

1. Графические примитивы

SetPixel (x,y,color: integer);	<u>Закрашивает один пиксел</u> с координатами (x,y) цветом color.
GetPixel (x,y): integer;	<u>Возвращает текущее значение цвета для пиксела</u> с координатами (x,y).
MoveTo (x,y: integer);	<u>Передвигает невидимое перо к точке</u> с координатами (x,y); эта функция работает в паре с функцией LineTo (x,y).
LineTo (x,y: integer);	<u>Рисует отрезок</u> от текущего положения пера до точки (x,y); координаты пера при этом также становятся равными (x,y).
Line (x1,y1,x2,y2: integer);	<u>Рисует отрезок</u> с началом в точке (x1,y1) и концом в точке (x2,y2).
Circle (x,y,r: integer);	<u>Рисует окружность</u> с центром в точке (x,y) и радиусом r.
Ellipse (x1,y1,x2,y2: integer);	<u>Рисует эллипс</u> , заданный своим описанным прямоугольником с координатами противоположных вершин (x1,y1) и (x2,y2).
Rectangle (x1,y1,x2,y2: integer);	<u>Рисует прямоугольник</u> , заданный координатами противоположных вершин (x1,y1) и (x2,y2).
RoundRect (x1,y1,x2,y2,w,h: integer);	<u>Рисует прямоугольник со скругленными краями</u> ; (x1,y1) и (x2,y2) задают пару противоположных вершин, a w и h – ширину и высоту эллипса, используемого для скругления краев.
Arc (x,y,r,a1,a2: integer);	<u>Рисует дугу окружности</u> 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).
Pie (x,y,r,a1,a2: integer);	<u>Рисует сектор окружности</u> , ограниченный дугой (параметры процедуры имеют тот же смысл, что и в процедуре Arc).
Chord (x,y,r,a1,a2: integer);	<u>Рисует фигуру, ограниченную дугой окружности и отрезком</u> , соединяющим ее концы (параметры процедуры имеют тот же смысл, что и в процедуре Arc).
TextOut (x,y: integer; s: string);	<u>Выводит строку</u> s в позицию (x,y) (точка (x,y) задает верхний левый угол прямоугольника, который будет содержать текст из строки s).
FloodFill (x,y,color: integer);	<u>Заливает область</u> одного цвета цветом color, начиная с точки (x,y).
FillRect (x1,y1,x2,y2: integer);	<u>Заливает прямоугольник</u> , заданный координатами противоположных вершин (x1,y1) и (x2,y2), цветом текущей кисти.
Polygon (var a; n: integer);	<u>Строит ломаную</u> по n точкам, координаты которых заданы в массиве a элементов типа Point.
Polyline (var a; n: integer);	<u>Строит замкнутую ломаную</u> по n точкам, координаты которых заданы в массиве a элементов типа Point.

2. Цветовые константы и функции для работы с цветом

Модуль GraphABC содержит константы и функции для работы с цветами. Тип ColorType, описывающий цвет, определен следующим образом:

```
type ColorType=integer;
```

Стандартные цвета задаются символическими константами:

clBlack – черный	clAqua – бирюзовый
clPurple – фиолетовый	clOlive – оливковый
clWhite – белый	clFuchsia – сиреневый
clMaroon – темно-красный	clTeal – сине-зеленый
clRed – красный	clGray – темно-серый
clNavy – темно-синий	clLime – ярко-зеленый
clGreen – зеленый	clMoneyGreen – цвет зеленых денег
clBrown – коричневый	clLtGray – светло-серый
clBlue – синий	clDkGray – темно-серый
clSkyBlue – голубой	clMedGray – серый
clYellow – желтый	clSilver – серебряный
clCream – кремовый	

Для работы с цветами используются следующие функции.

RGB (r,g,b: integer): ColorType;	Возвращает целое значение, являющееся кодом цвета, который содержит красную, зеленую и синюю составляющие с интенсивностями r, g и b соответственно (r, g и b – целые в диапазоне от 0 до 255, причем, 0 соответствует минимальной интенсивности, 255 – максимальной).
GetRed (color: ColorType): integer;	Выделяет красную составляющую из цвета color (целое в диапазоне от 0 до 255);
GetGreen (color: ColorType): integer;	Выделяет зеленую составляющую из цвета color (целое в диапазоне от 0 до 255);
GetBlue (color: ColorType): integer;	Выделяет синюю составляющую из цвета color (целое в диапазоне от 0 до 255).

3. Действия с пером

<code>PenX: integer;</code> <code>PenY: integer;</code>	Возвращают текущие координаты пера.
<code>SetPenColor(color: integer);</code>	Устанавливает цвет пера, задаваемый параметром <code>color</code> .
<code>PenColor: integer;</code>	Возвращает текущий цвет пера
<code>SetPenWidth(w: integer);</code>	Устанавливает ширину пера, равную <code>w</code> пикселям.
<code>PenWidth: integer;</code>	Возвращает текущую ширину пера
<code>SetPenStyle(ps: integer);</code>	Устанавливает стиль пера, задаваемый параметром <code>ps</code> .
<code>PenStyle: integer;</code>	Возвращает текущий стиль пера.
<code>SetPenMode(m: integer);</code>	Устанавливает режим пера, задаваемый параметром <code>m</code> .
<code>PenMode: integer;</code>	Возвращает текущий режим пера. Режим пера определяет, как цвет пера взаимодействует с цветом поверхности.

Стили пера задаются следующими именованными константами :

<code>psSolid</code>	_____
<code>psClear</code>	
<code>psDash</code>	-----
<code>psDot</code>
<code>psDashDot</code>	- . - . - . - . - . - . - . - .
<code>psDashDotDot</code>	- . - . - . - . - . - . - . - .

Режимы пера задаются следующими именованными константами:

<code>pmCopy</code>	– обычный режим; при рисовании цвет поверхности заменяется цветом пера;
<code>pmNot</code>	– режим инвертирования; при рисовании цвет поверхности инвертируется (становится негативным), а цвет пера при этом игнорируется.

4. Действия с кистью

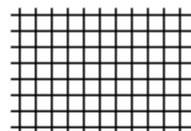
SetBrushColor (color: integer);	Устанавливает цвет кисти, задаваемый параметром color.
BrushColor : integer;	Возвращает текущий цвет кисти.
SetBrushPicture (fname: string);	Устанавливает в качестве образца для закраски кистью образец, хранящийся в файле fname, при этом текущий цвет кисти при закраске игнорируется.
ClearBrushPicture ;	Очищает рисунок-образец, выбранный для кисти.
SetBrushStyle (bs: integer);	Устанавливает стиль кисти, задаваемый параметром bs.
BrushStyle : integer;	Возвращает текущий стиль кисти.

Стили кисти задаются следующими именованными константами:

`bsSolid`



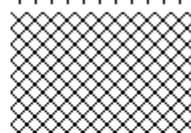
`bsCross`



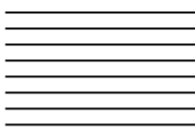
`bsClear`



`bsDiagCross`



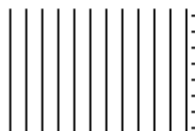
`bsHorizontal`



`bsBDiagonal`



`bsVertical`



`bsFDiagonal`



5. Действия со шрифтом

SetFontColor (color: integer);	Устанавливает цвет шрифта.
FontColor : integer;	Возвращает текущий цвет шрифта.
SetFontSize (sz: integer);	Устанавливает размер шрифта в пунктах.
FontSize : integer;	Возвращает текущий размер шрифта в пунктах.
SetFontName (name: string);	Устанавливает наименование шрифта.
FontName : string;	Возвращает текущее наименование шрифта.
По умолчанию установлен шрифт, имеющий наименование MS Sans Serif. Наиболее распространенные шрифты – это Times, Arial и Courier New. Наименование шрифта можно набирать без учета регистра.	
SetFontStyle (fs: integer);	Устанавливает стиль шрифта.
FontStyle : integer;	Возвращает текущий стиль шрифта.
TextWidth (s: string): integer;	Возвращает ширину строки s в пикселях при текущих настройках шрифта.
TextHeight (s: string): integer;	Возвращает высоту строки s в пикселях при текущих настройках шрифта.

Стили шрифта задаются следующими именованными константами:

fsNormal – обычный;

fsBold – жирный;

fsItalic – наклонный;

fsBoldItalic – жирный наклонный;

fsUnderline – подчеркнутый;

fsBoldUnderline – жирный подчеркнутый;

fsItalicUnderline – наклонный подчеркнутый;

fsBoldItalicUnderline – жирный наклонный подчеркнутый.

6. Действия с рисунками

Работа с рисунками в модуле GraphABC производится либо с помощью описателей (целых чисел, однозначно определяющих рисунок), либо с помощью объектов класса [Picture](#). Работа с помощью описателей позволяет манипулировать рисунками, используя обычные процедуры и функции и не используя классы.

Каждый рисунок, хранящийся во время работы программы в оперативной памяти, имеет описатель (дескриптор), представляющий собой целое число. Это число возвращается функцией загрузки рисунка из файла `LoadPicture`, а также функцией создания рисунка `CreatePicture`, после чего передается в качестве первого параметра во все остальные процедуры и функции работы с рисунками.

<code>LoadPicture (fname: string): integer;</code>	<code>n:=LoadPicture (fname)</code> – загружает рисунок из файла с именем <code>fname</code> в оперативную память и возвращает описатель рисунка в целую переменную <code>n</code> ; если файл не найден, то возникает ошибка времени выполнения. Загружать можно рисунки в формате <code>.bmp</code> , <code>.jpg</code> или <code>.gif</code> .
<code>DrawPicture (n, x, y: integer);</code>	Выводит рисунок с описателем <code>n</code> в позицию <code>(x, y)</code> графического окна.
<code>DrawPicture (n, x, y, w, h: integer);</code>	Выводит рисунок с описателем <code>n</code> в позицию <code>(x, y)</code> графического окна, масштабируя его размеры к ширине <code>w</code> и высоте <code>h</code> . Если <code>w<0</code> , то рисунок зеркально отражается относительно вертикальной прямой, проходящей через <code>x</code> , если же <code>h<0</code> , то рисунок зеркально отражается относительно горизонтальной прямой, проходящей через <code>y</code> .
<code>DrawPicture (n: integer; x, y: integer; r: Rect);</code>	Выводит часть рисунка с описателем <code>n</code> , заключенную в прямоугольнике <code>r</code> , в позицию <code>(x, y)</code> графического окна.
<code>DrawPicture (n: integer; x, y, w, h: integer; r: Rect);</code>	Выводит часть рисунка с описателем <code>n</code> , заключенную в прямоугольнике <code>r</code> , в позицию <code>(x, y)</code> графического окна, масштабируя его размеры к ширине <code>w</code> и высоте <code>h</code> . Если <code>w<0</code> , то рисунок зеркально отражается относительно вертикальной прямой, проходящей через <code>x</code> , если же <code>h<0</code> , то рисунок зеркально отражается относительно горизонтальной прямой, проходящей через <code>y</code> .
<code>CopyRect (n: integer; dest: Rect; n1: integer; src: Rect);</code>	Копирует часть рисунка с описателем <code>n1</code> , заключенную в прямоугольнике <code>src</code> , в прямоугольник <code>dest</code> рисунка с описателем <code>n</code> .
<code>DestroyPicture (n: integer);</code>	Удаляет рисунок с описателем <code>n</code> из оперативной памяти, описатель рисунка при этом становится недействительным.
<code>SetPictureSize (n, w, h: integer);</code>	Устанавливает размер рисунка с описателем <code>n</code> равным <code>w</code> на <code>h</code> пикселей.
<code>PictureWidth (n: integer): integer;</code>	Возвращает ширину рисунка с описателем <code>n</code> .
<code>PictureHeight (n: integer): integer;</code>	Возвращает высоту рисунка с описателем <code>n</code> .

Приложение к уроку мастер класса : «Урок по графике PascalABC»

PictureTransparent (n: integer): boolean;	Возвращает режим прозрачности рисунка с описателем n.
SetPictureTransparent (n: integer; b: boolean);	Устанавливает (b=True) или отключает (b=False) режим прозрачности при рисовании рисунка с описателем n. Если b=True, то при его рисовании фон не отображается. Фоновым считается цвет левого нижнего пиксела рисунка.