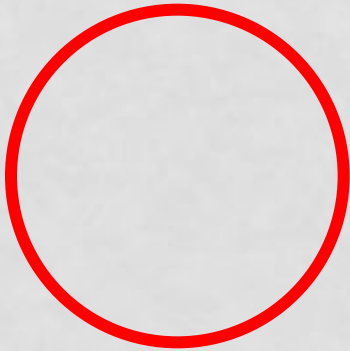


# **«Программирование движения на экране компьютера».**

При изображении движения на  
экране физически ничего не  
движется, а изменяется цвет точек  
экрана в определенном  
направлении



# АЛГОРИТМ ДВИЖЕНИЯ

- Определить координаты объекта
- Определить цвет объекта
- Нарисовать объект
- Задержать его на экране
- Закрасить объект цветом фона
- Повторить с п.1.

Координаты объекта при движении удобно определять в цикле

# ОПРЕДЕЛЕНИЕ НАПРАВЛЕНИЯ ДВИЖЕНИЯ

Направление	X	у
	Увеличивается	Const
	Уменьшается	Const
	Const	Увеличивается
	Const	Уменьшается
	Увеличивается	Увеличивается
	Увеличивается	Уменьшается
	Уменьшается	Увеличивается
	Уменьшается	Уменьшается

# ДВИЖЕНИЕ ОКРУЖНОСТИ ПО ГОРИЗОНТАЛИ

- Так как объект движется по горизонтали, то будет изменяться только координата X
- Для организации задержки изображения используется процедура **Delay(число)**
- Для того, чтобы использовать эту процедуру, нужно подключить модуль **CRT**
- Значение числа в **Delay(число)** зависит от скорости процессора. Чем больше число – тем медленнее движется объект.

```
program Circle1;
uses crt,graph;
var x,vga,vgahi:integer;
begin
initgraph(vga,vgahi,'c:\prog\bp\bgi');
for x:=20 to 620 do
begin
SetColor(4);
Circle(x,240,50);
delay(1000);
SetColor(0);
Circle(x,240,50);
end;
closegraph;
end.c
```

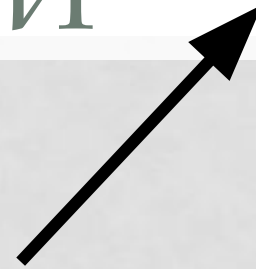
Подключаем модули CRT и Gfaph

**Важно!**  
Координаты окружности при рисовании и стирании должны быть одинаковыми, иначе получим «трубу», а не движущуюся окружность

Закрашиваем объект цветом фона (стираем)

# ДВИЖЕНИЕ ПО ДИАГОНАЛИ

Направление движения:



Координата **X** увеличивается, **Y** уменьшается.

Но они должны изменяться в одном цикле.

Если  $x$  – счетчик цикла, то команда рисования окружности будет такая:

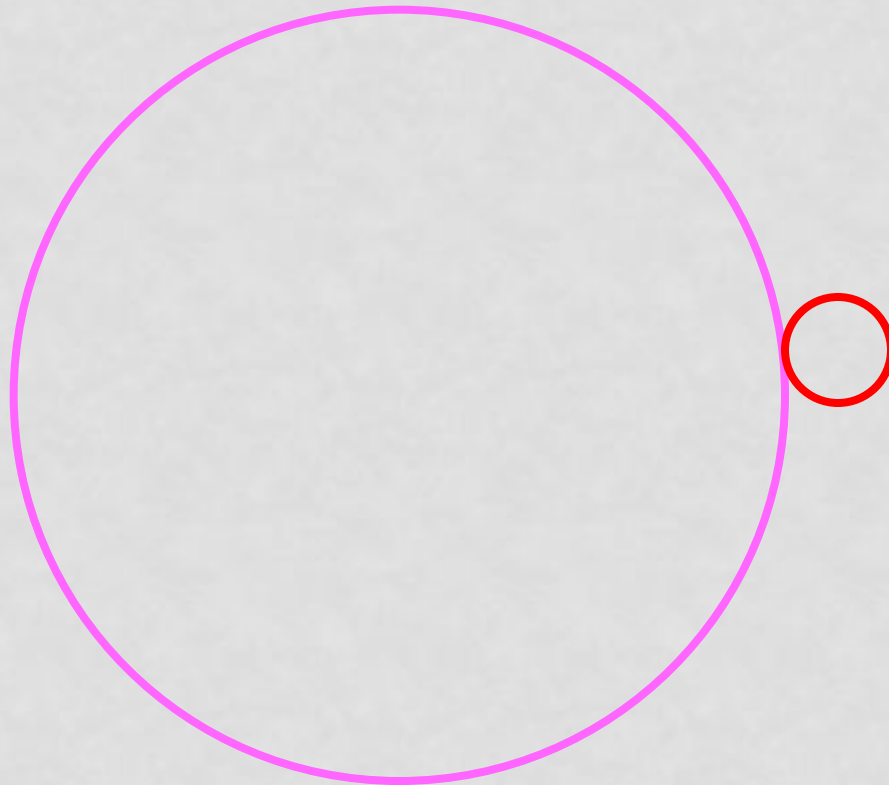
**Circle ( $x$ ,  $480-x$ , радиус)**

Как изменить координаты при движении в других направлениях?

```
program Circle1_1;
uses crt,graph;
var x,vga,vgahi:integer;
begin
initgraph(vga,vgahi,'c:\prog\bp\bgi');
for x:=20 to 620 do
  begin
    SetColor(4);
    Circle(x,480-x,50);
    delay(1000);
    SetColor (0);
    Circle(x, 480-x,50);
  end;
closegraph;
end.
```



НАРИСОВАТЬ ДВИЖУЩИЙСЯ ПО  
БОЛЬШОЙ ОКРУЖНОСТИ  
МАЛЕНЬКИЙ КРУЖОЧЕК



# ПОЯСНЕНИЯ

Пусть окружность с радиусом **R1** размещена в обычной системе координат с центром в точке **(0,0)**

Движущаяся по окружности т. **A** определяет угол **φ**, образованный радиусом **OA** и осью **OX**.

Т.о. координаты т.А в любом месте на окружности имеют вид:

$$X=R1 \cos \phi$$

$$Y=R1 \sin \phi$$

Если центр окружности находится в центре экрана – в т. **(320,240)**, то

$$X=320+R1 \cos \phi$$

$$Y=240-R1 \sin \phi$$

Угол **φ** должен возрастать в цикле от **0<sup>0</sup>** до **2π**:

**φ=6,28(i/n)** где **n** – число точек на окружности, **i** – счетчик цикла (от **0** до **n**)

Получаем следующие команды для вычисления координат центра движущейся маленькой окружности:

```
x:=trunc(320+r1*cos(6.28*(i/200)));  
y:=trunc(240-r1*sin(6.28*(i/200)));
```

Функцию **trunc** необходимо использовать, т. к. координаты могут быть только целыми числами.

```
program Circle2;
uses crt,graph;
var x, y, r1, r2, i, n, vga,vgahi:integer;
begin
initgraph(vga,vgahi,'c:\prog\bp\bgi');
r1:=200; r2:=20; n:=200;
SetColor (5);
Circle(320,240,178);
for i:=0 to n do
begin
x:=trunc(320+r1*cos(6.28*(i/n)));
y:=trunc(240-r1*sin(6.28*(i/n)));
SetColor(4);
Circle (x,y,r2);
Delay (10000);
SetColor(0);
Circle (x,y,r2);
end; closegraph; end.
```

# ПОСТРОЕНИЕ ГРАФИКА ФУНКЦИИ $\sin x$

Точку начала координат переносим в центр экрана:

**X0:=320;**

**Y0:=240;**

Т.к. значения функции **sin x** находятся в диапазоне **[-1,1]**, необходимо использовать коэффициент отображения функции **k**. Чем больше **k**, тем крупнее график.

**y:=trunc ( k \* sin ( x / k )**

Сам график рисуем командой **PutPixel** в цикле.

```
program sin_x;
uses crt,graph;
var x, y, x0, y0, k, vga,vgahi:integer;
begin
initgraph(vga,vgahi,'c:\prog\bp\bgi');
line(20,240,620,240); {ось x}
line(320,20,320,400); {ось y}
X0:=320; Y0:=240; k:=50;
for X:=-320 to 320 do
begin
y:=trunc(k*sin(x/k));
PutPixel(x0+x,y0-y,4);
end;
OutTextXY (50,440,'Grafic y=sin x');
readln; closegraph; end.
```

# ЗАДАНИЕ НА УРОК:

Нарисовать:

- Шарик, скатывающийся по ступенькам
- Луч, движущийся по кругу
- График функции  $x^2$

