

# **ОСНОВЫ ЯЗЫКА Pascal**

**Графика: циклы,  
операторы Case, Random**

# Операторы циклы

---

**Цикл** — это многократное выполнение одинаковой последовательности действий.

- ЦИКЛ С **ИЗВЕСТНЫМ** ЧИСЛОМ ШАГОВ
- ЦИКЛ С **НЕИЗВЕСТНЫМ** ЧИСЛОМ ШАГОВ  
(цикл с условием)

# Оператор цикла FOR

---

## Увеличение переменной на 1:

```
for <переменная> := <начальное значение> to  
    <конечное значение> do begin  
    {тело цикла}  
end;
```

## Уменьшение переменной на 1:

```
for <переменная> := <начальное значение>  
    downto  
    <конечное значение> do begin  
    {тело цикла}  
end;
```

# Пример кода

---

```
for i:=1 to 5 do begin  
  a := i*i;  
  writeln(a:4);  
end;
```

```
1  
4  
9  
16  
25
```

```
for i:=5 downto 1 do begin  
  a := i*i;  
  writeln(a:4);  
end;
```

```
25  
16  
9  
4  
1
```

# Оператор цикла WHILE

---

```
while <условие> do begin
    {тело цикла}
end;
```

## Особенности:

- МОЖНО ИСПОЛЬЗОВАТЬ СЛОЖНЫЕ УСЛОВИЯ:

```
while (a<b) and (b<c) do begin
    {тело цикла}
end;
```

- если в теле цикла только один оператор, слова **begin** и **end** можно не писать:

```
while a < b do
    a := a + 1;
```

# Цикл с условием

---

## Особенности:

- условие пересчитывается **каждый раз** при входе в цикл
- если условие на входе в цикл ложно, цикл не выполняется ни разу

```
a := 4; b := 6;  
while a > b do  
    a := a - b;
```

- если условие никогда не станет ложным, программа **зацикливается**

```
a := 4; b := 6;  
while a < b do  
    d := a + b;
```

# Цикл с постусловием (цикл REPEAT)

---

**Цикл с постусловием** – это цикл, в котором проверка условия выполняется в конце цикла.

```
repeat
    writeln('Введите положительное число');
    read(n);
    until n > 0;
```

условие **ВЫХОДА**

## Особенности:

- тело цикла всегда выполняется хотя бы один раз
- после слова **until** ("до тех пор, пока не...") ставится условие **ВЫХОДА** из цикла

# Оператор выбора CASE

---

## Особенности:

- после **case** может быть имя переменной или арифметическое выражение целого типа (**integer**)

```
case i+3 of
  1: begin a := b; end;
  2: begin a := c; end;
end;
```

ИЛИ СИМВОЛЬНОГО ТИПА (**char**)

```
var c: char;
...
case c of
  'a': writeln('Антилопа');
  'б': writeln('Барсук');
  else writeln('Не знаю');
end;
```



# Оператор выбора

---

## Особенности:

- если нужно выполнить только один оператор, слова **begin** и **end** можно не писать

```
case i+3 of
  1: a := b;
  2: a := c;
end;
```

- нельзя ставить два одинаковых значения

```
case i+3 of
  1: a := b;
  1: a := c;
end;
```

# Оператор выбора

---

## Особенности:

- значения, при которых выполняются одинаковые действия, можно группировать

перечисление

диапазон

смесь

```
case i of
  1:           a := b;
  2, 4, 6:    a := c;
  10..15:     a := d;
  20, 21, 25..30: a := e;
  else writeln('Ошибка');
end;
```

# Датчик случайных чисел

---

`random` - генерирует случайное вещественное число в диапазоне  $[0, 1)$ .

`random(x)` - генерирует случайное целое число в диапазоне  $[0, x)$ .

`x + random * (y-x)` - генерирует случайное вещественное число в диапазоне  $[x, y)$ .

`x + random(y-x)` - генерирует случайное целое число в диапазоне  $[x, y)$ .

Для повышения «степени случайности» существует процедура **randomize**, которая меняет базу генерации, ее используют до функции `random`.

```
randomize ;  
random(n) ;
```

Выдаст число в диапазоне  $0 \dots n-1$

```
random(5) ;
```

Выдаст число в диапазоне  $0 \dots 4$

# Случайный цвет

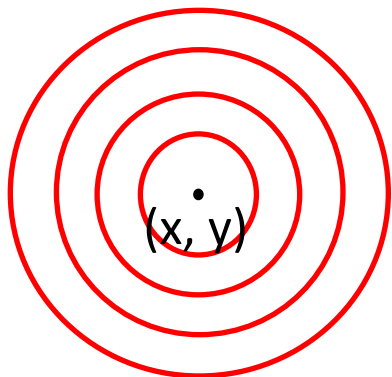
Цвет в `Wingraph` записывается с помощью констант и имеет тип `longint`

Константа	Значение	Константа	Значение
Black	0 (черный)	DarkGray	8 (темно-серый)
Blue	1 (синий)	LightBlue	9 (светло-синий)
Green	2 (зеленый)	LightGreen	10 (светло-зеленый)
Cyan	3 (голубой)	LightCyan	11 (светло-голубой)
Red	4 (красный)	LightRed	12 (розовый)
Magenta	5 (малиновый)	LightMagenta	13 (светло-малиновый)
Brown	6 (коричневый)	Yellow	14 (желтый)
LightGray	7 (светло-серый)	White	15 (белый)

```
Randomize;  
a:=random(10);  
case a of  
  0: color:=Blue;  
  ...  
  9: color:=Yellow;  
end;
```

# Концентрические окружности

---



```
Setcolor(red);  
for i:=1 to 4 do  
  circle(x,y,30 + (i-1)*20);
```

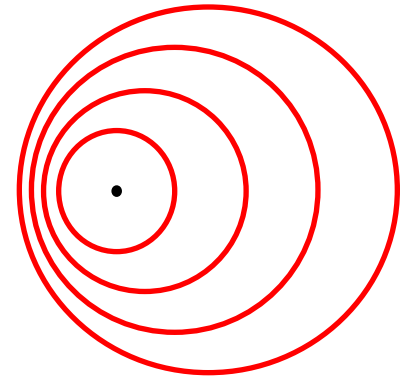
Радиус внутренней  
окружности

Расстояние между  
кольцами

# Задание

---

1. Нарисовать разноцветные окружности, пользователь задает их количество.
2. Нарисовать разноцветные окружности со смещенным центром.
3. Нарисовать цветные кольца.
4. Нарисовать радугу со случайными цветами.
5. Нарисовать радугу с правильными цветами на фоне неба.
6. Нарисовать правильный n-угольник.
7. Нарисовать циферблат.



# Задание

---

$(0,0)$

X

Y

