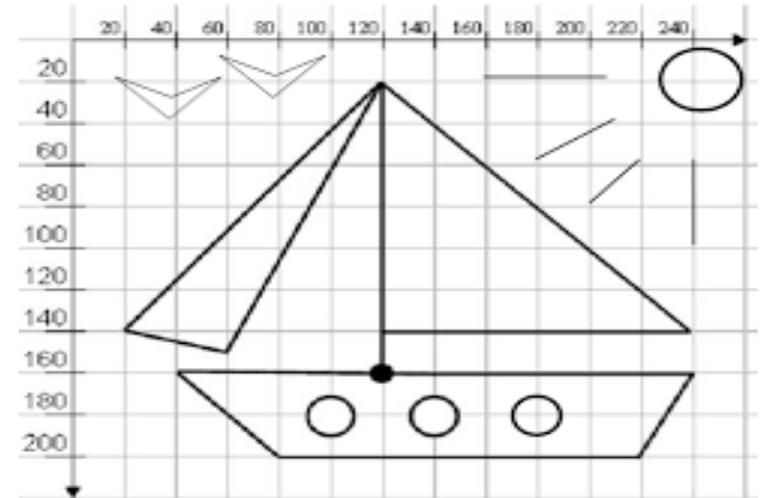


PASCAL

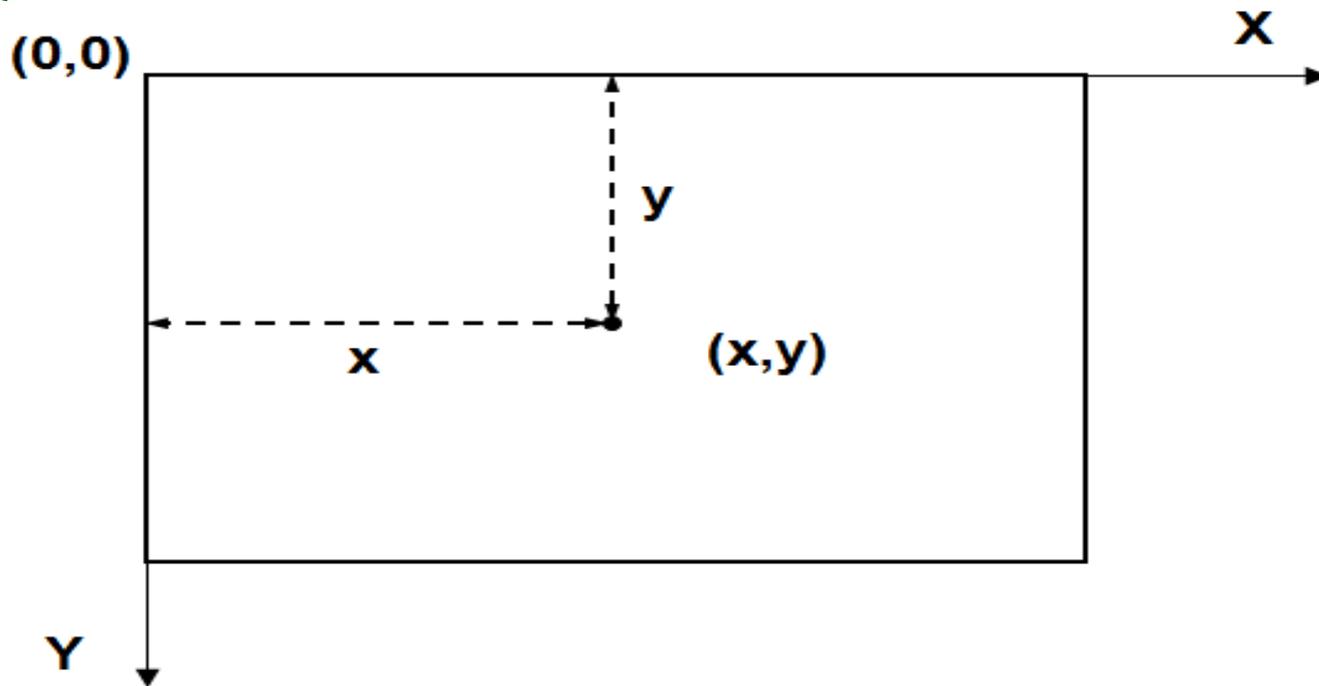
Графика в Паскале ABC



Для работы с графикой в pascal abc используется модуль **GraphABC**.

Система координат в Паскале соответствует экранной системе координат и выглядит следующим образом:

```
uses  
GraphABC;  
  
begin  
  
...  
  
end.
```



Управление размером экрана (окна)

SetWindowWidth(w) — устанавливает ширину графического окна

SetWindowHeight(h) — устанавливает высоту графического окна

Очистка графического окна

ClearWindow; - очищает графическое окно белым цветом.

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

Пример программы очистки экрана:

```
program clear;  
uses GraphABC;  
begin  
  ClearWindow;  
  ClearWindow (clGreen);  
end.
```

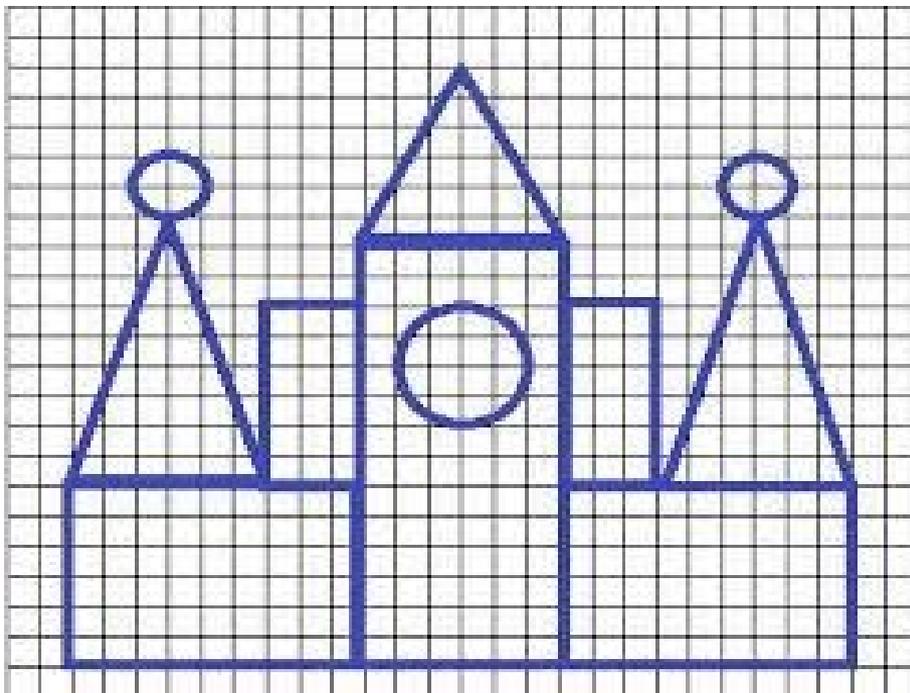


Установка графического окна

Для установки размеров графического окна используется процедура `SetWindowSize(ширина, высота)`, например:

```
SetWindowWidth(600);
```

```
SetWindowHeight(400);
```



Цвета в Паскаль ABC

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

clAqua – бирюзовый
clOlive – оливковый
clFuchsia – сиреневый
clTeal – сине-зеленый
clGray – темно-серый
clLime – ярко-зеленый
clLtGray – светло-серый
clDkGray – темно-серый
clMedGray – серый
clSilver – серебряный

Графические примитивы

1. Точка

2. Линия

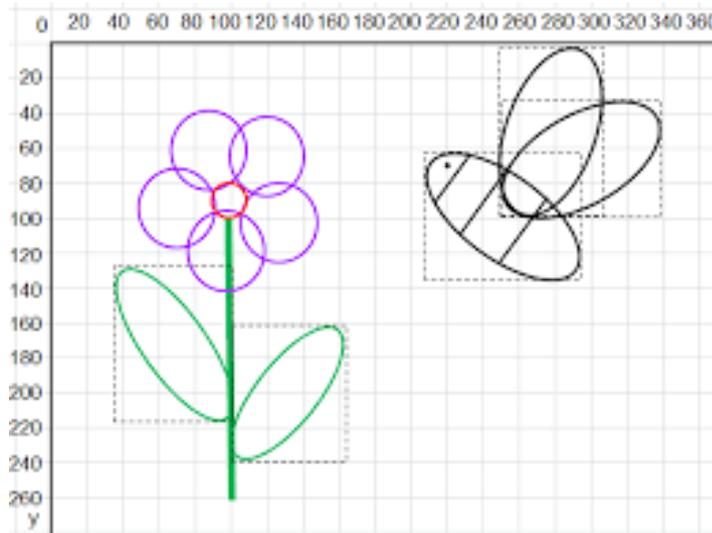
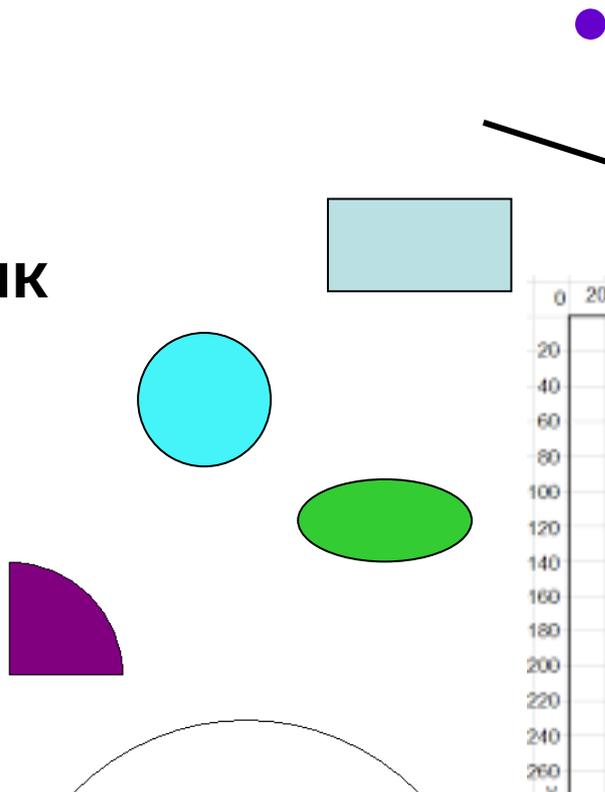
3. Прямоугольник

4. Окружность

5. Эллипс

6. Сектор

7. Дуга



Точка

Для отображения точки в паскале используется процедура:

SetPixel(x,y,color) — закрашивает один пиксел с координатами (x,y) цветом color точки в Паскале

Программа:

```
uses GraphABC;
```

```
begin
```

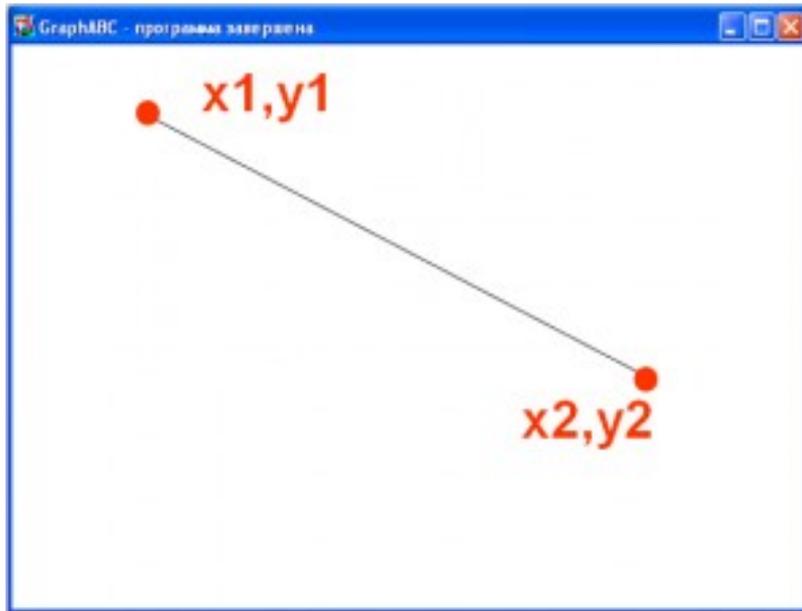
```
    SetPixel(300, 200, clred);
```

```
end.
```

Линия

Line(x1,y1,x2,y2) — рисует отрезок с началом в точке (x1,y1) и концом в точке (x2,y2)

Программа:

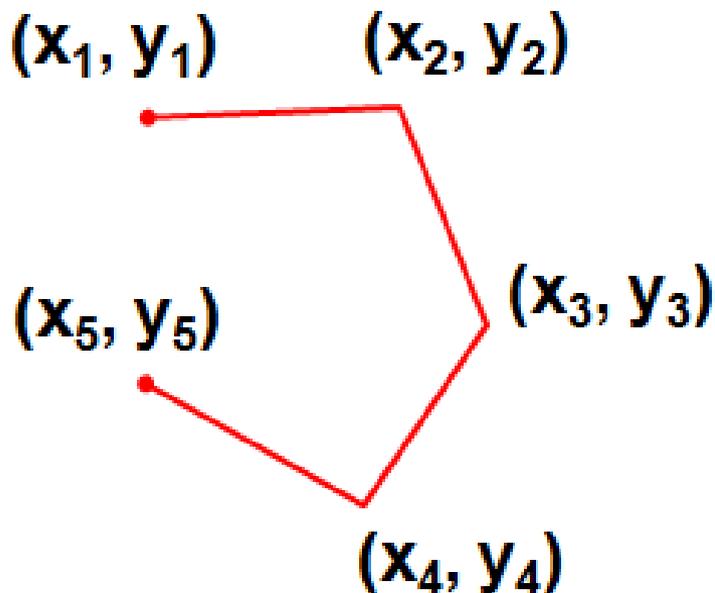


```
Program liniay;  
uses GraphABC;  
begin  
    line(100,50,500,250);  
end.
```

Ломаные линии

Процедуры **MoveTo (x1, y1)** и **LineTo (x2, y2)**.

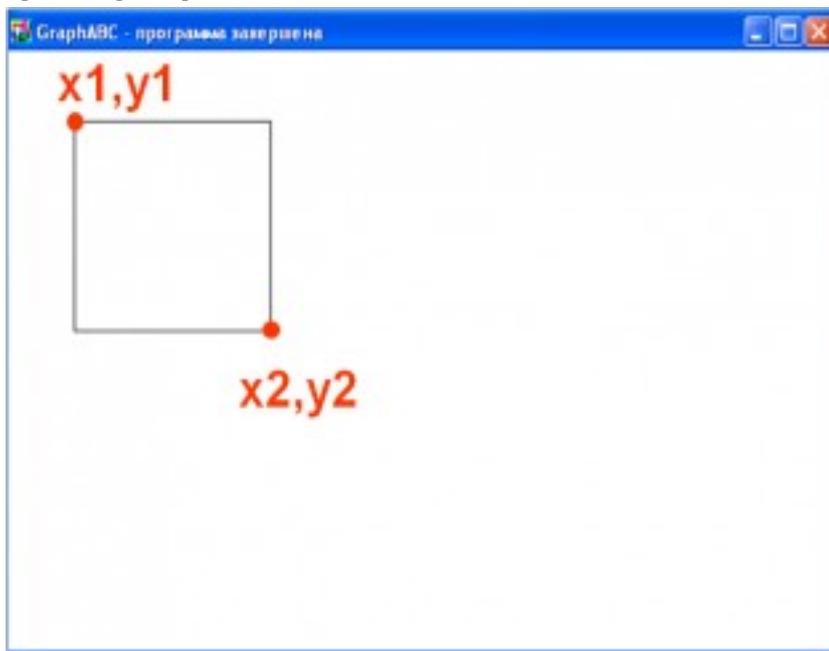
Процедуры работают в паре: **MoveTo** передвигает курсор в определенную точку, а процедура **LineTo** рисует линию с этой точки до точки, определенной параметром данной процедуры.



```
SetPenColor (clred) ;  
MoveTo (x1, y1) ;  
LineTo (x2, y2) ;  
LineTo (x3, y3) ;  
LineTo (x4, y4) ;  
LineTo (x5, y5) ;
```

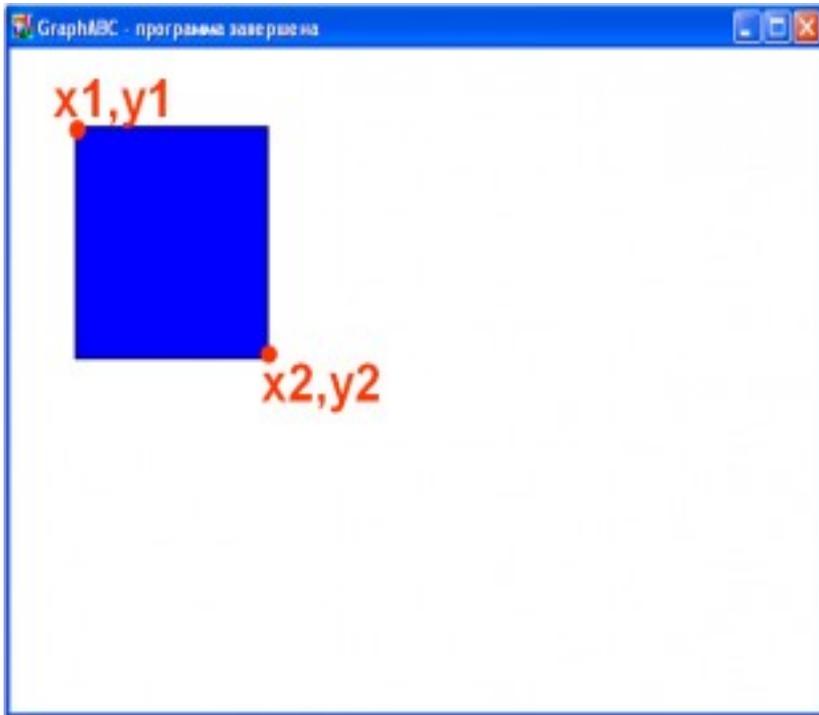
Рисование фигур

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



```
Program pryamougolnik;  
uses GraphABC;  
begin  
    Rectangle(50, 50, 200, 200);  
end.
```

Прямоугольник с заливкой

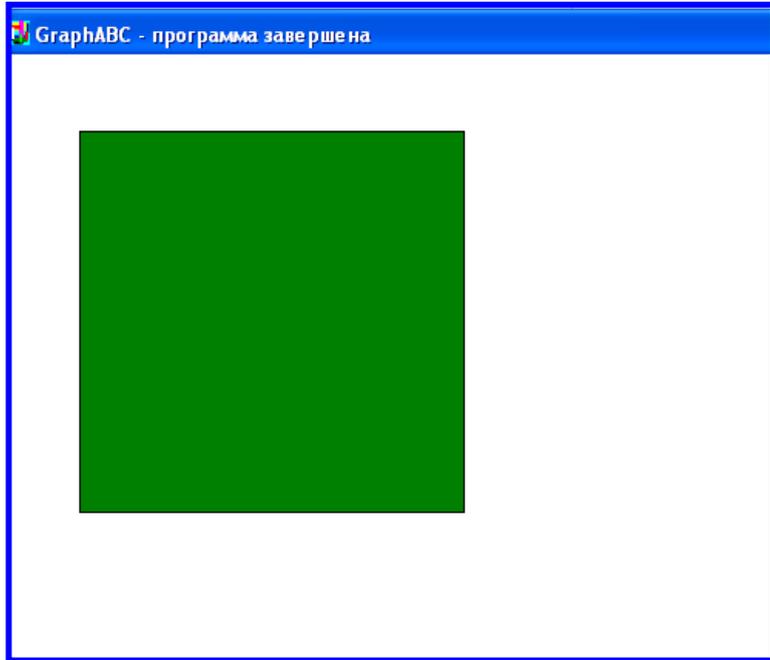


```
Program pryamougolnik;  
uses GraphABC;  
begin  
    Rectangle (50, 50, 200, 200);  
    FloodFill (100, 100, clBlue);  
end.
```

Заливка кистью

SetBrushColor(color) – устанавливает цвет кисти.

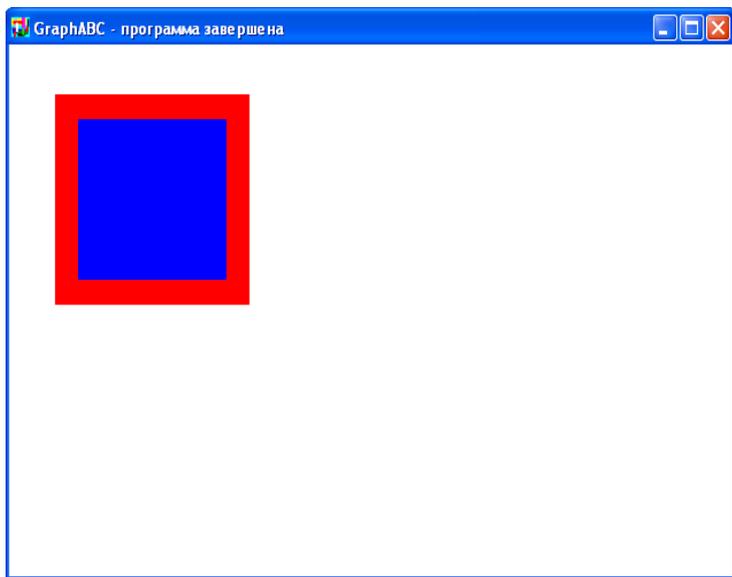
Заливка кистью распространяется на замкнутый контур, описание которого следует за процедурой установки цвета кисти.



```
Program zalivka_kist;  
uses GraphABC;  
Begin  
    SetBrushColor(c1Green);  
    Rectangle(50, 50, 300, 300);  
end.
```

Цвет и толщина контура

Задаются процедурами: **SetPenWidth(w)** и **SetPenColor(color);**

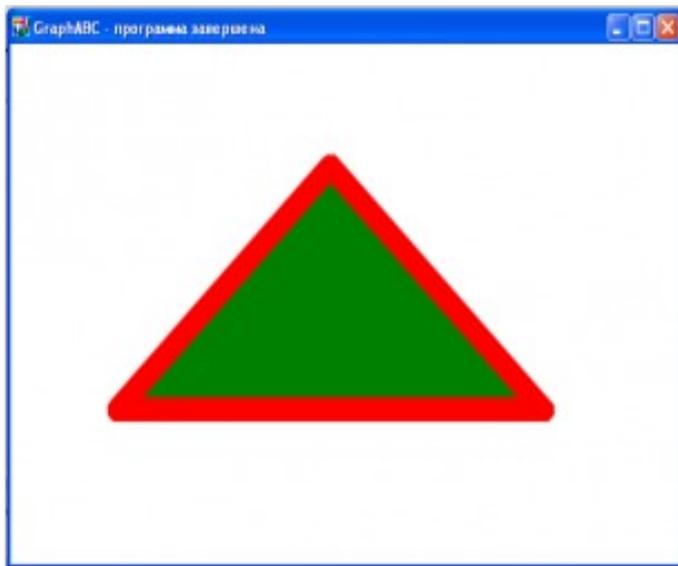


```
Program pryamougolnik;  
uses GraphABC;  
begin  
    SetPenColor(clred);  
    SetPenWidth(20);  
    Rectangle(50, 50, 200, 200);  
    FloodFill(100, 100, clBlue);  
end.
```

Треугольник

Line(x1,y1,x2,y2);

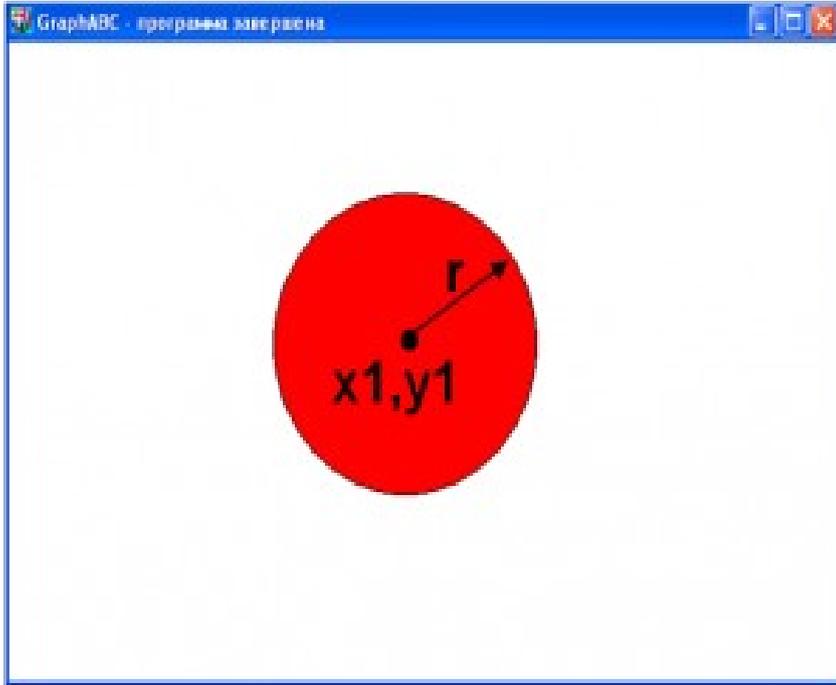
LineTo(x,y);



```
Program treugolnik;  
uses GraphABC;  
begin  
    setpenwidth(20);  
    setpencolor(clred);  
    line(300,100,500,300);  
    lineto(100,300);  
    lineto(300,100);  
    floodfill(300,200,clgreen);  
end.
```

Окружность

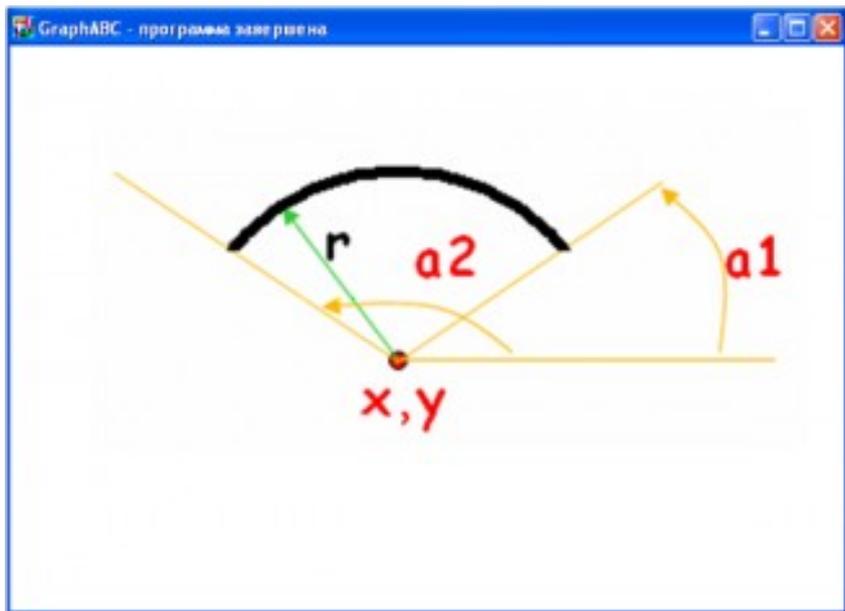
Circle(x,y,r) — рисует окружность с центром в точке (x,y) и радиусом r.



```
Program circle;  
uses GraphABC;  
begin  
    Circle(500,200,100);  
    FloodFill(500,200,clred);  
end.
```

Дуга окружности

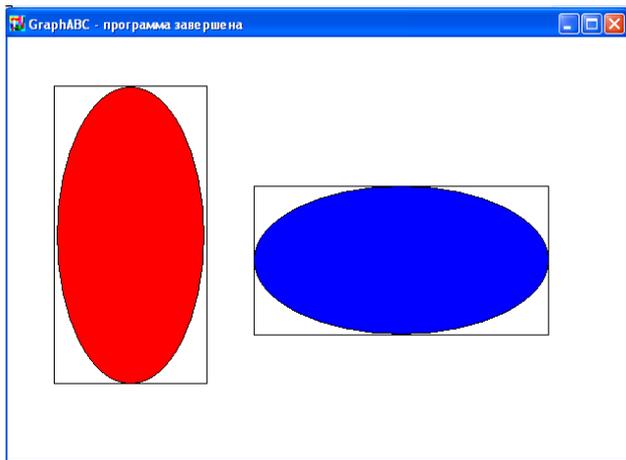
Arc(x,y,r,a1,a2) — рисует дугу окружности с центром в точке (x,y) и радиусом r, заключенной между двумя лучами, образующими углы a1 и a2 с осью OX (a1 и a2 – вещественные, задаются в градусах и отсчитываются против часовой стрелки).



```
Program duga;  
uses GraphABC;  
Begin  
SetPenWidth(10);  
Arc(300,250,150,45,135);  
end.
```

Эллипс

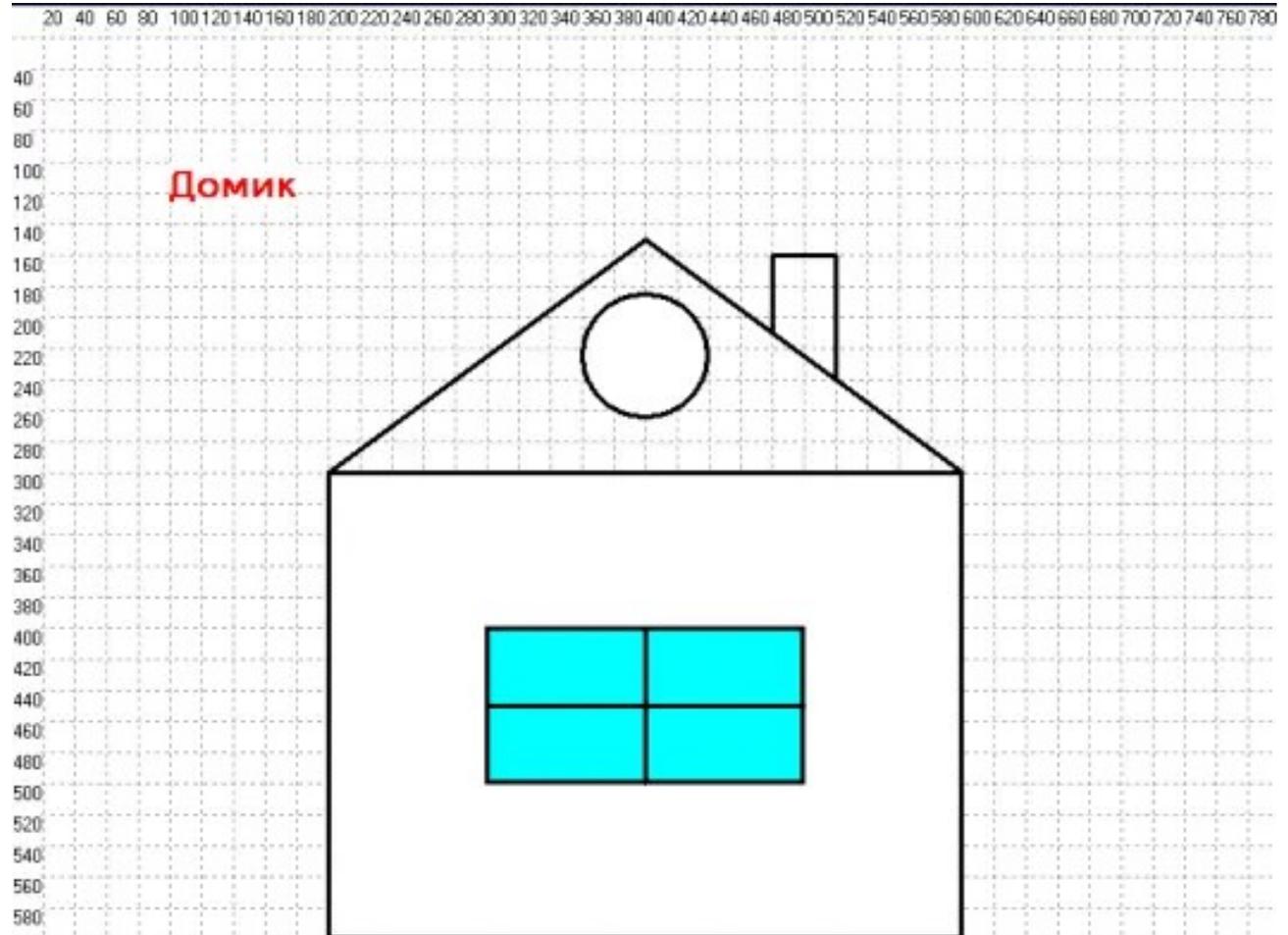
Ellipse(x1,y1,x2,y2) - рисует эллипс, заданный своим описанным прямоугольником с координатами противоположных вершин.



```
Program oval;  
uses GraphABC;  
begin  
    Ellipse(50, 50, 200, 350);  
    FloodFill(50+100, 50+100, clRed);  
    Ellipse(250, 150, 550, 300);  
    FloodFill(250+100, 150+100, clBlue);  
end.
```

Задание 1. Домик

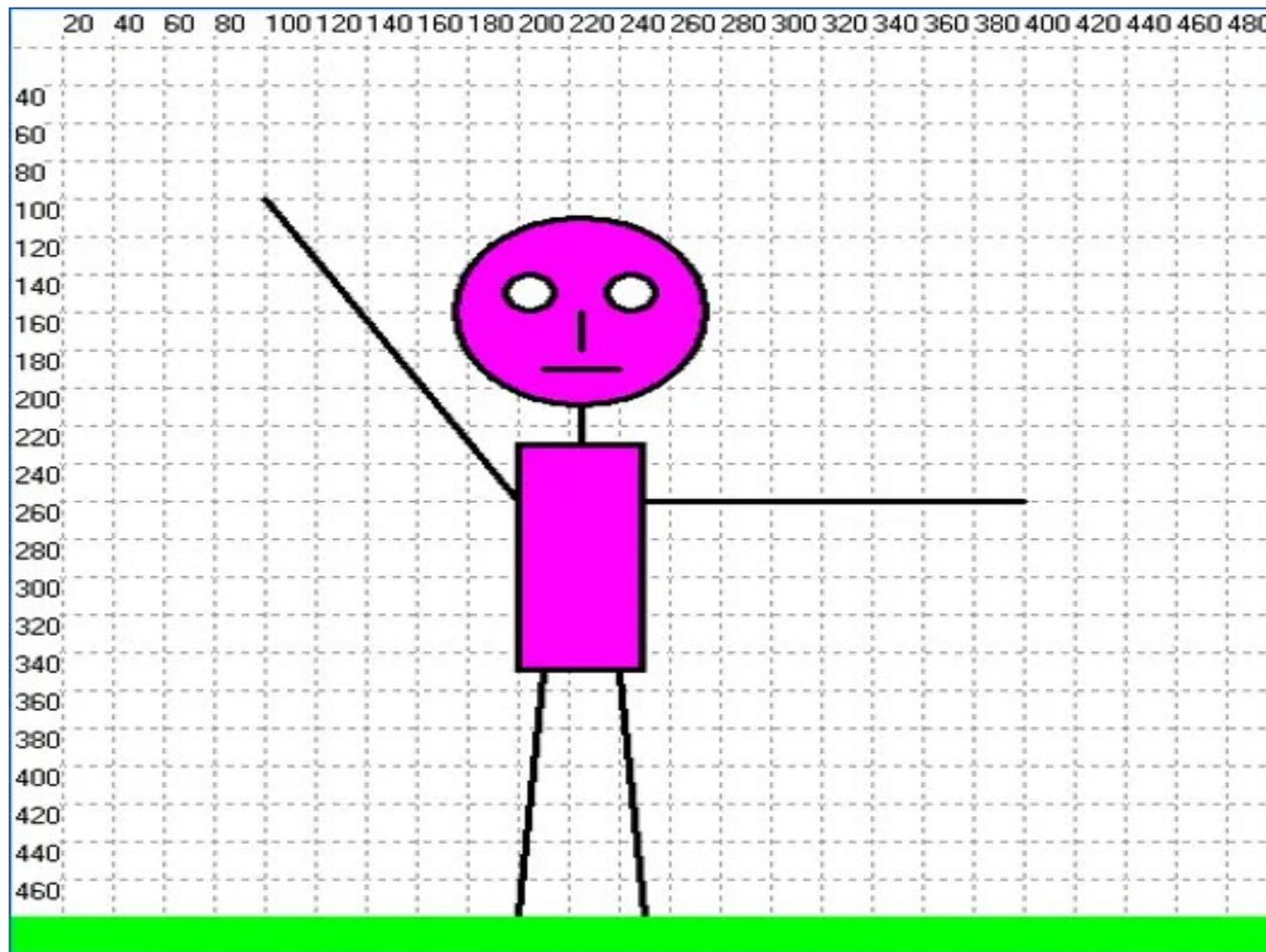
Нарисовать домик:



```
Program Domik;
uses Graphabc; //подключение модуля
begin
  SetWindowWidth(800); //ширина окна
  SetWindowHeight(600); //высота окна
  SetPenWidth(3); //толщина пера
  SetFontSize(18); //размер шрифта
  SetFontColor(clRed); //цвет шрифта
  SetFontStyle(fsBold); //жирный стиль
  TextOut(100,100,'Домик'); //текст
  Rectangle(200, 300, 600,600); //дом
  Circle(400,225, 40); //круг
```

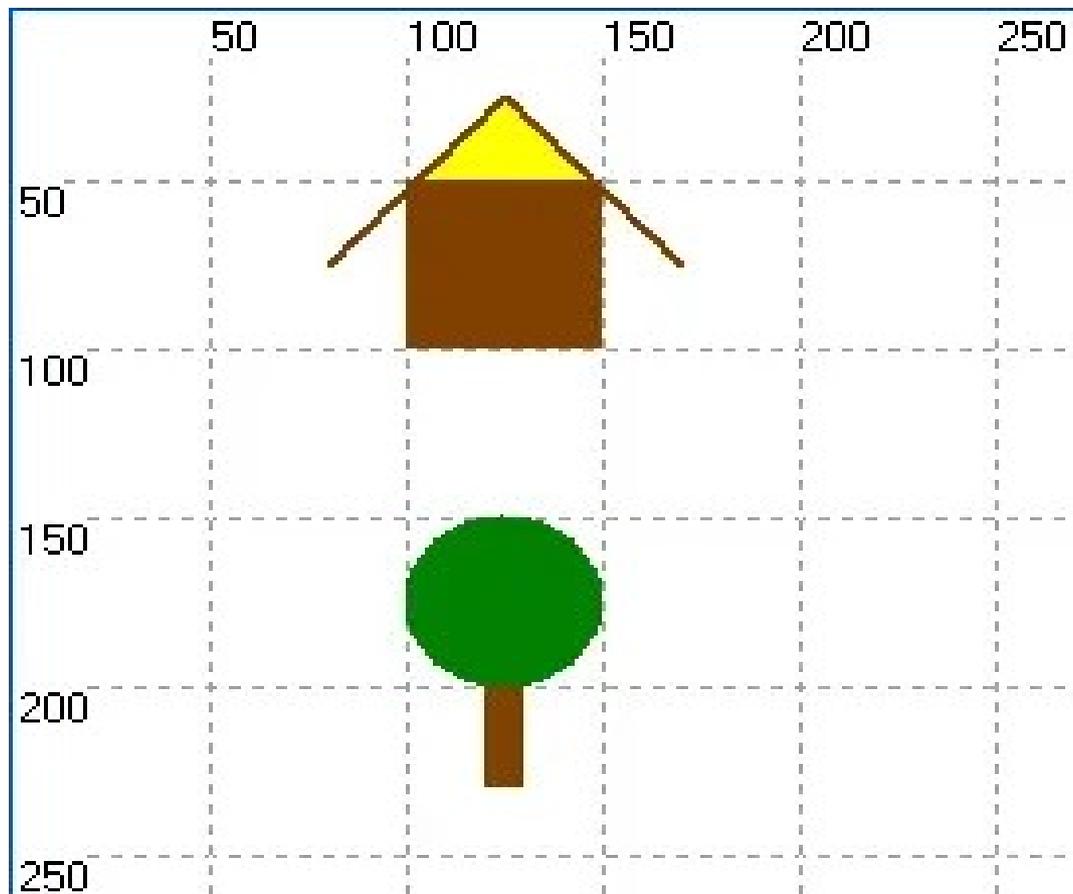
```
//SetBrushColor(clBlue; //цвет заливки
окна
  FillRect(300, 400,500,500);
//процедура заливки окна
  rectangle(300,400,500,500); //окно
  Line(400,400,400,500); // окно
  Line(300,450,500,450); //окно
  Line(200,300,400,150); //крыша
  Line(400,150,600,300); //крыша
  Line(480,210,480,160); //труба
  Line(480,160,520,160); //труба
  Line(520,160,520,240); //труба
end.
```

Задание 2. Фигурка



```
Program Figurka;
  uses GraphABC;
  begin
    SetWindowSize(500, 500); //задаем
размер графического окна
    SetPenWidth(3); //устанавливаем стиль
пера
    SetBrushColor(clFuchsia);
//устанавливаем цвет кисти
    Circle(225,160,50); //рисуем окружность
    Line(225,160,225,180); //рисуем линии
    Line( 210,190,240,190);
    Line( 225,210,225,250);
    Line( 100,100,200,260);
    Line( 200,260,400,260);
    Line( 210,350,200,480);
    Line(240,350,250,480);
    Rectangle(200,230,250,350); //рисуем
прямоугольник SetBrushColor(clLime);
    FillRect(0, 480,500,500); //рисуем
закрашенный прямоугольник
    SetBrushColor(clWhite);
    Circle(205,150,10);
    Circle(245,150,10);
  end.
```

Задание 3. Определите координаты и составьте программу, выводящую на экран рисунок дома и дерева.



```
Program domik_2;
uses GraphABC;
var i: integer;
st:string;
begin
  SetWindowSize(500, 500);
  {Коричневая стена}
  SetPenWidth(2);
  SetBrushColor(clBrown);
  FillRect(100,50,150,100);
  {Крыша желтого цвета}
  SetPenColor(clBrown);
  Line(125,25,80,75);
  Line(125,25,170,75);
  FloodFill(125, 30,clYellow);
```

```
{Крона дерева}
  SetPenColor(clGreen);
  SetBrushColor(clGreen);
  Ellipse(100,150,150,200);
  {ствол дерева}
  SetBrushColor(clBrown);
  FillRect(120,200,130,230);
end.
```

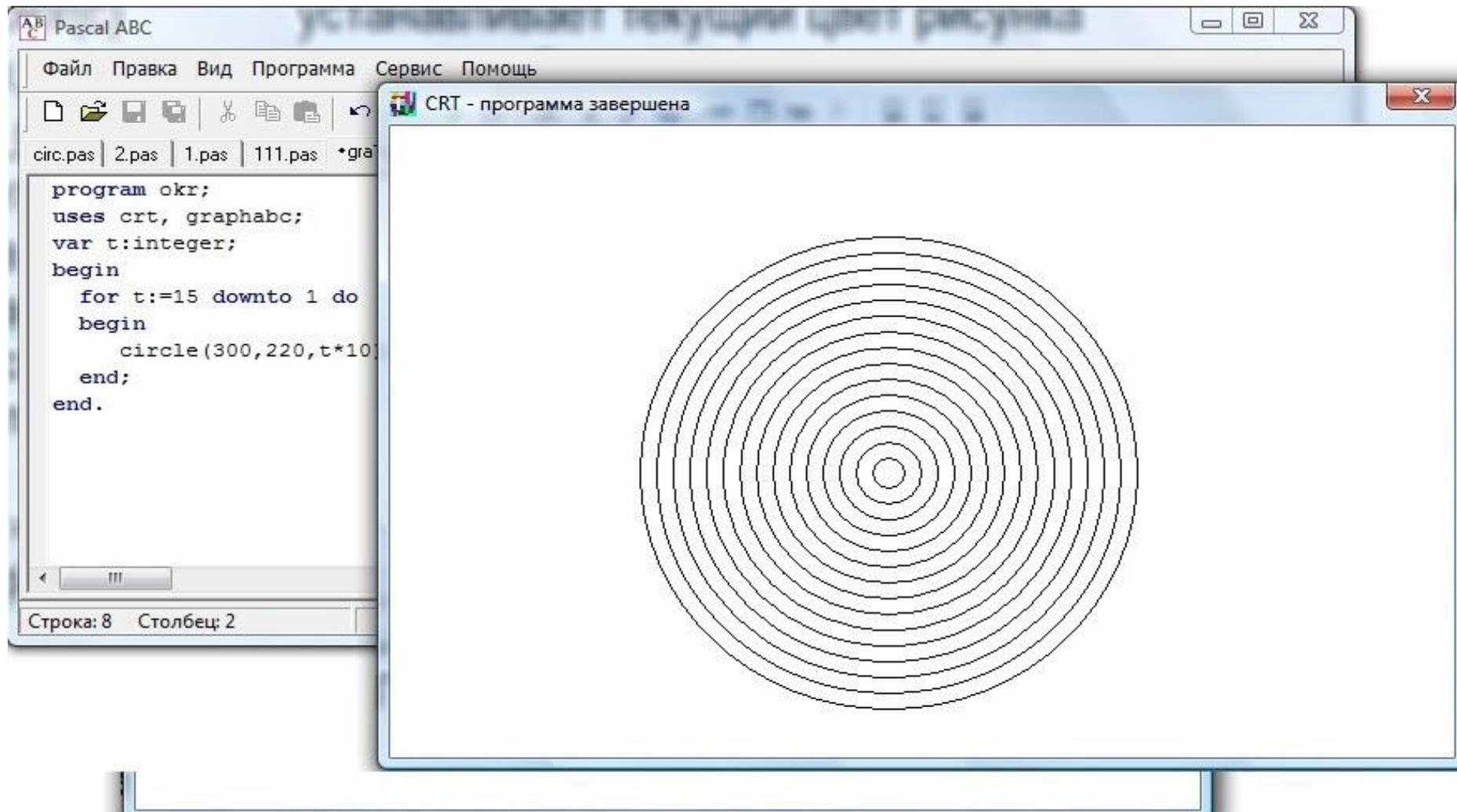
Пример программы: начертить на экране **15 окружностей** уменьшающегося радиуса, чтобы радиус был в 10 раз больше номера окружности.

Алгоритм:

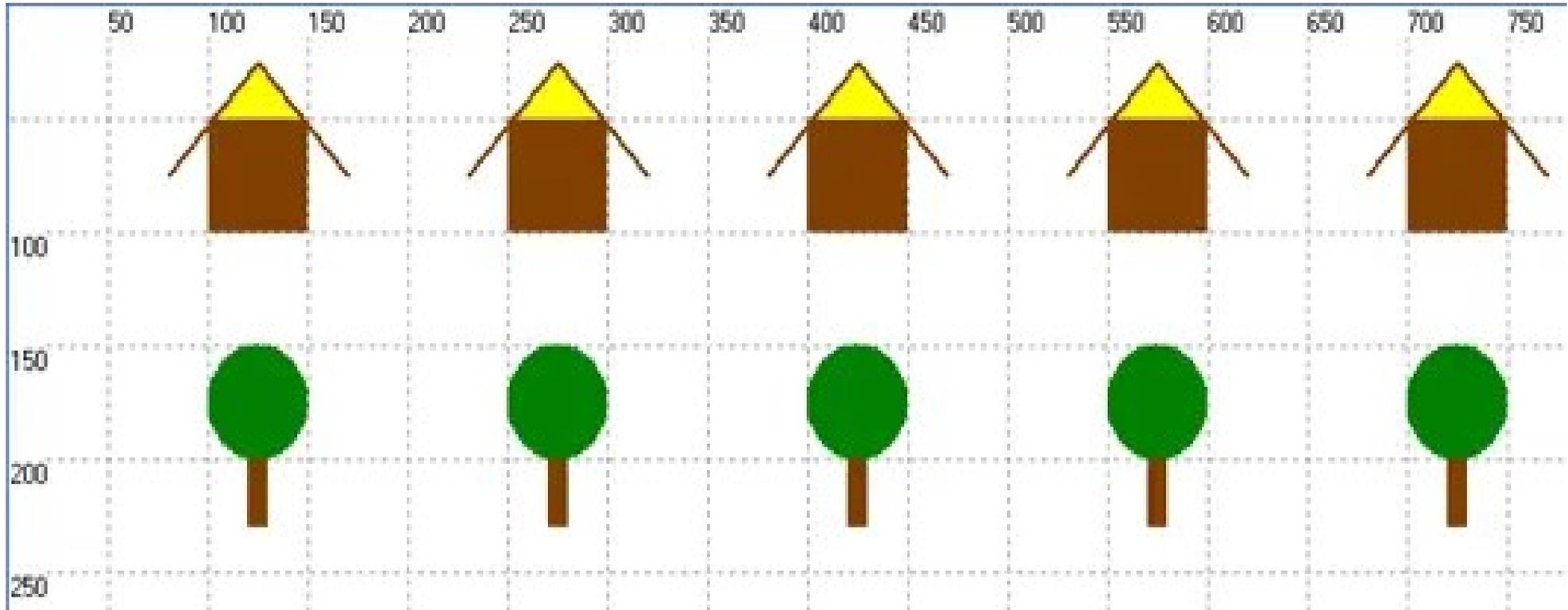
- Для рисования 15 окружностей нужно использовать цикл.
- В цикле радиус новой окружности увеличивается **в 10 раз — $t*10$** , а **центр окружностей (300, 200) не изменяется**

```
program okr;  
uses crt, graphabc;  
var t:integer;  
begin  
  for t:=15 downto 1 do  
    circle(300,220,t*10);  
  end.
```

Пример программы:



Задание 4. 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.



Алгоритм решения

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);**

Алгоритм решения

Сформулируем условие выполнения циклических 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: $x \leq 700$.

Словесное описание алгоритма:

- переменной цикла x присвоить начальное значение **100**;
- **пока $x \leq 700$** выполнять серию действий:
 - ✓ **«Нарисуй дом и дерево»;**
 - ✓ **после каждого фрагмента рисунка увеличивать значение переменной цикла на 150.**

Пример программы

```
Program Domik;  
uses GraphABC;  
var i: integer;  
st:string;  
x:integer;  
begin  
    SetWindowSize(800, 500);  
    x:=100;  
    While x<=700 do  
        begin  
            {Коричневая стена}  
            SetPenWidth(2);  
            SetBrushColor(c1Brown);  
            FillRect(x, 50, x+50, 100);
```

```
                {Крыша желтого цвета}  
                SetPenColor(c1Brown);  
                Line(x+25, 25, x-20, 75);  
                Line(x+25, 25, x+70, 75);  
                FloodFill(x+25, 30, c1Yellow);  
                {Крона дерева}  
                SetPenColor(c1Green);  
                SetBrushColor(c1Green);  
                Ellipse(x, 150, x+50, 200);  
                {ствол дерева}  
                SetBrushColor(c1Brown);  
                FillRect(x+20, 200, x+30, 230);  
                x:=x+150;  
            end;  
        end;  
    end.
```