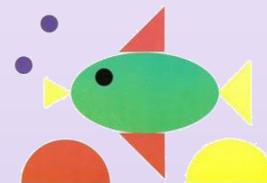


Автор: Зубкова Екатерина Александровна  
ГБОУ СОШ №1378 г. Москва



# ГРАФИКА



Язык программирования PascalABC

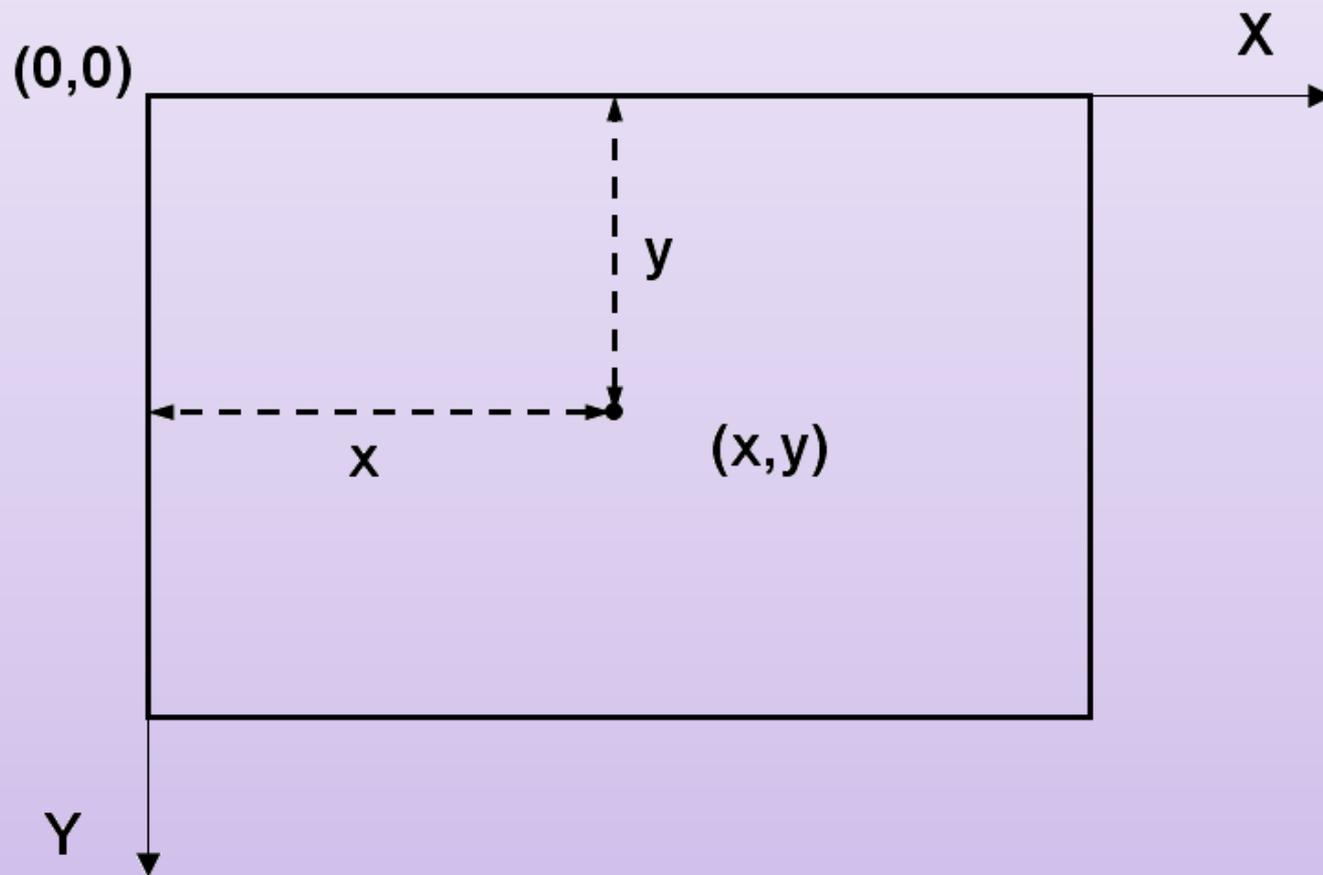
# МОДУЛЬ GraphABC

**Graph** – это сокращение слова «графика».

Чтобы рисовать на экране разноцветные точки, отрезки прямых, дуги, закрашенные и не закрашенные окружности, прямоугольники, а также выполнять ряд других действий, необходимо включить модуль **GraphABC**.

```
Program название;  
uses GraphABC;  
Begin  
...  
End.
```

# СИСТЕМА КООРДИНАТ

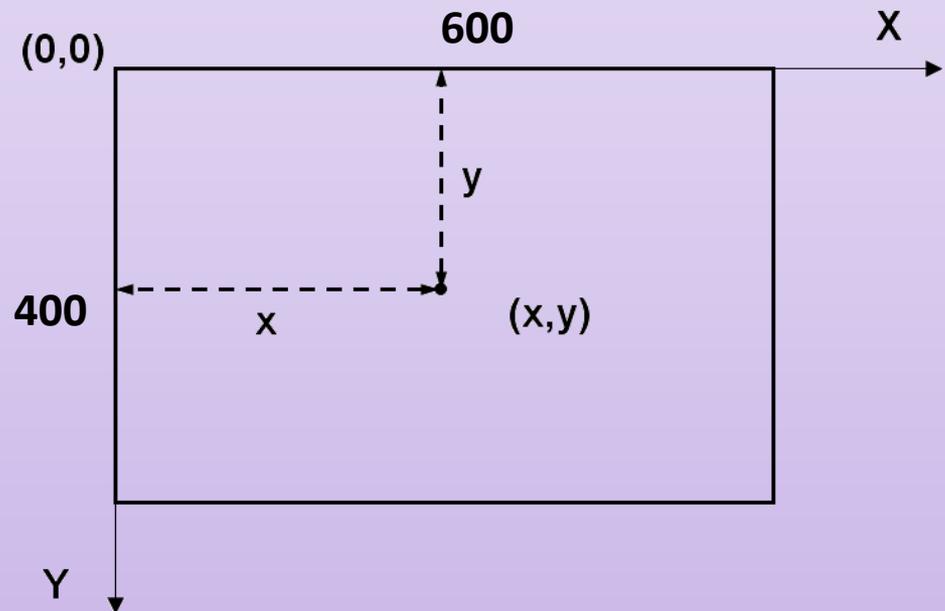


# Операторы, используемые в графике:

**SetWindowHeight(h);** - устанавливает высоту графического окна

**SetWindowWidth(w);** - устанавливает ширину графического окна

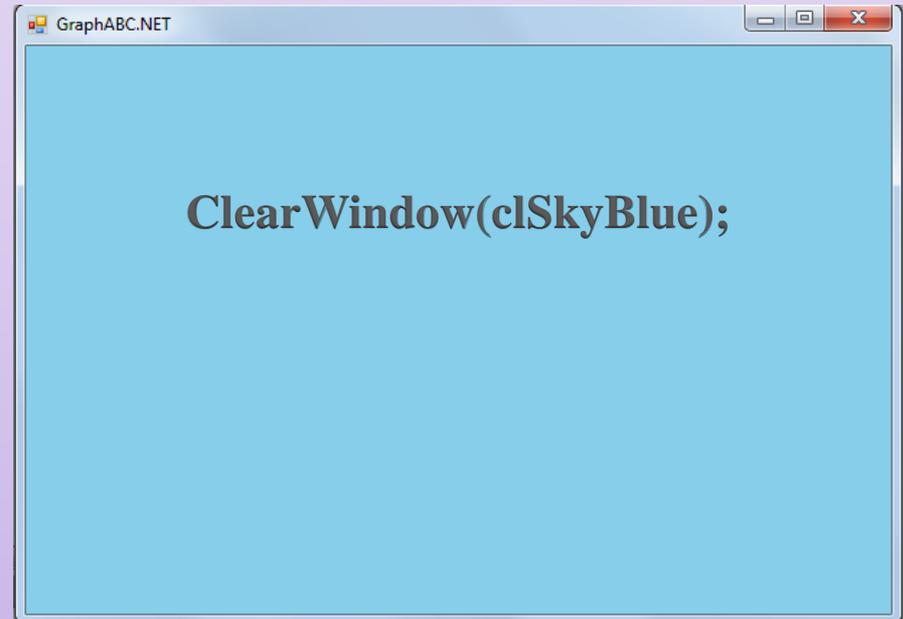
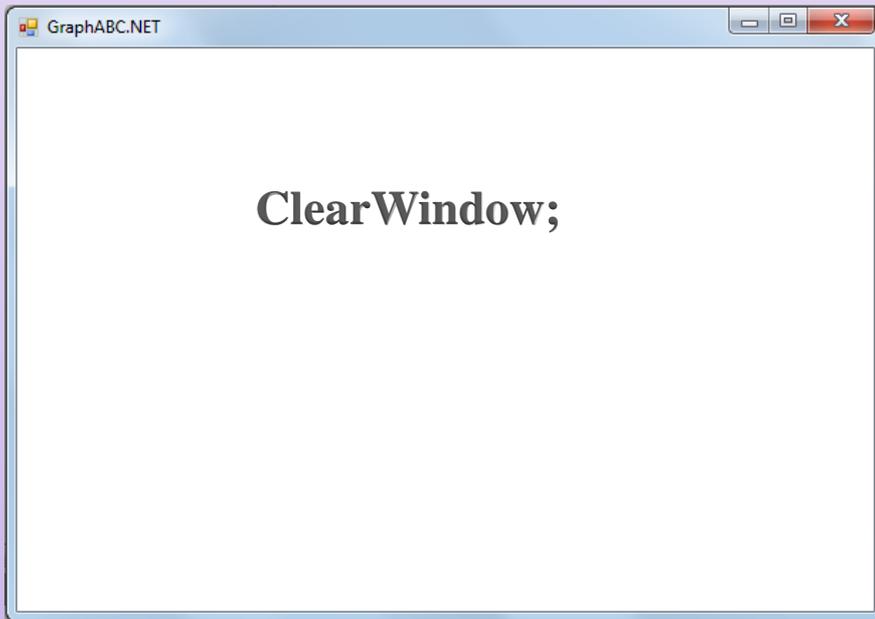
```
SetWindowWidth(600);  
SetWindowHeight(400);
```



# Цвет экрана

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

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

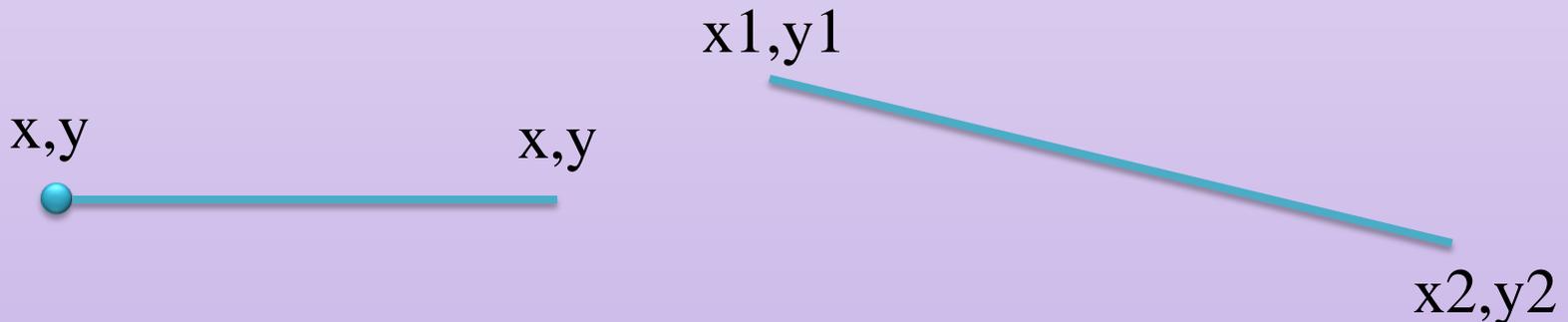


# Графические примитивы

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

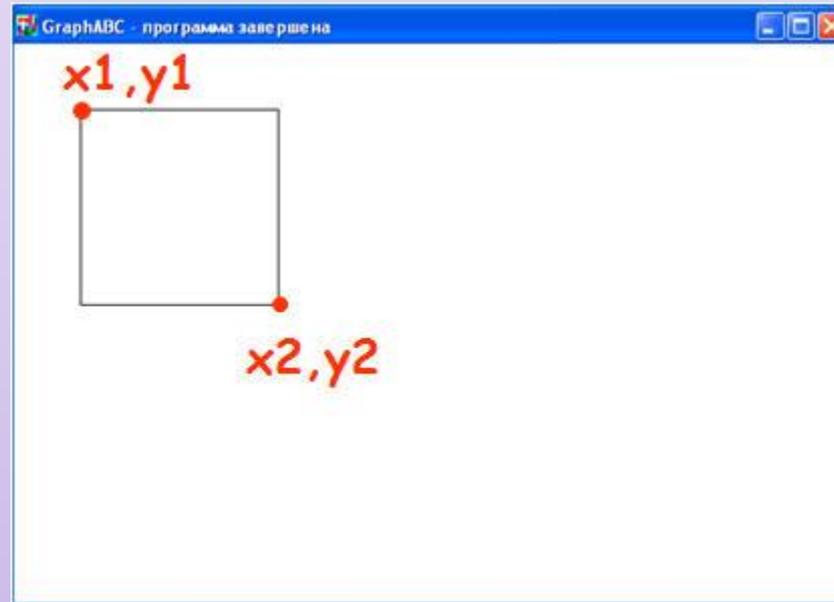
**LineTo(x,y);** - рисует отрезок от текущего положения пера до точки (x,y); координаты пера при этом также становятся равными (x,y).

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



# Графические примитивы

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

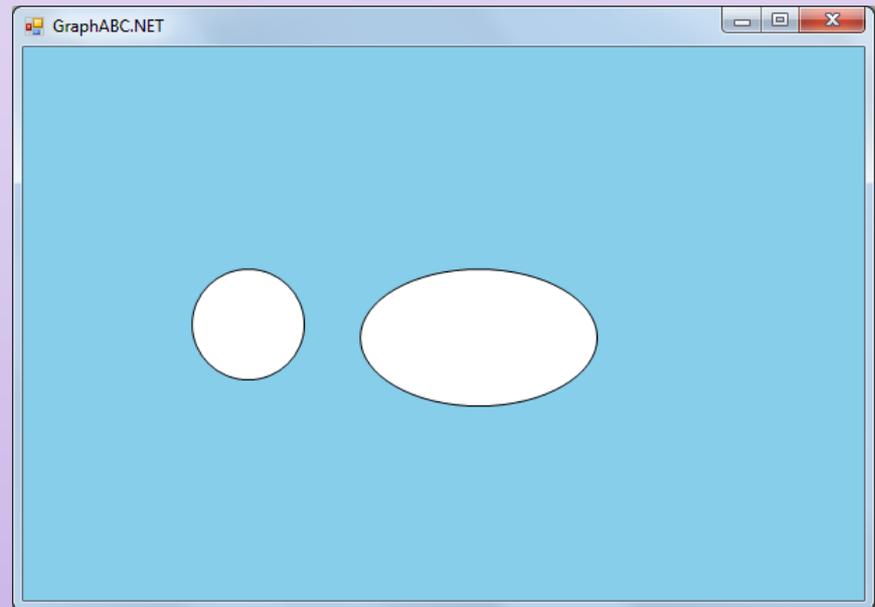


# Графические примитивы

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

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

```
SetWindowHeight(400);  
SetWindowWidth(600);  
ClearWindow(clSkyBlue);  
circle (160,200,40);  
Ellipse(240,160,410,260);
```

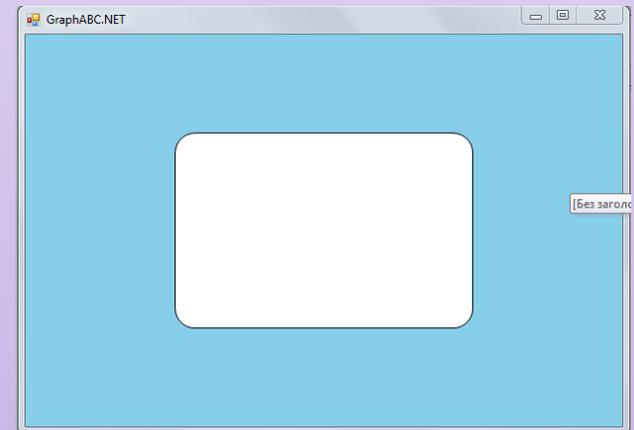


# Графические примитивы

**MoveTo(x,y: integer);** - передвигает невидимое перо к точке с координатами (x,y);

**RoundRect(x1,y1,x2,y2,w,h);** - рисует прямоугольник со скругленными краями; (x1,y1) и (x2,y2) задают пару противоположных вершин, а w и h – ширину и высоту эллипса, используемого для скругления краев

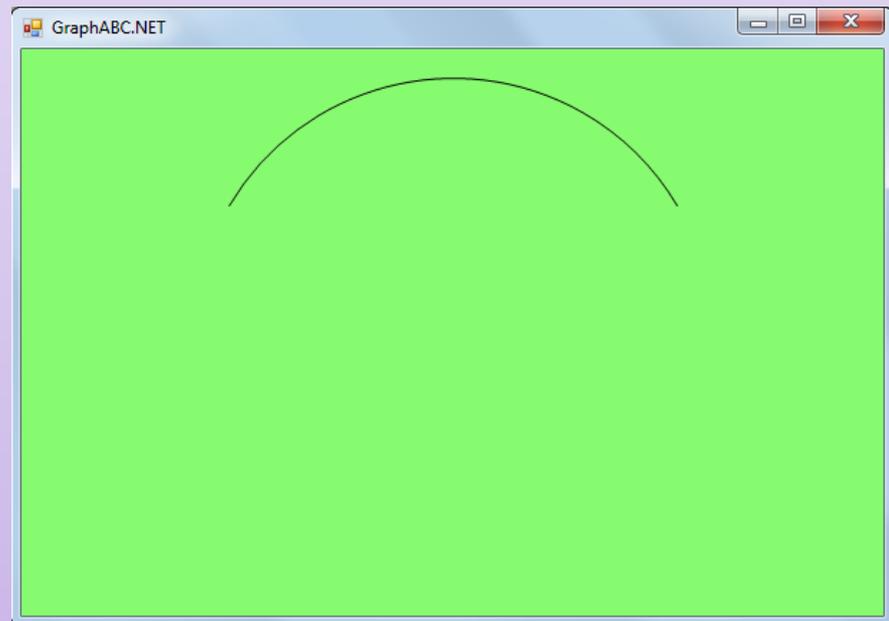
**RoundRect(150,100,450,300,40,40);**



# Графические примитивы

**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).

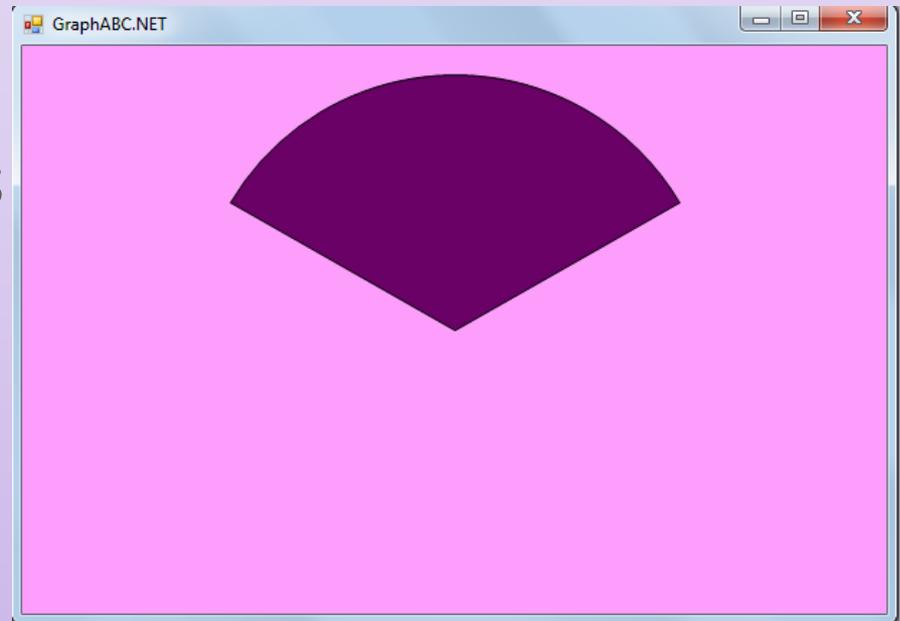
**Arc(300,200,180,30,150);**



# Графические примитивы

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

```
ClearWindow(rgb(254,158,252));  
SetBrushColor(rgb(105,1,103));  
Pie(300,200,180,30,150);
```



# Работа с кистью и областью

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

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

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

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

# Работа с кистью и областью

**SetPenColor(clblue);**

Line (200,200,400,200);

Line (400,200,300,140);

line (300,140,200,200);

**FloodFill (300,170,clblue);**

**SetPenColor(cllime);**

Line (200,200,400,200);

Line (400,200,300,260);

Line (300,260,200,200);

**FloodFill (300,230,cllime);**

**SetPenColor(clred);**

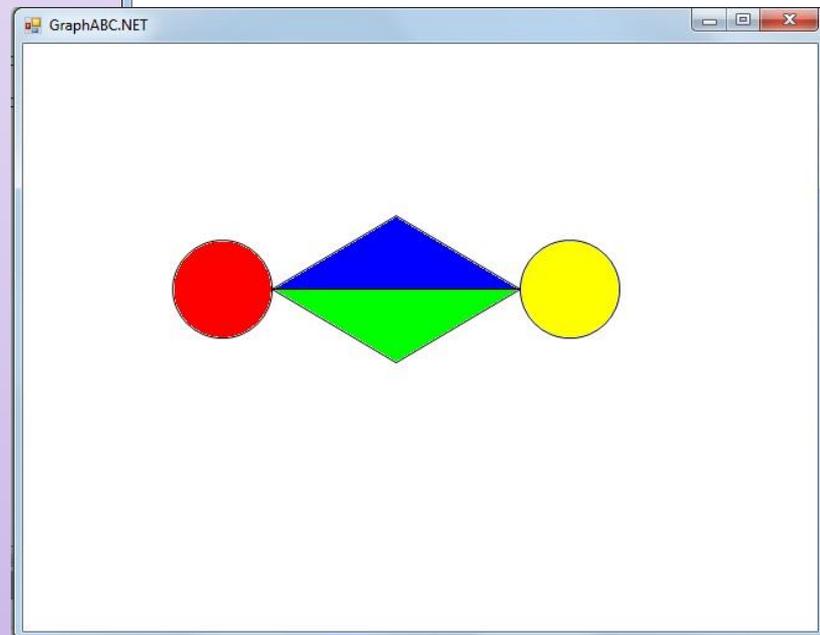
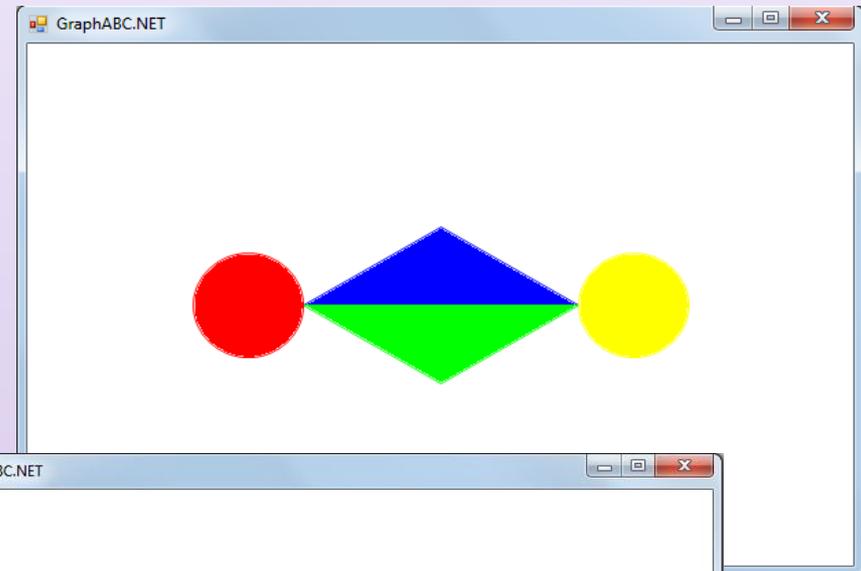
circle (160,200,40);

**FloodFill (160,200,clred);**

**SetPenColor(clyellow);**

circle (440,200,40);

**FloodFill (440,200,clyellow);**



# Таблица цветов

clBlack	черный	
clPurple	фиолетовый	
clWhite	белый	
clMaroon	темно-красный	
clRed	красный	
clNavy	темно-синий	
clGreen	зеленый	
clBrown	коричневый	
clBlue	синий	
clSkyBlue	голубой	
clYellow	желтый	
clOlive	оливковый	

# Таблица цветов

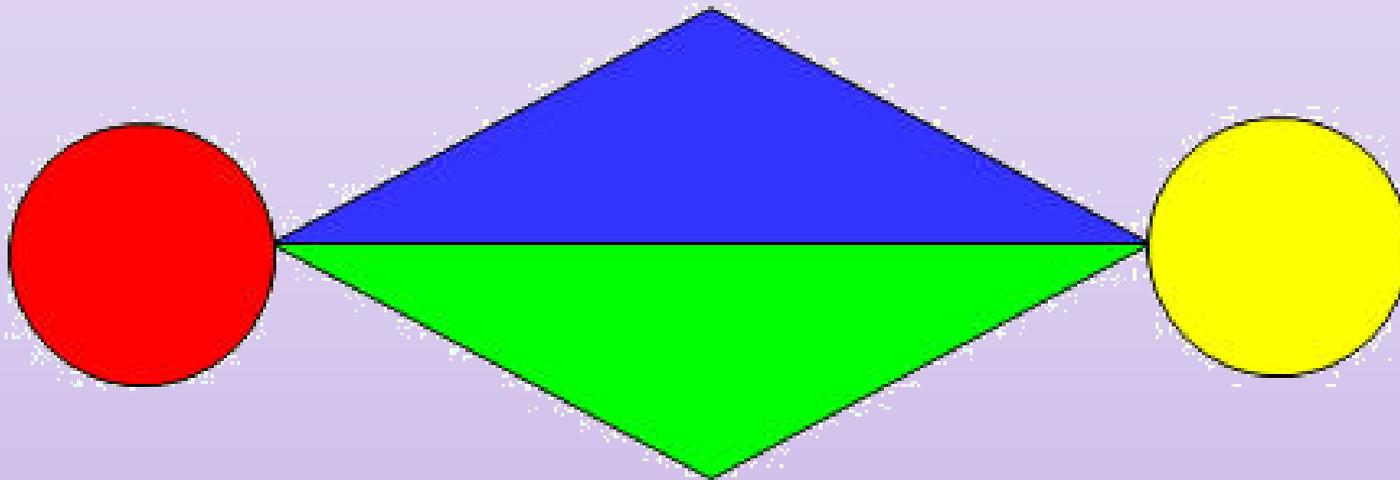
clFuchsia	сиреневый	
clTeal	сине-зеленый	
clGray	темно-серый	
clLime	ярко-зеленый	
clMoneyGreen	цвет зеленых денег	
clSilver	серебряный	

Цвет также можно задать с помощью палитры RGB для это за место color пишется `rgb(r,g,b)`, где `r,b,g` - числа от 0 до 255

```
FloodFill (300,170,rgb(138,248,68));
```

**Выполни на компьютере**

**Задача:** нарисовать  
представленный ниже рисунок



# Текст программы

**Program** Seventh;

**uses** GraphABC;

**Begin**

SetWindowHeight(400);

SetWindowWidth(600);

Line (200,200,400,200);

Line (400,200,300,140);

line (300,140,200,200);

FloodFill (300,170,clblue);

Line (200,200,400,200);

Line (400,200,300,260);

Line (300,260,200,200);

FloodFill (300,230,cllime);

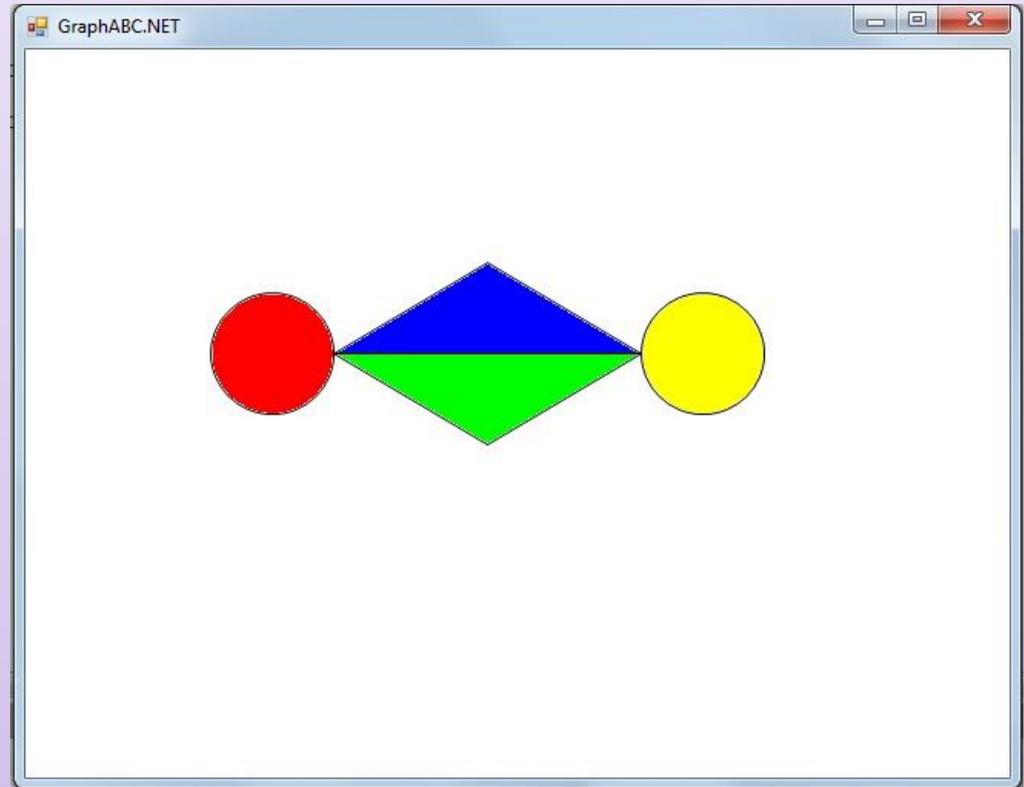
circle (160,200,40);

FloodFill (160,200,clred);

circle (440,200,40);

FloodFill (440,200,clyellow);

**End.**



# Работа с текстом

**TextOut(x,y,s);** - выводит строку s в позицию (x,y) (точка (x,y) задает верхний левый угол прямоугольника, который будет содержать текст из строки s).

**SetFontName('name');** - устанавливает наименование шрифта.

**SetFontColor(color);** - устанавливает цвет шрифта.

**SetFontSize(sz);** - устанавливает размер шрифта в пунктах.

# Работа с текстом

**SetFontStyle(fs);** - устанавливает стиль шрифта.

Стиль шрифта:

**fsNormal** – обычный;

**fsBold** – жирный;

**fsItalic** – наклонный;

**fsBoldItalic** – жирный наклонный;

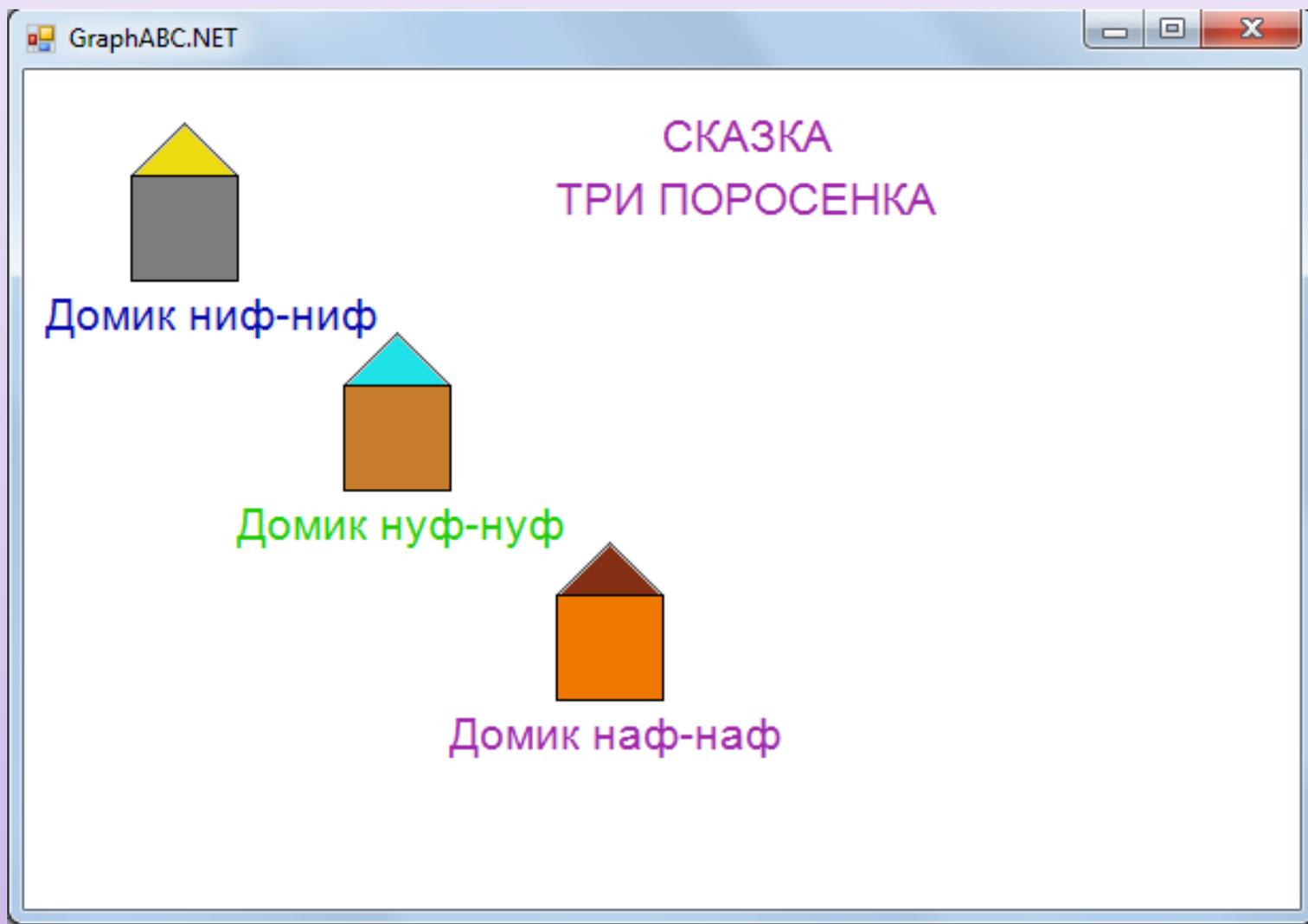
**fsUnderline** – подчеркнутый;

**fsBoldUnderline** – жирный подчеркнутый;

**fsItalicUnderline** – наклонный подчеркнутый;

**fsBoldItalicUnderline** – жирный наклонный подчеркнутый.

# Работа с текстом



# Самостоятельная работа

**Задача:** нарисовать рисунок,  
используя рассмотренные выше  
КОМАНДЫ

---

