Работа с графикой в PascalABC

После запуска PascalABC, по умолчанию, запускается текстовый режим. Для работы с графикой служит отдельное графическое окно.

Чтобы его открыть, необходимо подключить модуль GraphABC. В этом модуле содержится набор процедур и функций, предназначенных для работы с графическим экраном, а также некоторые встроенные константы и переменные, которые могут быть использованы в программах с графикой.

С их помощью можно создавать разнообразные графические изображения и сопровождать их текстовыми надписями.

Подключение осуществляется в разделе описаний.

Формат подключения модуля GraphABC:

Uses GraphABC;

Графический экран PascalABC (по умолчанию) содержит 640 точек по горизонтали и 400 точек по вертикали. Начало отсчета – левый верхний угол экрана. Ось x направлена вправо, а ось y –вниз. Координаты исчисляются в пикселях.

Все команды библиотеки GraphABC являются подпрограммами и описаны в виде процедур и функций. Для того, что бы команда выполнилась необходимо указать команду и задать значения параметров.

Управление графическим окном

Процедура/Функция	Описание
SetWindowWidth(w);	Устанавливает ширину графического окна
SetWindowHeight(h);	Устанавливает высоту графического окна
ClearWindow;	Очищает графический экран белым цветом.
clearWindow(color);	Очищает графическое окно указанным цветом.

Процедуры рисования графических примитивов

Процедура	Описание
SetPixel(x, y, color);	Закрашивает один пиксел с координатами x, y цветом color
Line (x1, y1, x2, y2);	Рисут отрезок прямой ограниченный точками с координатами (x1, y1) и (x2, y2)
LineTo(x, y)	Рисует отрезок из текущей точки в точку с координатами (x, y)
Rectangle(x1, y1, x2, y2);	Рисует контур прямоугольника, левая верхняя вершина которого имеет координаты (x1, y1), правая нижняя – (x2, y2)
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)

Процедуры, используемые для работы с цветом

Процедура	Описание
SetPenColor(color);	Устанавливает цвет пера (линий).
SetPenWidth(n);	Устанавливает ширину (толщину) пера, равную n пикселям.
SetPenStyle(ps);	Устанавливает стиль пера, задаваемый параметром ps. Стили пера задаются следующими именованными константами: **Значение** / **Описание** / **Вид линии** psSolid / Сплошная линия psDash / Штриховая линия psDot / Пунктирная линия psDashDot / Штрихпунктирная линия psDashDotDot / Линия - штрих и два пунктира psClear / Отсутствие линии
SetBrushColor(color);	Устанавливает цвет кисти. Заливка кистью распространяется на замкнутый контур, описание которого следует за процедурой установки цвета кисти.
FloodFill(x, y, color);	Закрашивает в выбранном стиле замкнутый контур, x, y – координаты точки внутри контура, color – цвет контура.
FillRect(x1, y1, x2, y2);	Закрашивает прямоугольник, заданный координатами противоположных вершин (x1, y1) и (x2, y2), текущим цветом кисти.

Процедуры для работы с текстом

Процедура/Функция	Описание
TextOut(x, y, St);	Выводит строку текста St в заданную координатами x и y позицию
SetFontColor(color)	Устанавливает цвет шрифта
SetFontSize(cz);	Устанавливает размер шрифта в пунктах.
SetFontName(name);	Устанавливает наименование шрифта. По умолчанию устанавливается шрифт, имеющий название MS Sans Serif. Наиболее распространенные шрифты: Times, Arial, Courier New.
SetFontStyle(fs)	Устанавливает текущий стиль 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.**

Цвета в PascalABC

```
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     -          серебристый
clCream     -  кремовый
```

Пример графической программы, рисующей изображение дома:

```
1. Program Domik;
2. uses Graphabc; //подключение модуля GraphABC
3. begin
4. SetWindowWidth(800); //ширина окна программы
5. SetWindowHeight(600); //высота окна программы
6. SetPenWidth(3); //толщина пера
7. SetFontSize(18); //размер шрифта
8. SetFontColor(clRed); //цвет шрифта
9. SetFontStyle(fsBold); //жирный стиль шрифта
10.      TextOut(100,100,'Домик'); //текст
11.      Rectangle(200, 300, 600,600); //дом
```

```
12.       Circle(400,225, 40); //круг
13.       SetBrushColor(clAqua); //цвет заливки окна
14.       FillRect(300, 400,500,500); //процедура заливки окна
15.       rectangle(300,400,500,500); //окно
16.       Line(400,400,400,500); // окно
17.       Line(300,450,500,450); //окно
18.       Line(200,300,400,150); //крыша
19.       Line(400,150,600,300); //крыша
20.       Line(480,210,480,160); //труба
21.       Line(480,160,520,160); //труба
22.       Line(520,160,520,240); //труба
23.       end.
```

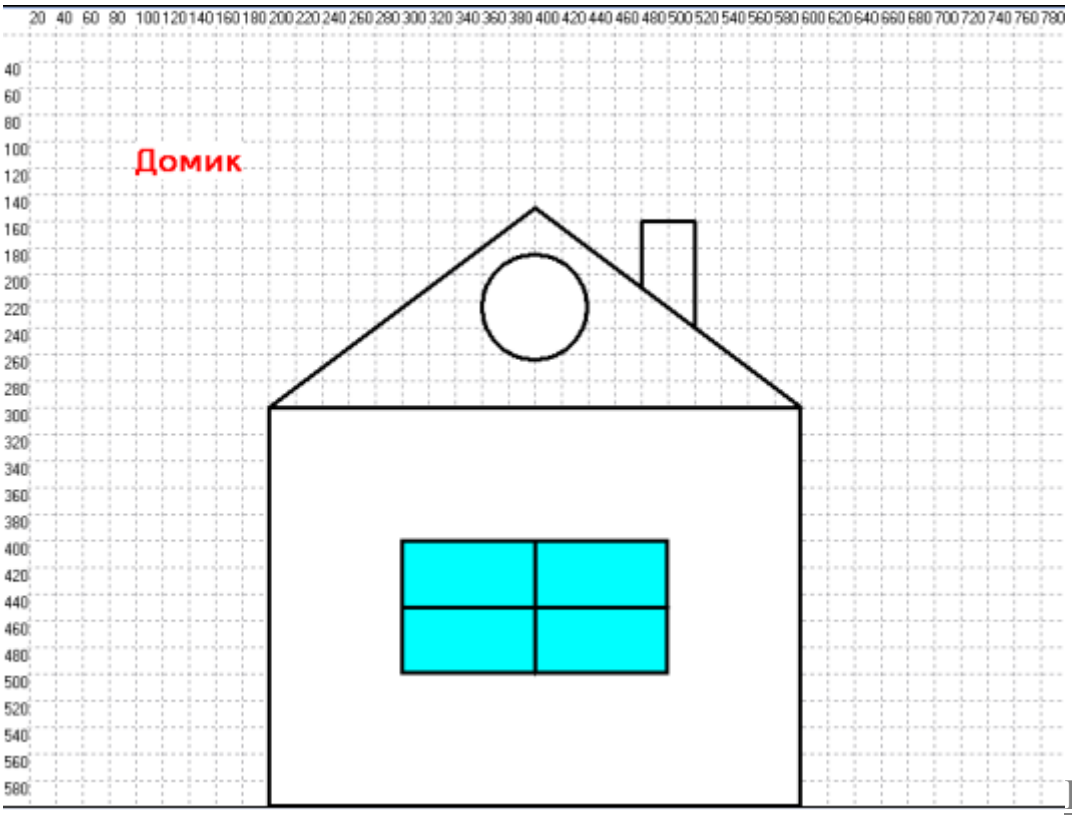
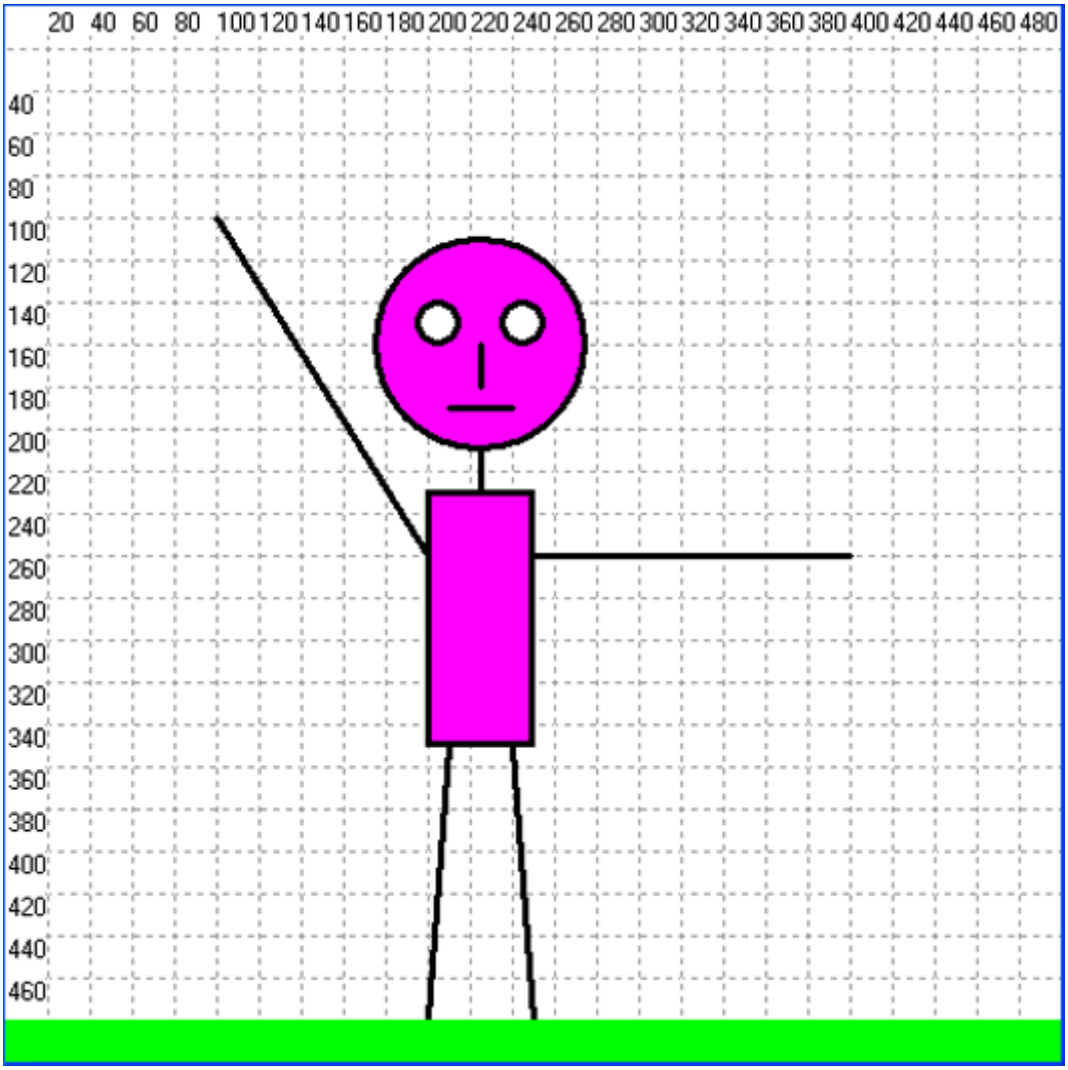


Рисунок в PascalABC

Программа, рисующая фигурку:

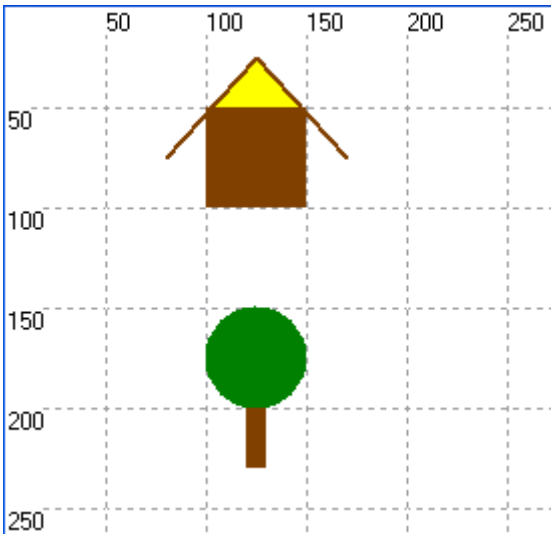
```
1. Program Figurka;
2. uses GraphABC;
3. begin
4. SetWindowSize(500, 500); //задаем размер графического окна
5. SetPenWidth(3); //устанавливаем стиль пера
6. SetBrushColor(clFuchsia); //устанавливаем цвет кисти
```

```
7. Circle(225,160,50); //рисуем окружность
8. Line(225,160,225,180); //рисуем линии
9. Line( 210,190,240,190);
10.     Line( 225,210,225,250);
11.     Line( 100,100,200,260);
12.     Line( 200,260,400,260);
13.     Line( 210,350,200,480);
14.     Line(240,350,250,480);
15.     Rectangle(200,230,250,350); //рисуем прямоугольник
    SetBrushColor(clLime);
16.     FillRect(0, 480,500,500); //рисуем закрашенный
    прямоугольник
17.     SetBrushColor(clWhite);
18.     Circle(205,150,10);
19.     Circle(245,150,10);
20.     end.
```



Практическая работа за компьютером

Задание 1. Определите координаты и составьте программу, выводящую на экран рисунк дома и дерева.

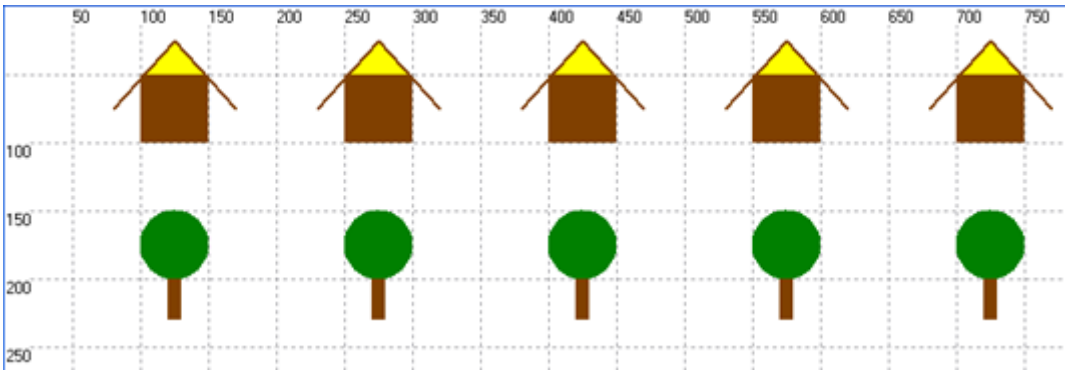


Программа будет иметь вид:

```
1. Program domik_2;
2. uses GraphABC;
3. var i: integer;
4. st:string;
5. begin
6. SetWindowSize(500, 500);
7. 
8. {Коричневая стена}
9. SetPenWidth(2);
10.     SetBrushColor(clBrown);
11.     FillRect(100,50,150,100);
12.     
13.     {Крыша желтого цвета}
14.     SetPenColor(clBrown);
15.     Line(125,25,80,75);
16.     Line(125,25,170,75);
17.     FloodFill(125, 30,clYellow);
18.     
19.     {Крона дерева}
20.     SetPenColor(clGreen);
21.     SetBrushColor(clGreen);
22.     Ellipse(100,150,150,200);
```

```
23.____
24.     {ствол дерева}
25.     SetBrushColor(clBrown);
26.     FillRect(120,200,130,230);
27.     end.
```

Задание 2. Используя оператор цикла и введя переменную для пересчета координат по оси x, постройте «поселок», состоящий из 5 домов. Внесите соответствующие дополнения и изменения в предыдущую программу.



Весь наш «поселок» выстроился вдоль горизонтальной оси экрана — оси X. Построение рисунка начинается с левого верхнего угла стены первого дома — точки с координатами (100, 50). Координата Y не изменяется. Чтобы начать рисовать второй домик, нужно координату X увеличить на 150 (50 точек — ширина первого дома и 100 точек — расстояние между домиками).

Выберем в качестве параметра цикла целочисленную переменную X.
Для всех элементов нашего рисунка абсолютное значение координаты X заменим на относительное. Например, для стены дома процедура для рисования запишется следующим образом:

```
FillRect(x,50,x+50,100);
```

Сформулируем условие выполнения цыклических действий для нашей задачи.Какие координаты имеет левый верхний угол пятого дома? Конечное значение выбранного нами параметра цикла x = 700. Тогда условие выполнения цикла записывается так: x<=700.

Словесное описание алгоритма коротко можно записать так:

Переменной цикла x присвоить начальное значение 100. Пока x<=700 выполнять серию действий «Нарисуй дом и дерево», после каждого фрагмента рисунка увеличивать значение переменной цикла на 150.

Программа будет иметь вид:

```
1. Program Domik;
2. uses GraphABC;
3. var i: integer;
4. st:string;
5. x:integer;
6. begin
7. SetWindowSize(800, 500);
8. x:=100;
9. While x<=700 do
10.       begin
11.       {Коричневая стена}
12.       SetPenWidth(2);
13.       SetBrushColor(clBrown);
14.       FillRect(x,50,x+50,100);
15.
16.       {Крыша желтого цвета}
17.       SetPenColor(clBrown);
18.       Line(x+25,25,x-20,75);
19.       Line(x+25,25,x+70,75);
20.       FloodFill(x+25, 30,clYellow);
21.       {Крона дерева}
22.       SetPenColor(clGreen);
23.       SetBrushColor(clGreen);
24.       Ellipse(x,150,x+50,200);
25.
26.       {ствол дерева}
27.       SetBrushColor(clBrown);
28.       FillRect(x+20,200,x+30,230);
29.       x:=x+150;
30.       end;
31.       end.
```