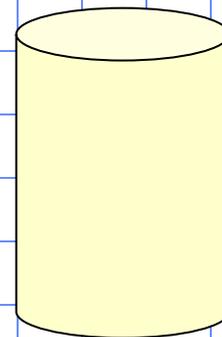
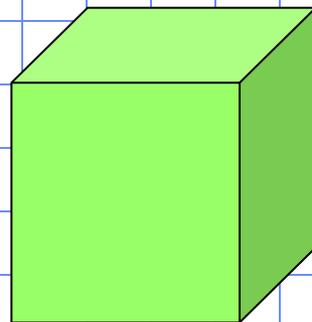
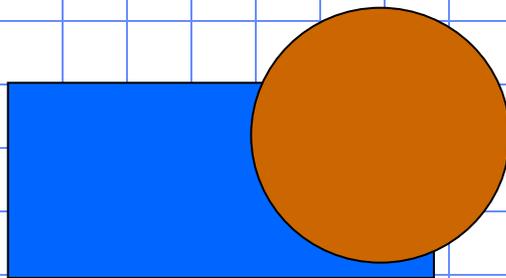
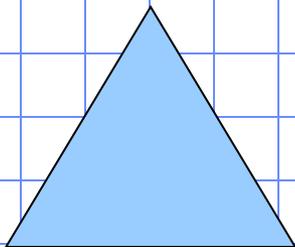


ГРАФИКА ЯЗЫКА ПАСКАЛЬ



Режимы работы монитора:

Текстовый (crt) –

на экране 25 строк по 80 символов в строке.

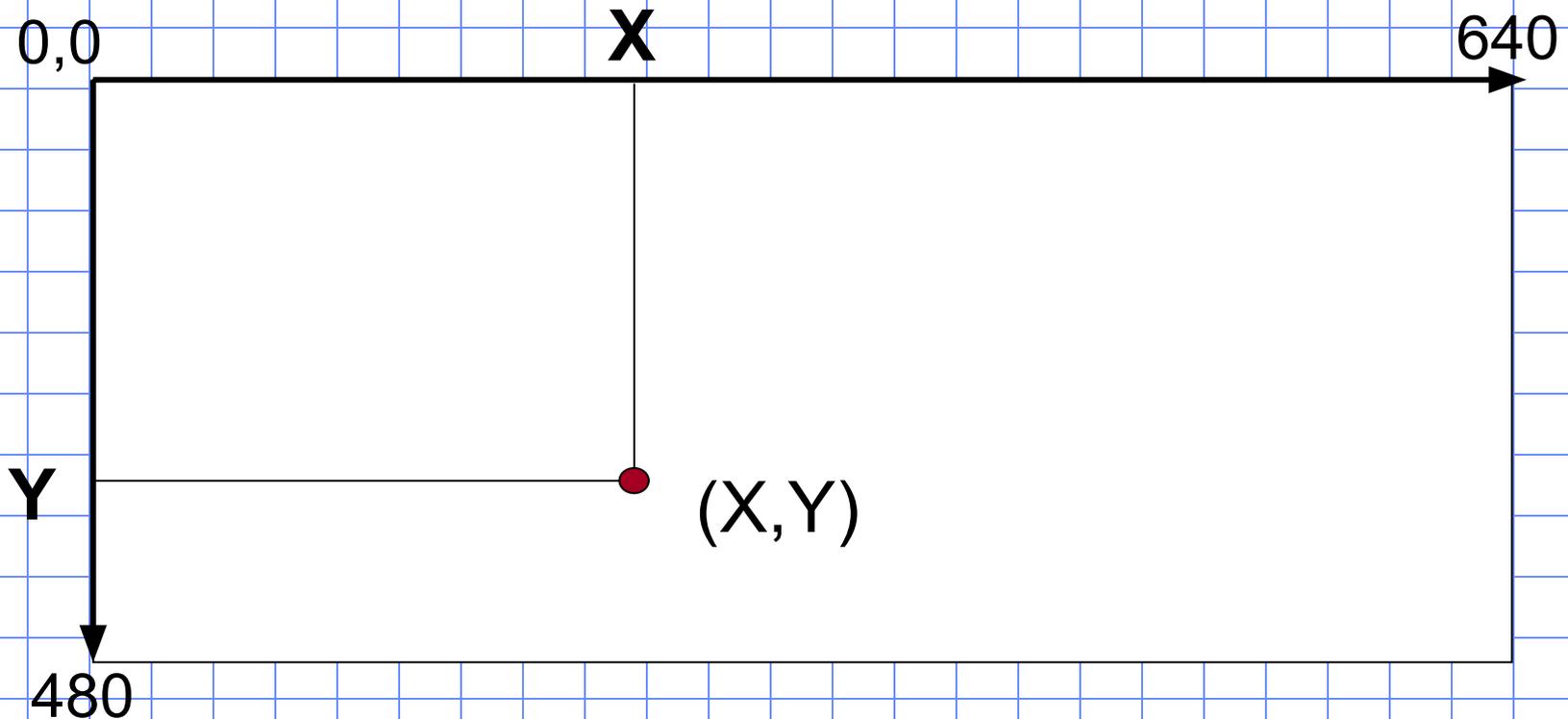
Минимальный элемент вывода – символ.

Графический (graph) –

на экране точки. В Паскале разрешение экрана 640*480. Каждая точка может быть одним из 16-ти цветов.

Минимальный элемент вывода – пиксель.

Графический экран



Координаты измеряются в пикселях

Шаблон графической программы:

Uses Graph;

Var

gd, gm: integer; {переменные, определяющие графический драйвер и монитор}

Begin

gd:=detect; {автоматический режим настройки графического драйвера}

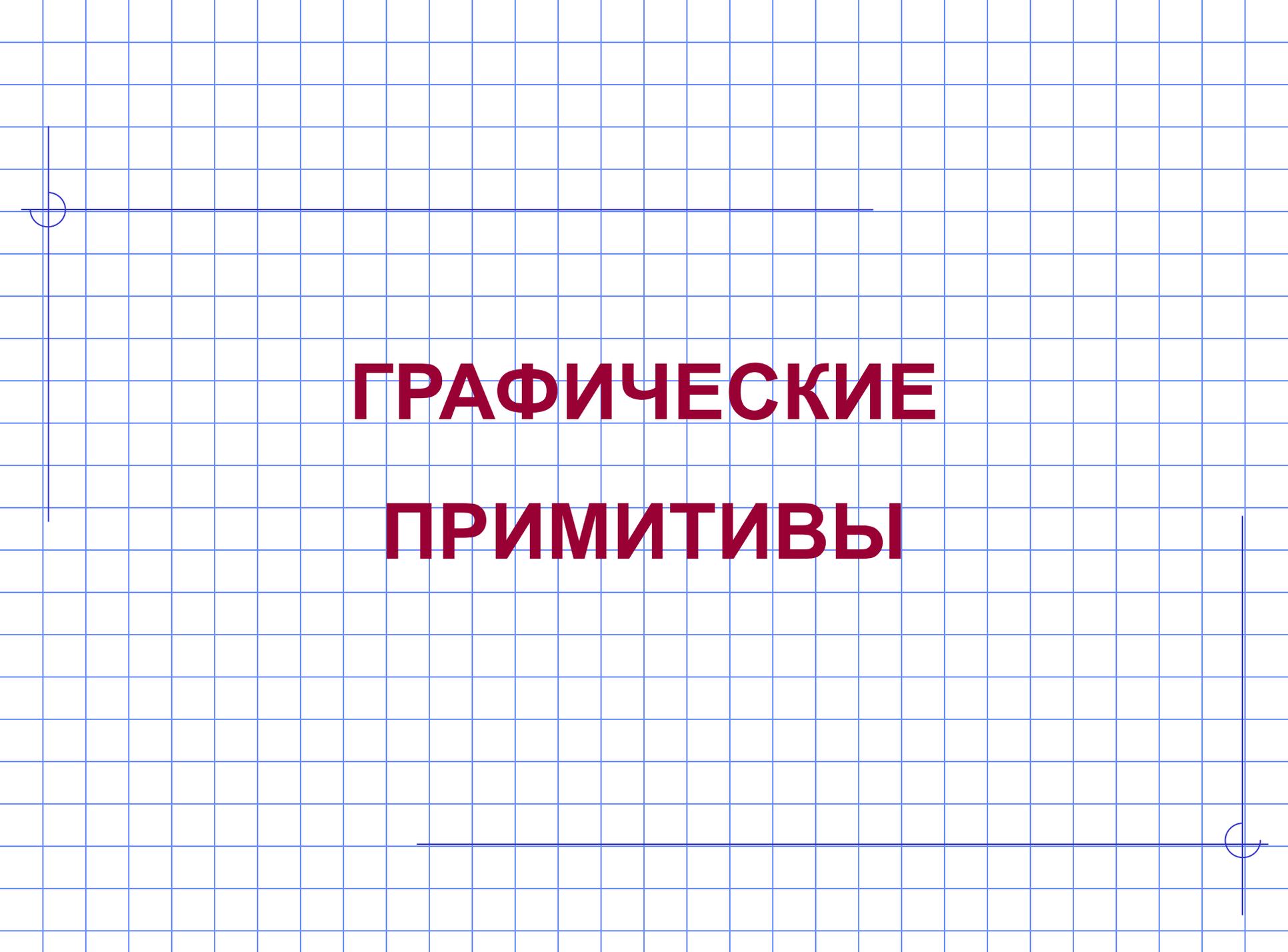
initGraph (gd, gm, ""); {задание графического режима}

Операторы графических изображений

readkey; {задержка изображения на мониторе}

closegraph; {закрытие графического режима}

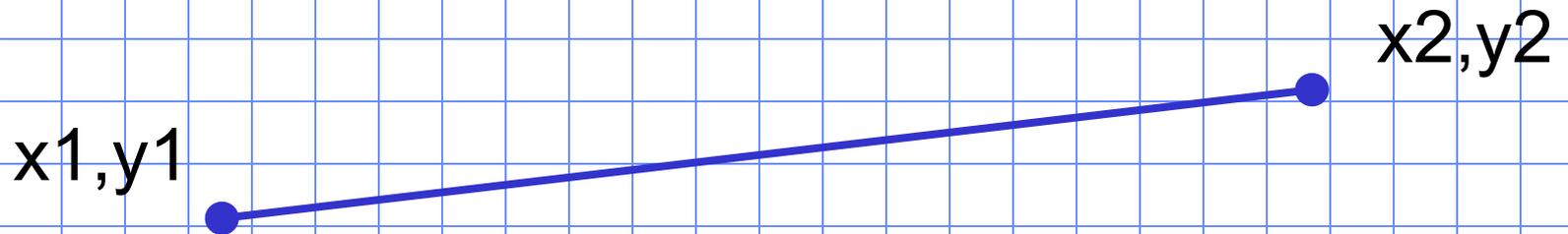
End.



ГРАФИЧЕСКИЕ ПРИМИТИВЫ

Процедура прорисовки линии

Line (x1, y1, x2, y2)



Пример Line (30, 45, 100, 142)

Стили линии

SetLineStyle (Style, 0, Wid)

Style – устанавливает тип линии

Wid – устанавливает толщину линии

тип линии (style) – устанавливается, исходя из таблицы:

Значение	Тип
0	непрерывная линия 
1	линия из точек 
2	линия из точек и тире 
3	штриховая линия 

Толщина линии (wid) – устанавливается исходя из таблицы

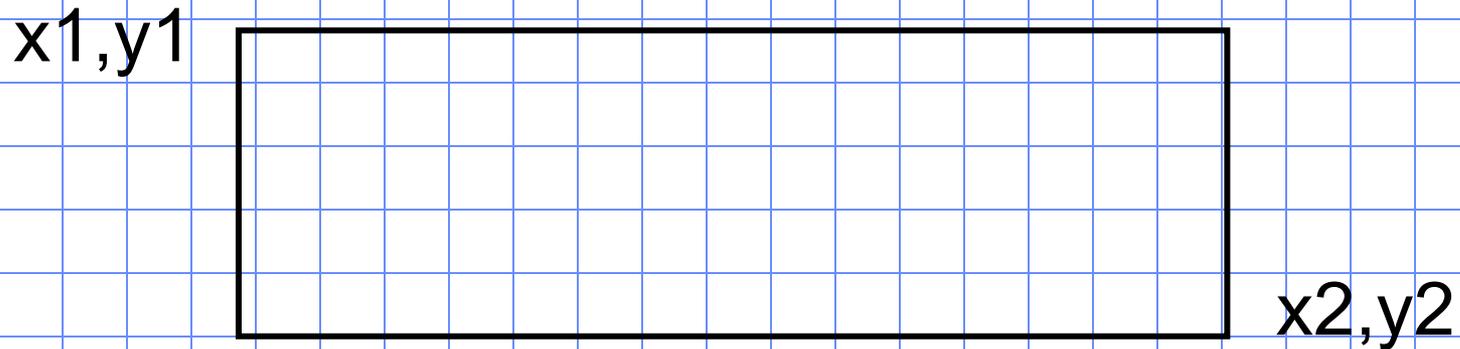
Значение	Толщина
1	Нормальная толщина (1 пиксел)
3	Жирная линия (3 пиксела)

Пример

```
SetLineStyle (3, 0, 1);  
Line (20, 30, 130, 30);
```

Процедура прорисовки прямоугольника

Rectangle (x1, y1, x2, y2)



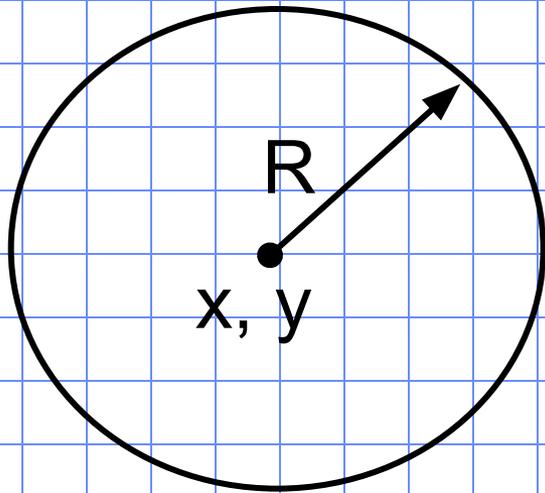
Пример `rectangle (30, 45, 100, 142)`

Процедура прорисовки окружности

Circle (x, y, R)

x, y — координаты центра окружности

R — радиус окружности



Пример

Circle (30, 45, 100)

Процедура прорисовки эллипса

Ellipse ($x, y, \alpha, \beta, R_x, R_y$)

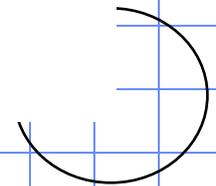
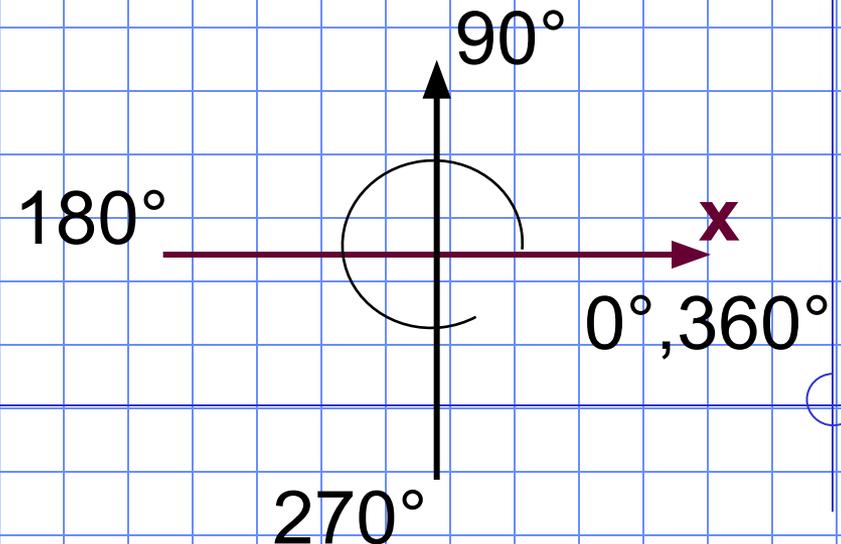
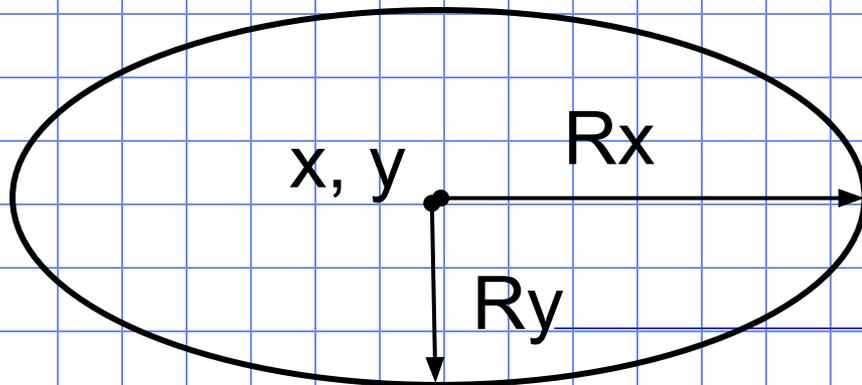
x, y — координаты центра эллипса;

α - начальный угол

β - конечный угол

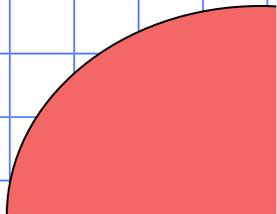
R_x — горизонтальный радиус эллипса

R_y - вертикальный радиус эллипса



Sector(x,y, α , β , Rx, Ry);

Выводит закрашенный эллиптический сектор с центром в точке (x,y) полуосями Rx, Ry и от начального угла α до угла β



```
SetFillStyle(1,6);
```

```
Sector(300,250,90,180,50,50);
```

Цветовая гамма

Цвета в операторах задаются с помощью своих кодов:

- 0 – черный;
- 1 – синий;
- 2 –зеленый;
- 3 – бирюзовый;
- 4 – красный;
- 5 – пурпурный;
- 6 – коричневый;
- 7 – светло-серый;
- 8 – темно-серый;
- 9 – ярко-синий;
- 10 – ярко-зеленый;
- 11 – ярко-бирюзовый;
- 12– ярко-красный;
- 13 – ярко-пурпурный;
- 14 – желтый;
- 15 – белый.



Установка цвета рисунка

Цвет графических примитивов задается
с помощью процедуры

SetColor (цвет)

цвет — код цвета

ЗАЛИВКА ЗАМКНУТЫХ ОБЛАСТЕЙ

Процедура определения типа заливки областей

SetFillStyle (тип штриховки, цвет)

Пример SetFillStyle (3, 4)

Параметр «тип штриховки» может принимать следующие значения:

0 – нет заливки

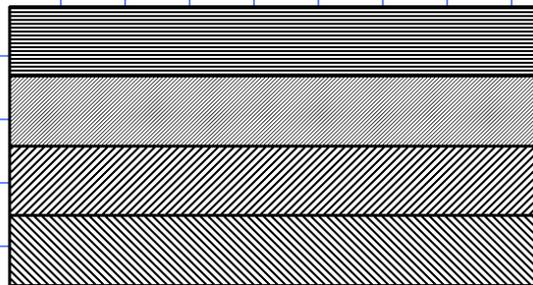
1 – сплошная

2 –

3 –

4 –

5 –



6 –

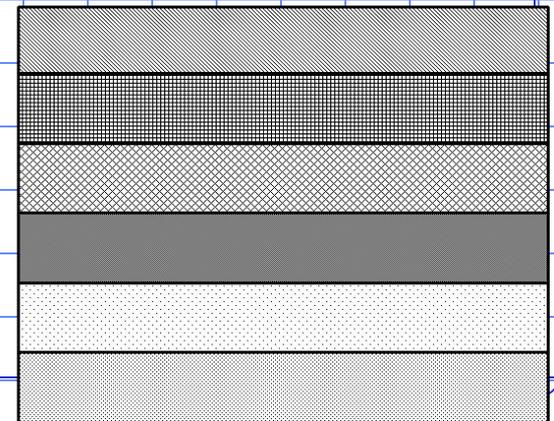
7 –

8 –

9 –

10 –

11 –

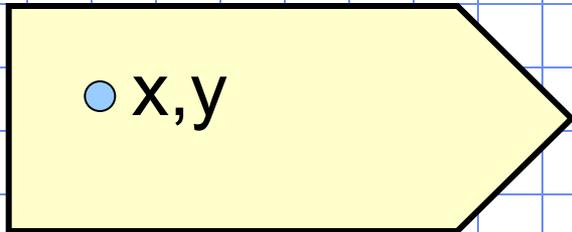


Заливка произвольной области

FloodFill (x, y, Border)

x, y - координаты точки внутри заливаемой замкнутой области

Border – цвет границы заливаемой области



Решение задач

Нарисовать цилиндр

```
Program cilindr;  
uses Graph, crt;  
Var gd, gm: integer;  
BEGIN
```

```
gd := detect;
```

```
InitGraph (gd, gm, "");
```

```
    Ellipse (420, 200, 0, 360, 100, 40); {верхнее основание}
```

```
    Line (320, 200, 320, 440);
```

```
    Line (520, 200, 520, 440);
```

```
    Ellipse (420, 440, 180, 360, 100, 40); {нижнее основание }
```

```
readKey;
```

```
closegraph;
```

```
END.
```

