

## Работа с графикой на PascalABC.NET

Подключение графической библиотеки: **uses graphABC;**

Размер графического экрана устанавливается с помощью процедуры **SetWindowSize(w,h)**, где **w** и **h** – ширина и высота окна. Значения координат принимают только целочисленные значения: **x** от 0 до **w**, а **y** от 0 до **h**.

Расположение осей координат:



### Основные операторы графики

**putpixel (x, y, c)** ставит точку с координатами (x, y) и цветом c.

**line (x1, y1, x2, y2)** рисует отрезок от точки (x1, y1) до точки (x2, y2).

**circle (x, y, R)** рисует окружность с центром в точке (x, y) и радиусом R.

**rectangle (x1, y1, x2, y2)** прямоугольник с вершинами (x1, y1) и (x2, y2), лежащими на диагонали.

**Пример 1.** Составить программу, которая рисует на экране графические примитивы: отрезок, прямоугольник, круг, эллипс.

Прежде всего подключим модуль graphABC. Зададим размеры графического окна **setWindowSize(600,240);** - ширина 600, а высота 240 пикселей.

Запишем процедуру рисования отрезка, соединяющего точки с координатами (80,40) и (500,40):

```
line(80,40,500,40);
```

Поскольку параметры пера не заданы, то по умолчанию его толщина 1 пиксель, цвет черный. Зададим толщину пера 5 пикселей **setPenWidth(5);** Это значение не изменится до тех пор, пока не будет задано новое. Таким образом, дальше все примитивы рисуются пером такой толщины. Для каждого примитива будем задавать цвет пера и кисти, например, для прямоугольника коричневый цвет пера **setPenColor(clBrown);** и желтый цвет кисти **setBrushColor(clYellow);**

Аналогично нарисуем круг (задаются координаты центра и радиус) и эллипс (задаются координаты противоположных вершин прямоугольника, описывающего эллипс).

Наконец, зададим параметры текста: размер символов 15 пикселей **setFontSize(15);** начертание жирное **setFontStyle(fsBold);** цвет коричневый **setFontColor(clBrown);** цвет фона (кисти) белый **setBrushColor(clWhite).**

Программа может выглядеть так:

```
uses graphABC; //подключение модуля graphABC
begin
setWindowSize(600,240); // размеры окна вывода
line(80,40,500,40); // отрезок
setPenWidth(5); // толщина пера
setPenColor(clBrown); setBrushColor(clYellow); // цвет пера и кисти
rectangle(40,80,200,160); // прямоугольник
setPenColor(clRed); setBrushColor(clGreen);
circle(300,120,40); // круг
setPenColor(clBlue); setBrushColor(clRed);
ellipse(400,80,540,160); // эллипс
setFontSize(15);
setFontStyle(fsBold); // размер и начертание шрифта
setFontColor(clBrown); setBrushColor(clWhite);
textOut(120,180,'Графические примитивы'); // вывод текста
end.
```

## Графические примитивы модуля GraphABC

**SetPixel(x,y,color: integer);** Закрашивает один пиксел с координатами (x, y) цветом color.

**MoveTo(x,y: integer);** Передвигает невидимое перо к точке с координатами (x, y); работает в паре с функцией **LineTo(x, y)**.

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

**Line(x1,y1,x2,y2: integer);** Рисует отрезок с началом в точке (x1, y1) и концом в точке (x2, y2).

**Circle(x,y,r: integer);** Рисует окружность с центром в точке (x, y) радиусом r.

**Ellipse(x1,y1,x2,y2: integer);** Рисует эллипс, заданный описанным прямоугольником с координатами противоположных вершин (x1,y1) и (x2,y2).

**Rectangle(x1,y1,x2,y2: integer);** Рисует прямоугольник, заданный координатами противоположных вершин (x1,y1) и (x2,y2).

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

**Pie(x,y,r,a1,a2: integer);** Рисует сектор круга, ограниченный дугой (параметры процедуры имеют тот же смысл, что и в процедуре Arc).

**FloodFill(x,y,color: integer);** Заливает однотонную область цветом color, начиная с точки (x, y).

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

### Константы стандартных 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

**clCream** – кремовый

**clOlive** – оливковый

**clFuchsia** – сиреневый

**clTeal** – сине-зеленый

**clGray** – темно-серый

**clLime** – ярко-зеленый

**clLtGray** – светло-серый

**clSilver** – серебряный

### Действия с графическим окном

**SetWindowSize(w,h: integer);** Устанавливает ширину и высоту графического окна.

**SetWindowCaption(s: string);** Устанавливает заголовок графического окна.

**ClearWindow;** Очищает графическое окно белым цветом.

**ClearWindow(c: ColorType);** Очищает графическое окно цветом c.

**Redraw;** Осуществляет перерисовку окна

### Действия с пером и кистью

**SetPenColor(color: integer);** Устанавливает цвет пера, задаваемый параметром color.

**SetPenWidth(w: integer);** Устанавливает толщину пера, равную w пикселям.

**SetBrushColor(color: integer);** Устанавливает цвет кисти, задаваемый параметром color.

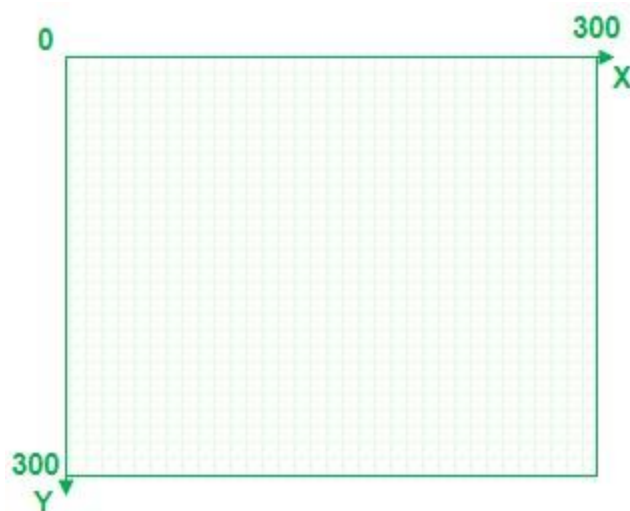
**SetBrushStyle(bs: integer);** Устанавливает стиль кисти, задаваемый параметром bs.

Стили кисти задаются именованными константами:

**bsSolid, bsCross, bsClear, bsDiagCross, bsHorizontal, bsBDiagonal, bsVertical, bsFDiagonal.**

## Работа с графикой на Basic 256

Для вывода графических объектов в Basic-256 существует специальное окно графического вывода. Его размер 300x300 точек (пикселей). Координатная сетка в этом окне задаётся осями координат, расположенными несколько непривычно (см. рисунок). Координаты и параметры графических объектов могут задаваться числом, переменной, получившей к этому моменту числовое значение, или выражением.



Команда	Формат	Описание
Circle	circle $x,y,r$	Рисует текущим цветом круг с центром в точке $(x, y)$ радиусом $r$ .
Clg	clg	Очищает окно графического вывода
Color	color <i>название цвета</i>	Устанавливает текущий цвет для команд рисования
FastGraphics	fastgraphics	Не разрешает обновления графического окна до выполнения команды Refresh (используется в анимации)
Line	line $x1,y1,x2,y2$	Рисует линию от точки $(x1, y1)$ до точки $(x2, y2)$
Plot	plot $x,y$	Рисует точку с координатами $(x, y)$ текущего цвета
Poly	poly <i>массив, n</i>	Рисует $n$ -угольник, координаты вершин которого парами считываются из массива
Rect	rect $x,y,a,b$	Рисует прямоугольник шириной $a$ и высотой $b$ от точки $(x, y)$
Refresh	refresh	Обновляет содержимое окна графического вывода

Пример программы:

```
# очищаем экран
clg
color white
rect 0,0,300,300
# рисуем лицо
color yellow
circle 150,150,150
# рисуем рот
color black
circle 150,200,70
color yellow
circle 150,150,70
# рисуем глаза
color black
circle 100,100,30
circle 200,100,30
```

<b>Образец цвета</b>	<b>код</b>	<b>название цвета</b>
	black (0,0,0)	чёрный
	white (248,248,248)	белый
	red (255,0,0)	красный
	darkred (128,0,0)	темно-красный
	green (0,255,0)	зеленый
	darkgreen (0,128,0)	темно-зеленый
	blue (0,0,255)	синий
	darkblue (0,0,128)	темносиний
	cyan (0,255,255)	голубой
	darkcyan (0,128,128)	темно-голубой
	purple (255,0,255)	пурпурный
	darkpurple (128,0,128)	темно-пурпурный
	yellow (255,255,0)	желтый
	darkyellow (128,128,0)	темно-желтый
	orange (255,102,0)	оранжевый
	darkorange (170,51,0)	темно-оранжевый
	gray или grey (164,164,164)	серый
	darkgray или darkgrey (128,128,128)	темно-серый
	clear (-1)	прозрачный