

# Операторы повтора (ЦИКЛЫ)

---

Цикл представляет собой последовательность операторов, которая выполняется неоднократно.

---

Три разновидности цикла:

цикл с постусловием (инструкция repeat),

цикл с предусловием (инструкция while)

цикл со счетчиком (инструкция for).

# Оператор REPEAT

---

Оператор повтора repeat состоит из заголовка (repeat), тела и условия окончания (until).  
Ключевые слова repeat, until обозначают "повторяй" и "пока" соответственно.

**repeat**

{ Инструкции }

**until** Условие выхода из цикла;



# Игра "Угадай число"

игрок должен угадать число, назовем его `comp`, "задуманное" компьютером — случайное число в диапазоне от 0 до 1000. Процесс продолжается до тех пор, пока значение переменной `igrok`, которая вводится с клавиатуры, не совпадет со значением переменной `com`

---

**var**

`comp: integer; { число, "задуманное" компьютером }`

`igrok: integer; { вариант игрока }`

**begin**

`randomize; { инициализация датчика случайных чисел } comp:=random(1000); { компьютер загадал число }`

**repeat** `write('Введите число: ');`

`readln(igrok);`

**if** `igrok>comp` **then** `writeln('Слишком много...');`

**if** `igrok<comp` **then** `writeln('Слишком мало...');`

**until** `comp=igrok;`

`writeln('Вы выиграли!')`

**end.**

# Оператор WHILE

---

Оператор повтора `while` состоит из заголовка и тела цикла. Ключевые слова `while` и `do` обозначают "до тех пор, пока" и "выполни" соответственно.

```
while Условие выполнения цикла do  
begin  
{ Инструкции }  
end;
```



# СТРОКИ:

1 2 3 4 5 6 7 8 9 10

10 9 8 7 6 5 4 3 2 1

---

```
var i: integer;  
begin  
  i:=0;  
  while i<10 do  
    begin  
      i:=succ(i);  
      write (i, ' ');  
    end;  
  Writeln;  
  i:=10;  
  while i>10 do  
    begin  
      write (i, ' ');  
      i:=pred(i);  
    end;  
end.
```

# Оператор FOR

---

Этот вид оператора цикла называют циклом со счетчиком или циклом с параметром. В нем важную роль играет переменная-параметр, которая на каждом шаге цикла автоматически изменяет свое значение ровно на единицу — поэтому ее и называют счетчиком.

Инструкцию `for` можно реализовать двумя способами.

# Вариант 1 (с увеличением счетчика).

---

```
for Счетчик := НачальноеЗначение to КонечноеЗначение do
```

```
begin
```

```
    { Инструкции }
```

```
end;
```

Ключевые слова `for`, `do` обозначают "для", "выполни" соответственно. Строка содержащая `for...do`, называется заголовком цикла, оператор, стоящий после `do` образует его тело. Очень часто тело цикла — составной оператор. Если тело цикла представлено одиночным оператором, то `begin` и `end` не пишутся.



# Вариант 2 (с уменьшением счетчика).

---

```
for Счетчик := НачальноеЗначение downto КонечноеЗначение do  
begin  
{ Инструкции }  
end;
```

Инструкции между `begin` и `end` выполняются столько раз, сколько определяет выражение  $[(\text{НачальноеЗначение} - \text{КонечноеЗначение}) + 1]$ . Если `НачальноеЗначение` меньше, чем `КонечноеЗначение`, то инструкции между `begin` и `end` не выполняются ни разу.

# Вывод на экран таблицы умножения

---

```
var
i,j: integer; { номер строки и столбца таблицы }
begin
for i:=1 to 10 do
    begin
    for j:=1 to 10 do
        write(i*j:5);
        writeln; { перевод строки }
    end;
end.
```

# Написать программу для решения задачи

---

1. Составить программу, которая запрашивает пароль (например, четырехзначное число) до тех пор, пока не будет правильно введен.
2. В подъезде  $N$  ступенек. Сколько шагов будет сделано, если шагать через 3 ступеньки