Положение каждой точки изображения задано координатами X и Y . Координаты – целые числа, они задают номера колонки и строки растра и не зависят от физического размера экрана. Оси координат направлены следующим образом: горизонтальная ось X направлена слева направо; вертикальная ось Y направлена сверху вниз; верхний левый угол имеет координаты (0,0).

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

Множество графических процедур и функций среды программирования Pascal собраны в модуле Graph . Для подключения библиотеки графических функций и процедур необходимо подключить модуль к вашей программе строкой : **uses** graphabc;

SetWindowWidth(w) -Устанавливает ширину графического окна;   
SetWindowHeight(h) -Устанавливает высоту графического окна;

type **ColorType**=integer; - стандартные цвета задаются символическими константами:

|  |  |  |
| --- | --- | --- |
| **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 |  | **clAqua** – бирюзовый  **clOlive** – оливковый  **clFuchsia** – сиреневый  **clTeal** – сине-зеленый  **clGray** – темно-серый  **clLime** – ярко-зеленый  **clMoneyGreen** – цвет зеленых денег  **clLtGray** – светло-серый  **clDkGray** – темно-серый **clMedGray** – серый **clSilver** – серебряный |

**Установка цвета фона.**

Очистка графического окна ClearWindow; - очищает графическое окно белым цветом. ClearWindow(color); - очищает графическое окно указанным цветом. Если процедура установки цвета фона не вызвана, экран будет ,белым.

SetPixel(x,y,color) - закрашивает один пиксель с координатами (x,y) цветом color.

Различные команды для работы с пером:

SetPenStyle(<номер от 1 до 6>); - устанавливает стиль пера, задаваемый номером. program prim; uses GraphABC; SetPenStyle(1 - длинный штрих), SetPenStyle (2 - короткий штрих) Line(10,125,350,125); SetPenStyle(3 - штрих- пунктир).

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

function **PenColor**: integer; - возвращает текущий цвет пера.

procedure **SetPenWidth**(w: integer); - устанавливает **ширину** **пера**, равную w пикселам.

**Рисование линий**

Процедура **Line** ( x1, y1, x2, y2: integer) вычерчивает линию между двумя точками экрана с координатами ( x1, y1) и ( x2, y2).

Процедура **LineTo** ( x, y: integer) вычерчивает линию от последнего положения указателя до точки с координатами ( x, y).

**Окружность, эллипс, дуга, сектор**

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

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

**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).

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

**Прямоугольник**

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

**Вывод текста в графическом режиме.**

Процедура TextOut ( x, y: integer; text: string) выводит строку символов text, начиная с точки с координатами ( x, y), при этом указатель своего положения не меняет, т.е. остается в точке ( x, y).

**Заполнение (закрашивание) произвольной замкнутой фигуры**

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

Заливка кистью SetBrushColor(color) – устанавливает цвет кисти. Заливка кистью распространяется на замкнутый контур, описание которого следует за процедурой установки цвета кисти.

Пример программы:

**program** traktor;

**uses** graphabc;

**begin**

ClearWindow(clOlive);

setwindowsize (400,300);

setwindowtitle ('Бульдозер');

setpencolor (clgreen);

setpenwidth (4);

rectangle(40,150,300,200);

rectangle(150,60,300,200);

rectangle(80,110,100,150);

circle(70,230,20);

circle(270,230,10);

line(70,210,280,210);

setfontcolor(clred);

setfontstyle(fsbold);

textout (20,20,'Ну допустим трактор.');

**end**.

sleep(ms) - задержка (пауза) на ms миллисекунд.

**Процедура LockDrawing;**

Рисование средствами PascalABC происходит очень медленно. Пока оно идет, картинка на экране успевает много раз обновиться. Отсюда мерцание. То есть, мы видим не окончательную картинку, а процесс ее рисования. Окончательную картинку в общем-то не видим — как только она построилась, мы ее тут же стираем и начинаем новый цикл рисования. Для борьбы с этим эффектом придумали двойную буферизацию. Идея в том, чтобы сначала построить картинку в памяти (не выводя на экран), а затем отобразить новый кадр на экране целиком, без промежуточных вариантов, где не хватает половины лучиков. Для этого в начале программы вызываем процедуру LockDrawing.

**Процедура ClearWindow (color);**

После каждой прохода по циклу очищает графическое окно указанным цветом, который передается в качестве параметра.

После этого все команды рисования будут создавать картинку в памяти (в так называемом вторичном буфере). Непосредственно на экране ничего рисоваться не будет. После того, как очередной кадр анимации полностью отрисован (то есть перед ClearWindow), нужно вызвать процедуру

**Redraw;**

Эта процедура вызовет перерисовку всей картины.

Пример анимации:

**uses** GraphABC;  
**begin**  
LockDrawing;  
**for var** i:=1 to 500 do  
**begin**  
[Window[http://cdncache-a.akamaihd.net/items/it/img/arrow-10x10.png](http://s.igmhb.com/click?v=Qlk6MTI3OTUxOjIxOTIyOndpbmRvd3M6ZjQ2NjZmMjFiNTVhOTljN2ZjYzEwYmEyZTNmZGU4ZmI6ei0yNDQ5LTg3ODIzMDY5OndtLWhlbHAubmV0OjM3Mjk5MDowOjRjZTM2YTM1MjBiMjRhMzY4NDcwYjc3OWI4YWE4ZDgxOjA6ZGF0YV9zcywxMDA2eDE2ODA7ZGF0YV9yYywxO2RhdGFfZmIsbm87OjQ0MDA3NTE6OjowLjAy&subid=g-87823069-3fc083322ed94f47bea28648c06eab49-&data_ss=1006x1680&data_rc=1&data_fb=no&data_tagname=A&data_ct=link_only&data_clickel=link&data_sid=90e66411aa93fae87f1463090dfaac92)](http://s.igmhb.com/click?v=Qlk6MTI3OTUxOjIxOTIyOndpbmRvd3M6ZjQ2NjZmMjFiNTVhOTljN2ZjYzEwYmEyZTNmZGU4ZmI6ei0yNDQ5LTg3ODIzMDY5OndtLWhlbHAubmV0OjM3Mjk5MDowOjRjZTM2YTM1MjBiMjRhMzY4NDcwYjc3OWI4YWE4ZDgxOjA6ZGF0YV9zcywxMDA2eDE2ODA7ZGF0YV9yYywxO2RhdGFfZmIsbm87OjQ0MDA3NTE6OjowLjAy&subid=g-87823069-3fc083322ed94f47bea28648c06eab49-&data_ss=1006x1680&data_rc=1&data_fb=no&data_tagname=A&data_ct=link_only&data_clickel=link&data_sid=90e66411aa93fae87f1463090dfaac92).Clear;  
Brush.Color := clGreen;  
Ellipse(i,100,i+100,200);  
Redraw;  
Sleep(1);  
**end**;  
**end**.

http://progmatem.ru/grafika-v-pascal.html