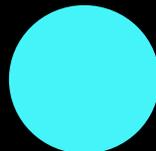
Линия



Прямоугольник



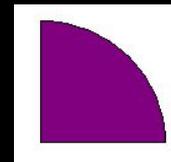
Окружность



Эллипс



Сектор



Дуга



# Точка

`SetPixel(x, y, color);`

закрашивает один пиксел с координатами (x,y)  
цветом color



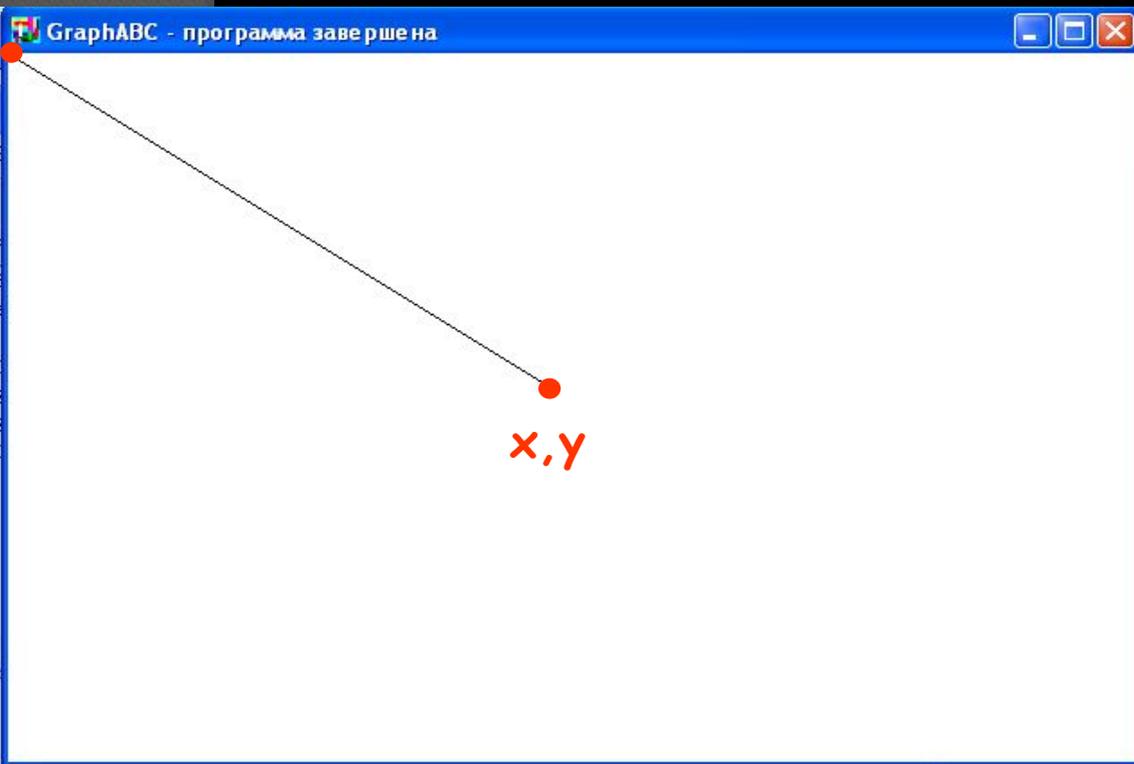
```
uses GraphABC;  
begin  
    SetPixel(300, 200, clred);  
end.
```

# Линии

## LineTo(x,y);

рисует отрезок от текущего положения пера до точки (x,y)

координаты пера при этом также становятся равными (x,y)

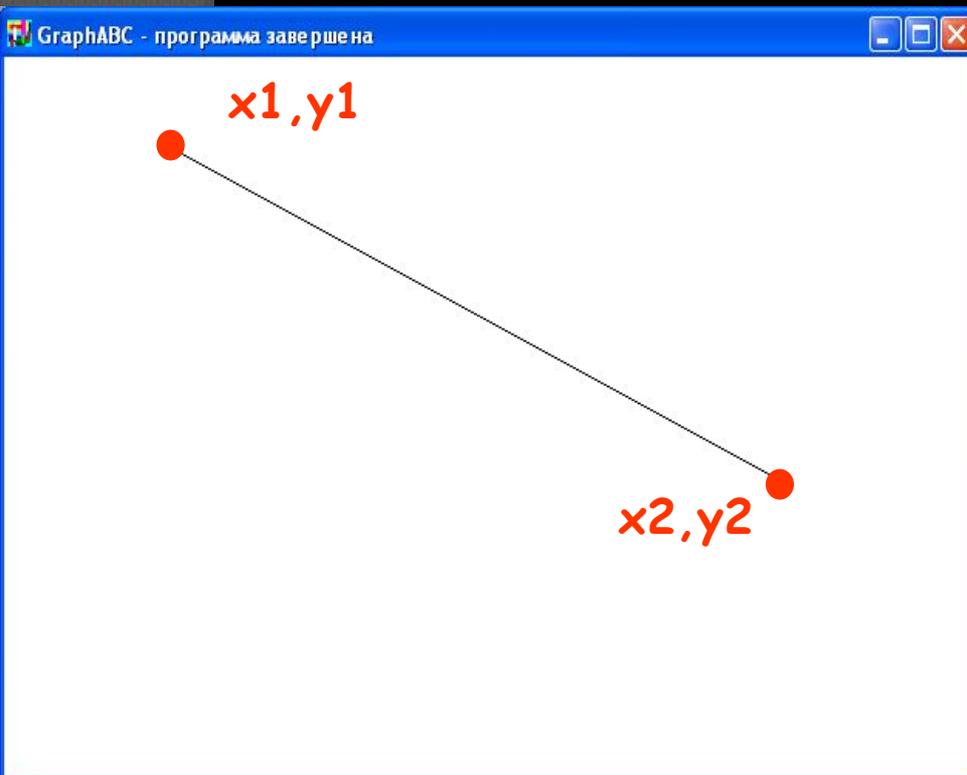


```
uses GraphABC;  
Begin  
    LineTo(300,200);  
end.
```

# Линии

## MoveTo(x, y);

устанавливает текущую позицию рисования в точку (x,y)

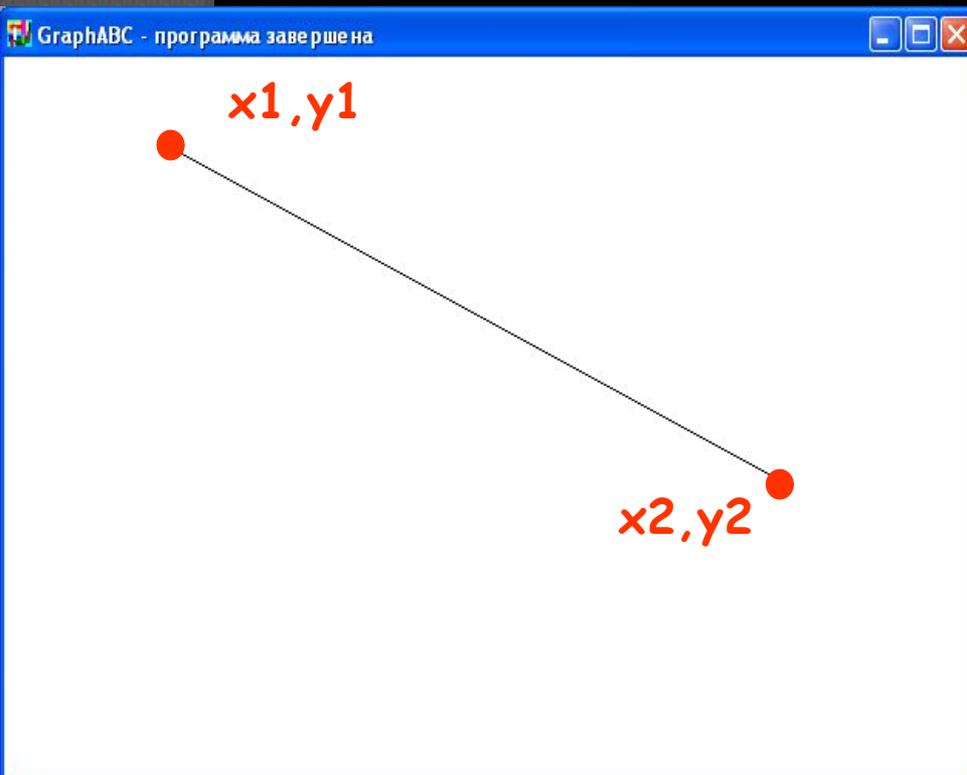


```
uses GraphABC;  
Begin  
    MoveTo(150, 50);  
    LineTo(500, 250);  
end.
```

# Линии

`Line(x1, y1, x2, y2);`

рисует отрезок с началом в точке  $(x1, y1)$  и концом в точке  $(x2, y2)$



```
uses GraphABC;  
begin  
    line(100, 50, 500, 250);  
end.
```

# Используемые цвета

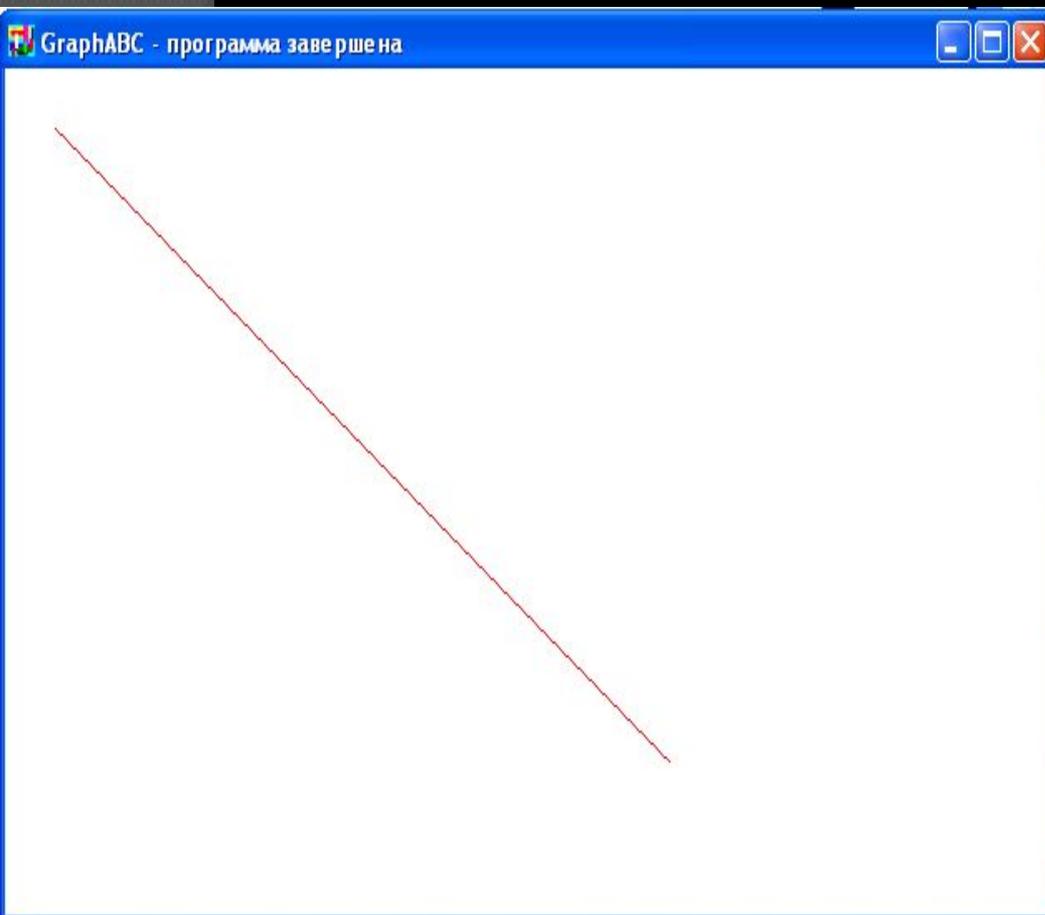
# Используемые цвета

`clRandom` – случайный цвет из всей палитры цветов Паскаля

# Цвет линии

```
SetPenColor(color);
```

устанавливает цвет пера, задаваемый параметром color



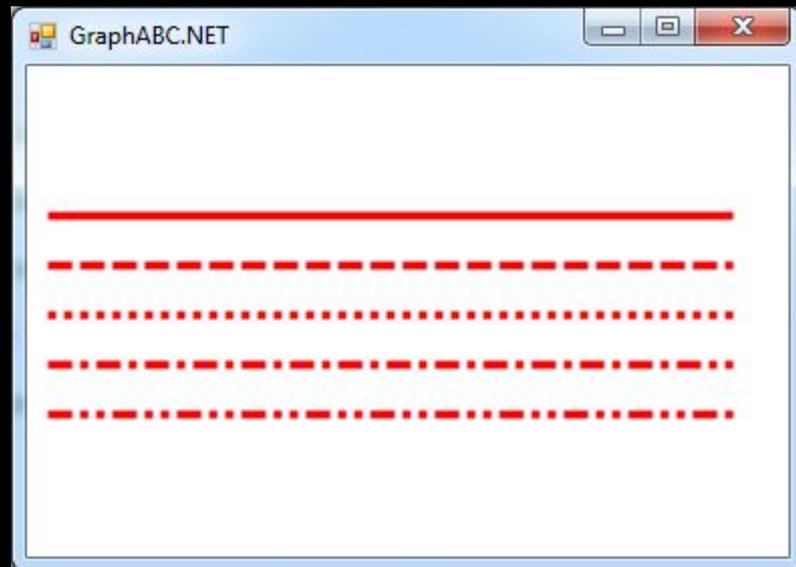
```
uses GraphABC;  
begin  
    SetPenColor(clred);  
    line(30,30,400,350);  
end.
```

# Пунктирная линия

## SetPenStyle(<...>);

устанавливает стиль пера

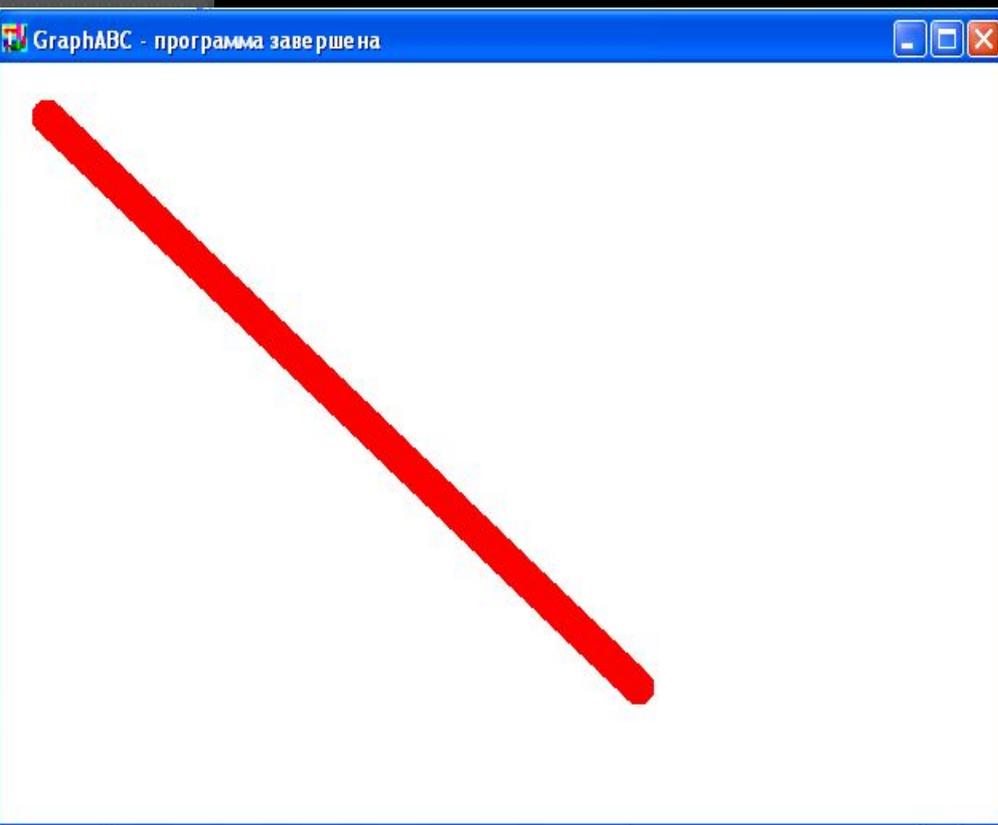
```
uses GraphABC;  
begin  
  SetPencolor(clred);  
  SetPenWidth(4);  
  SetPenStyle(psSolid); {Сплошное}  
  Line(10, 75, 350, 75);  
  SetPenStyle(psDash); {Штриховое}  
  Line(10, 100, 350, 100);  
  SetPenStyle(psDot); {Пунктирное}  
  Line(10, 125, 350, 125);  
  SetPenStyle(psDashDot); {Штрихпунктирное}  
  Line(10, 150, 350, 150);  
  SetPenStyle(psDashDotDot);  
    {Альтернативное штрихпунктирное}  
  Line(10, 175, 350, 175);  
end.
```



# Толщина линии

## SetPenWidth(n);

устанавливает ширину (толщину) пера, равную n  
пикселям

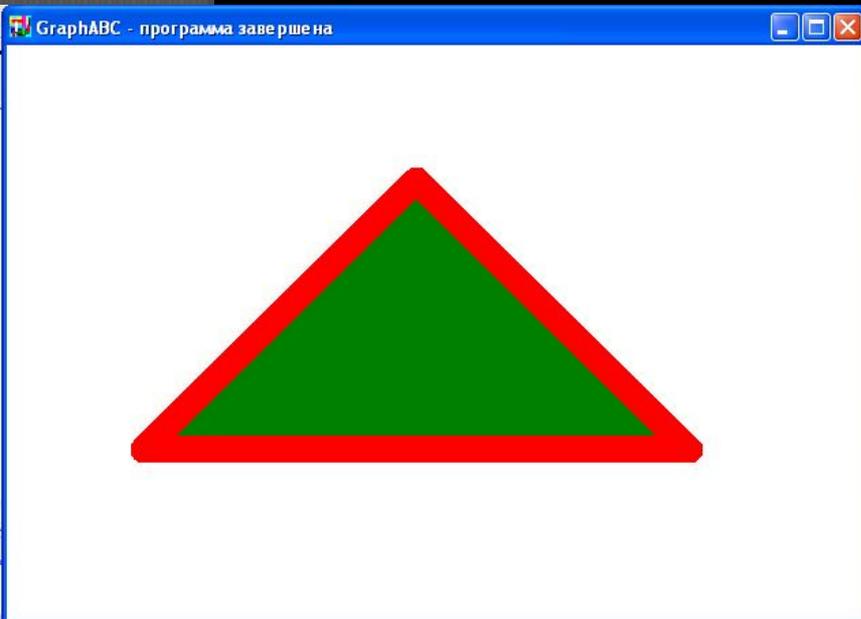


```
uses GraphABC;  
begin  
    setpenwidth(20);  
    setpencolor(clred);  
    line(30,30,400,350);  
end.
```

# Треугольник

```
Line(x1, y1, x2, y2);
```

```
LineTo(x, y);
```

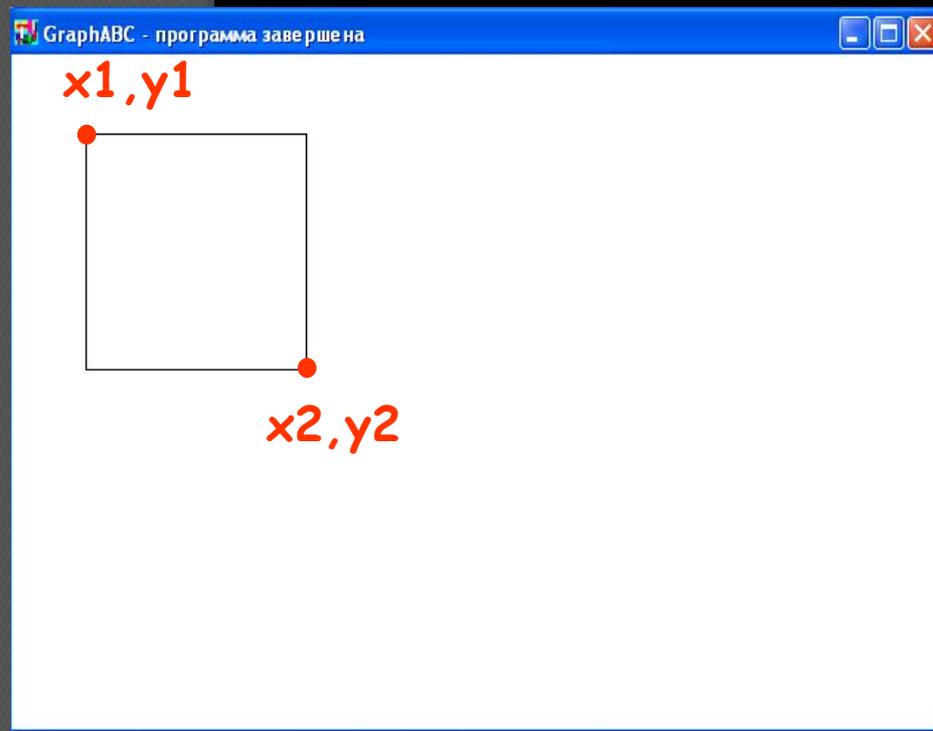


```
uses GraphABC;  
begin  
    setpenwidth(20);  
    setpencolor(clred);  
    line(300, 100, 500, 300);  
    lineto(100, 300);  
    lineto(300, 100);  
    floodfill(300, 200, clgreen);  
end.
```

# Прямоугольник

`Rectangle(x1, y1, x2, y2);`

рисует прямоугольник, заданный координатами противоположных вершин  $(x1, y1)$  и  $(x2, y2)$

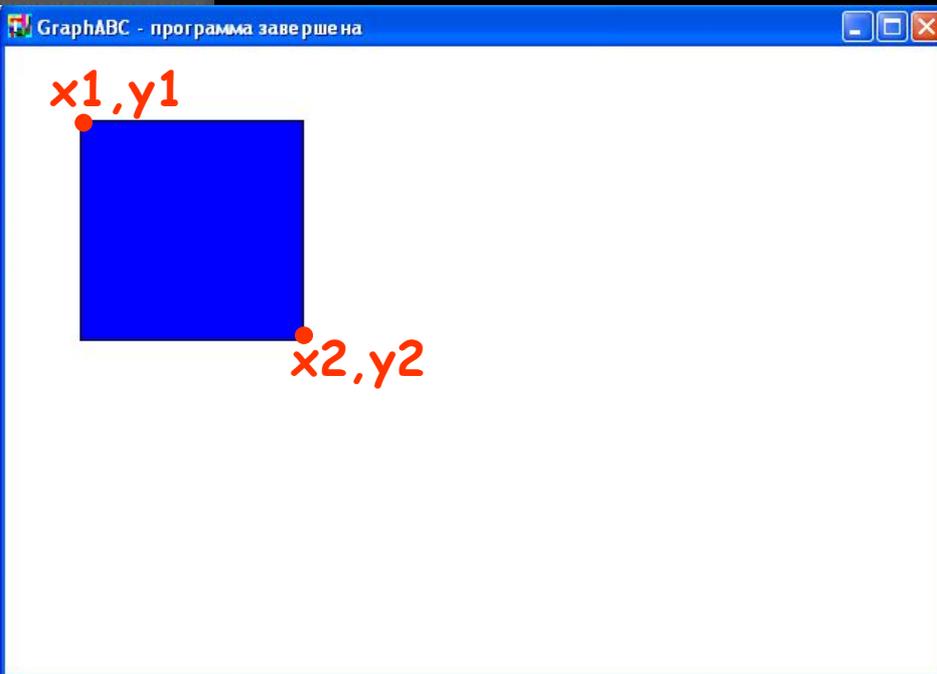


```
uses GraphABC;  
begin  
    Rectangle(50, 50, 200, 200);  
end.
```

# Заливка цветом

## `FloodFill(x, y, color);`

заливает область одного цвета цветом `color`, начиная с точки  $(x, y)$

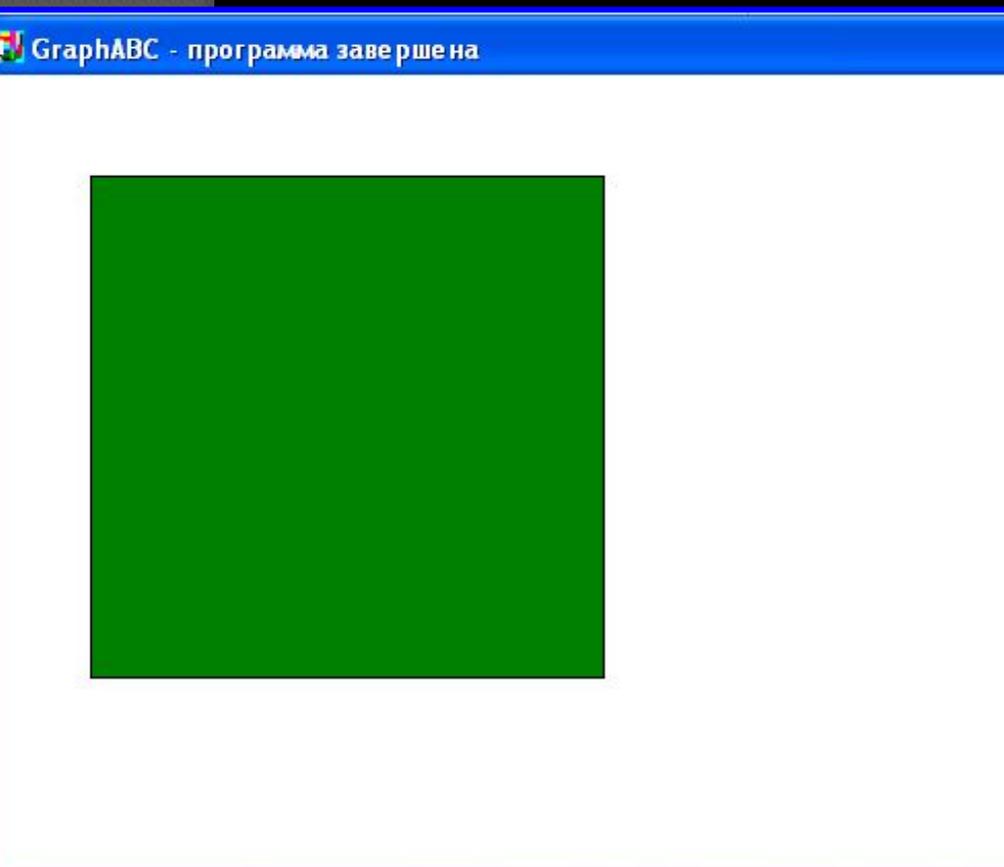


```
uses GraphABC;  
begin  
    Rectangle(50, 50, 200, 200);  
    FloodFill(100, 100, clBlue);  
end.
```

# Заливка кистью

## SetBrushColor(color);

устанавливает цвет кисти, распространяется на замкнутый контур, описание которого следует за процедурой установки цвета кисти



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

# Заливка кистью

```
SetBrushStyle(<...>);
```

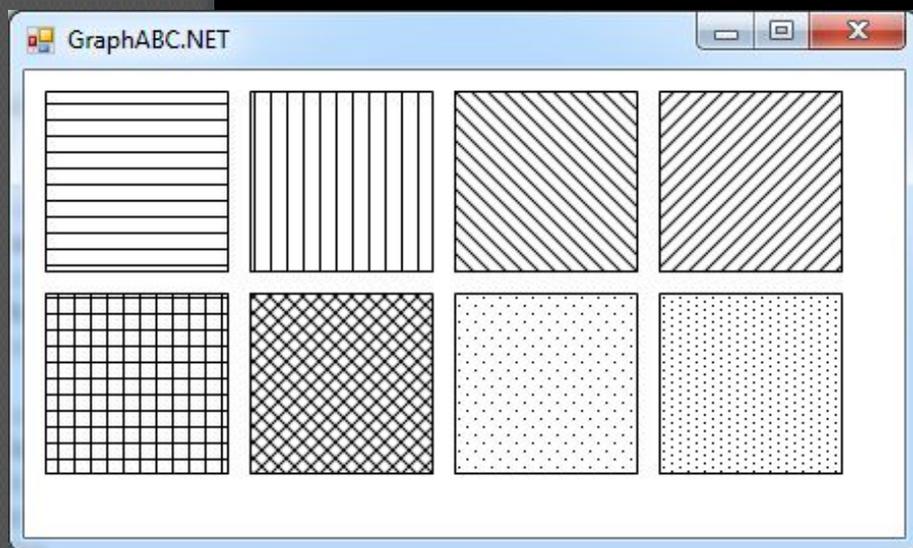
устанавливает тип стиля кисти

# Заливка кистью

Стили штриховки кисти задаются перечислимым типом

`SetBrushHatch(<...>);`

Для стилей штриховки кисти определены  
константы:

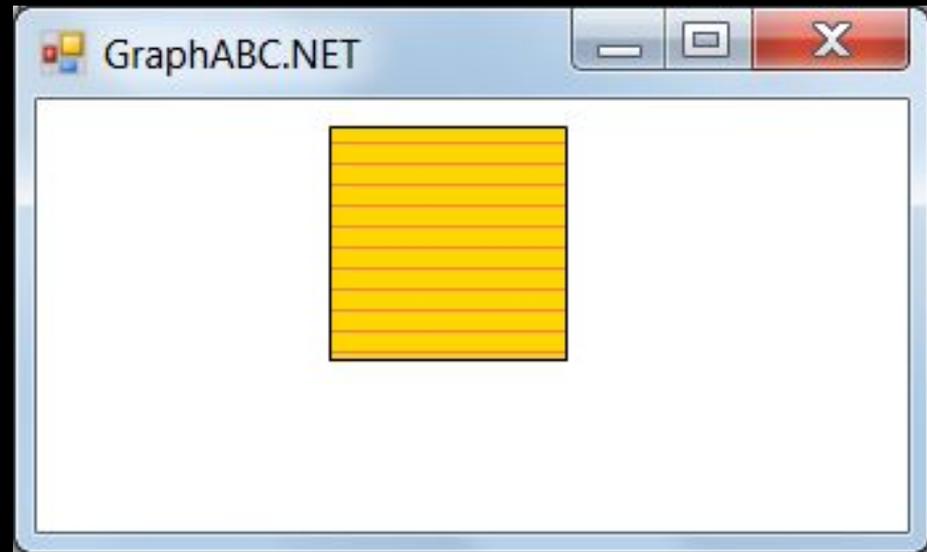


```
uses GraphABC;  
Begin  
  SetBrushStyle(bsHatch);  
  SetBrushHatch(bhHorizontal);  
  Rectangle(10,10,100,100);  
  ...  
end.
```

# Заливка кистью

Для штриховой кисти дополнительно можно устанавливать свойство:

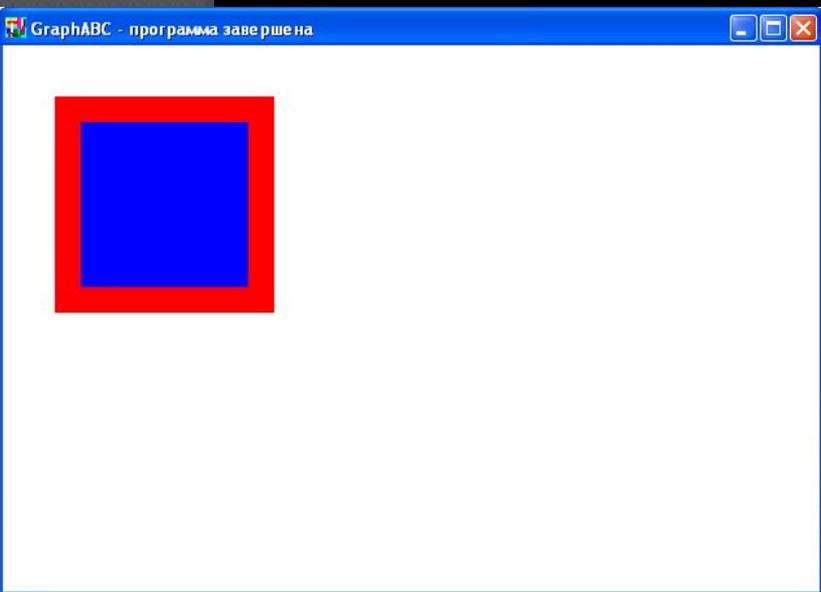
```
SetHatchBrushBackgroundColor(clGold);
```



```
uses GraphABC;  
Begin  
SetBrushStyle(bsHatch);  
SetHatchBrushBackgroundColor(clGold);  
SetBrushColor(clCoral);  
SetBrushHatch(bhHorizontal);  
Rectangle(110,10,200,100);  
end.
```

# Цвет и толщина контура

```
SetPenWidth(w);  
SetPenColor(color);
```

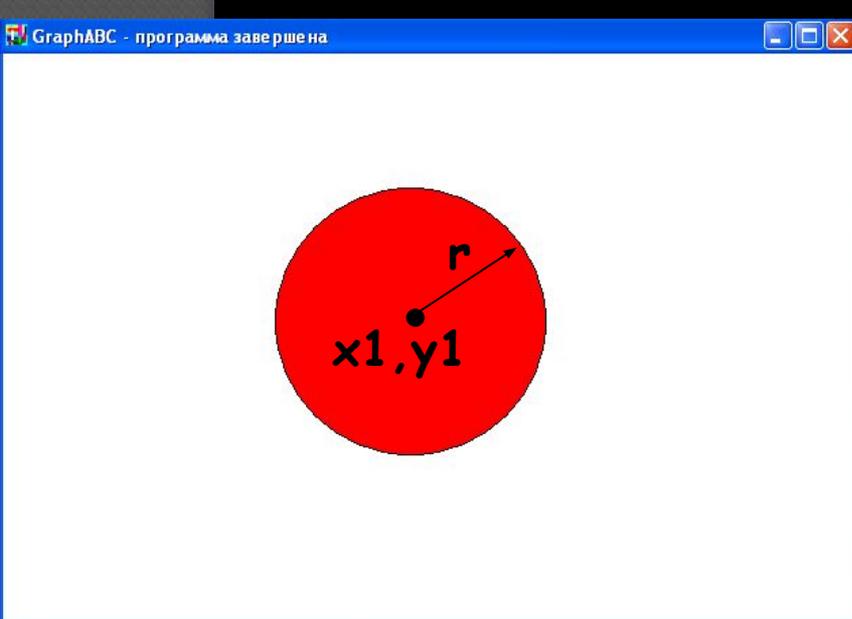


```
uses GraphABC;  
begin  
    SetPenColor(c1red);  
    SetPenWidth(20);  
    Rectangle(50,50,200,200);  
    FloodFill(100,100,c1Blue);  
end.
```

# Окружность

`Circle(x,y,r);`

рисует окружность с центром в точке  $(x,y)$  и радиусом  $r$

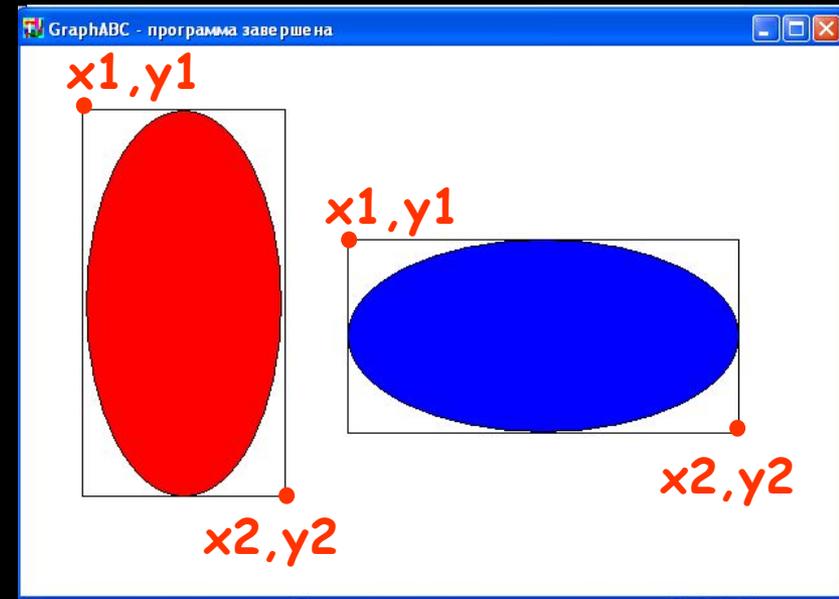


```
uses GraphABC;  
begin  
    Circle(500,200,100);  
    FloodFill(500,200,clred);  
end.
```

# Эллипс

`Ellipse(x1, y1, x2, y2);`

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

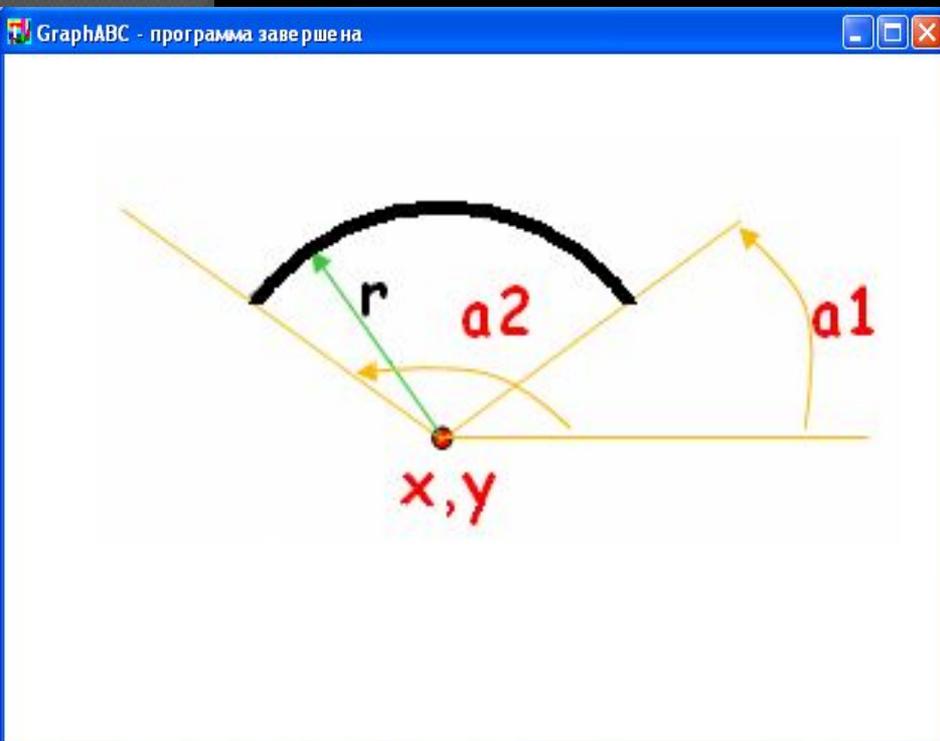


```
uses GraphABC;  
begin  
  Ellipse(50, 50, 200, 350);  
  FloodFill(50+100, 50+100, clRed);  
  Ellipse(250, 150, 550, 300);  
  FloodFill(250+100, 150+100, clBlue);  
end.
```

# 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)

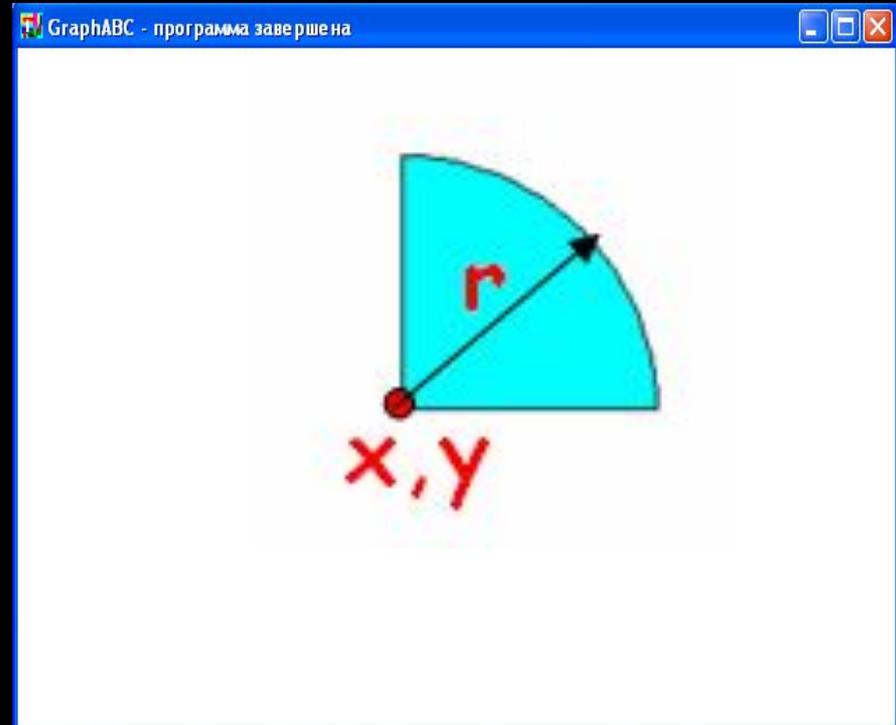


```
uses GraphABC;  
Begin  
SetPenWidth(10);  
Arc(300, 250, 150, 45, 135*);  
end.
```

# Сектор

`Pie(x,y,r,a1,a2);`

рисует сектор окружности, ограниченный дугой (параметры процедуры имеют тот же смысл, что и в процедуре Arc)



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

# Вывод текста

`TextOut(x, y, 'строка');`

выводит строку текста в позицию (x,y) (точка (x,y) задает верхний левый угол прямоугольника, который будет содержать текст)



```
uses GraphABC;  
begin  
  TextOut(100, 30, 'Квадрат');  
  Rectangle(50, 50, 200, 200);  
  FloodFill(55, 55, clBlue);  
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

# Название шрифта

По умолчанию установлен шрифт, имеющий наименование MS Sans Serif

Наиболее распространенные шрифты – это Times New Roman, Arial и Courier New

Наименование шрифта можно набирать без учета регистра

Например:

```
SetFontName('Times New Roman');
```

# Стиль шрифта

`fsNormal` – обычный

**`fsBold`** – жирный

*`fsItalic`* – наклонный

***`fsBoldItalic`*** – жирный наклонный

`fsUnderline` – подчеркнутый

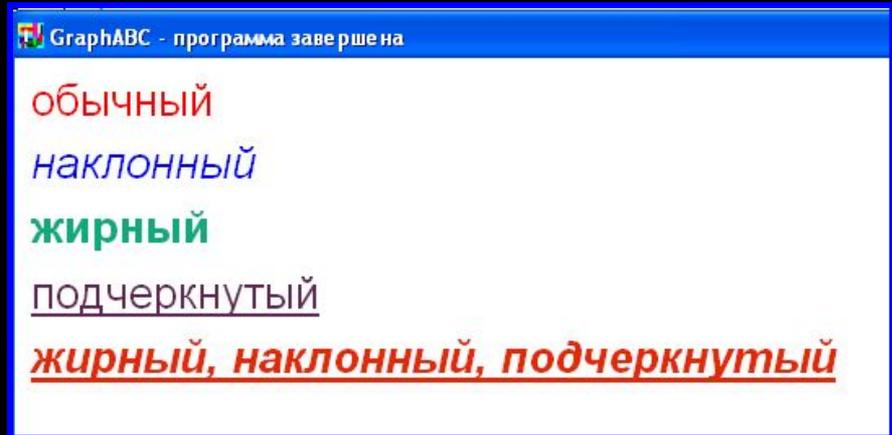
**`fsBoldUnderline`** – жирный подчеркнутый

*`fsItalicUnderline`* – наклонный подчеркнутый

***`fsBoldItalicUnderline`*** – жирный наклонный подчеркнутый

Например:

```
uses GraphABC;  
Begin  
  SetFontName('Arial');  
  SetFontSize(20);  
  SetFontColor(c1Red);  
  TextOut(10,10,'обычный');  
  SetFontStyle(fsItalic);  
  SetFontColor(c1Blue);  
  TextOut(10,50,'наклонный');  
  SetFontStyle(fsBold);  
  SetFontColor(c1Random);  
  TextOut(10,90,'жирный');  
  SetFontStyle(fsUnderline);  
  SetFontColor(c1Random);  
  TextOut(10,130,'подчеркнутый');  
  SetFontStyle(fsBoldItalicUnderline);  
  SetFontColor(c1Random);  
  TextOut(10,170,'жирный, наклонный, подчеркнутый');  
end.
```



# Используемые цвета

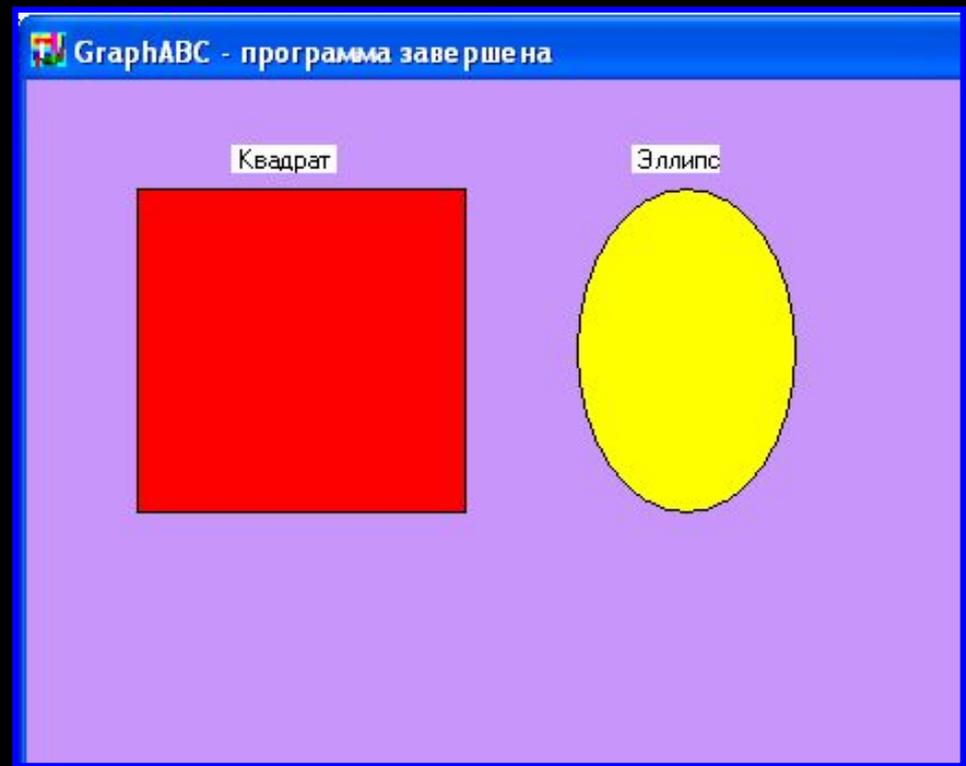
Цвет можно задавать и с помощью функции **RGB(r, g, b)** где r, g и b – целые числа в диапазоне от 0 до 255.

Функция возвращает целое значение, являющееся кодом цвета, который содержит красную, зеленую и синюю составляющие с интенсивностями r, g и b соответственно (0 соответствует минимальной интенсивности, 255 – максимальной).

**RGB(255, 255, 255)** – соответствует белому цвету.

**RGB(0, 0, 0)** – соответствует черному цвету.

Например:



```
uses GraphABC;  
begin  
  Clearwindow(rgb(200,150,250));  
  TextOut(93,30,' Квадрат ');  
  Rectangle(50,50,200,200);  
  FloodFill(55,55,c1Red);  
  TextOut(275,30,' Эллипс');  
  Ellipse(250,50,350,200);  
  FloodFill(250+50,50+50,c1Yellow);  
end.
```