

# Тема уроку

## КОМАНДИ ПОВТОРЕННЯ (Turbo Pascal 7.0)



# Поняття циклу

**Циклом** називається процес періодичного повторення певної дії або групи дій.

Конструкція **повторення** – **цикл** – дозволяє певну кількість разів виконати заданий оператор або групу операторів.

У Паскалі передбачено три різновиди операторів циклу:

- **з параметром** – арифметичний цикл
- **з передумовою** – ітераційний цикл
- **з післяумовою** – ітераційний цикл

# 1. Цикл з параметром

Коли кількість повторень відома – використовують команду повторення з параметром (**for...**).

**FOR** лічильник := поч.знач. **TO** кінц.знач. **DO**

**BEGIN**

серія команд, що повторюються;

**END;**

- Лічильник автоматично збільшується на 1 при кожному виконанні серії команд.
- Лічильник тільки цілого типу (**integer**).
- Лічильник можна використовувати для розрахунків всередині циклу, але **не можна** змінювати його значення ніякими командами.

# 1. Цикл з параметром

Наприклад: *вивести на екран 7 зірочок.*

```
Program P01;  
Var i : integer;  
Begin  
  
    For i:=1 to 7 do WriteLn('*');  
  
end.|
```



Результат  
виконання  
програми

# Складові циклу з параметром

До складу конструкції циклу з параметром входять:

- 1) **For** – службове слово, з якого починається запис конструкції і яке в перекладі означає «Для».
- 2) **i:=1** – вказівка присвоювання **параметру циклу** (в нашому прикладі – змінній **i**, обов'язково цілого типу) початкового значення (в нашому прикладі **1**. Може бути змінна або вираз).
- 3) **to 7** – службове слово **to** (“до”) та число **7** вказують на кінцеве значення параметру циклу. (Може бути змінна або вираз).
- 4) **do** – службове слово, що означає “виконати”.
- 5) Вказівка **WriteLn(\*)** є тією вказівкою, яка буде виконуватися. Вона є **тілом циклу**.



В цьому прикладі значення параметру циклу (змінної **i**) змінюватиметься **від 1 до 7 із кроком 1** і при цьому тіло циклу виконуватиметься 7 разів: при  $i=\{1,2,3,4,5,6,7\}$ .

## Як вивести на екран 100 зірочок?

Вказівка **For i:=1 to 7 do WriteLn('\*');**  
виводить на екран 7 зірочок.

Щоб вивести на екран 100 зірочок досить у  
вказівці число 7 замінити на 100:

**For i:=1 to 100 do WriteLn('\*');**

# Використання параметру циклу в тілі циклу

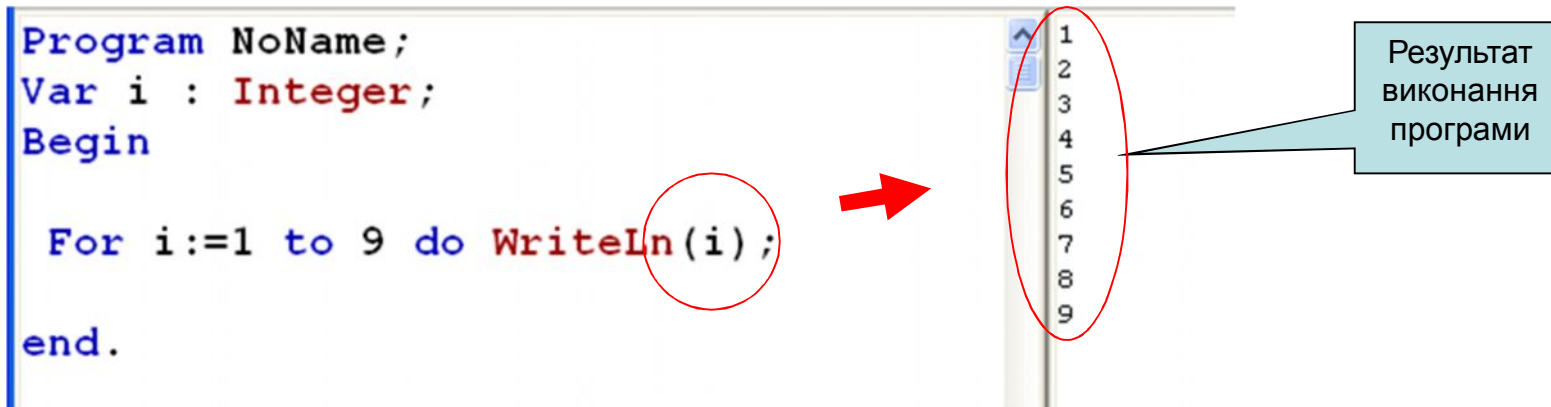
**Параметр циклу** – це змінна величина цілого типу **Integer** (допустимі також типи **Char** та **Boolean**), значення якої автоматично змінює конструкція циклу після кожного виконання тіла циклу.

Ці значення можна використовувати у вказівках тіла циклу.

Розглянемо приклад.

*Вивести числа від 1 до 9.*

```
Program NoName;  
Var i : Integer;  
Begin  
  
  For i:=1 to 9 do WriteLn(i);  
  
end.
```



Результат виконання програми

# Як бути, коли в тіло циклу треба включити декілька вказівок

За правилами **тілом циклу може бути лише одна вказівка** – та, яка записана відразу за словом **do**.

Якщо є необхідність (а це буває часто) в тіло циклу включити декілька вказівок, то їх необхідно помістити між операторними дужками **begin** та **end**, тоді вони сприйматимуться, як одна вказівка, наприклад:

```
Program NoName;  
Var i,k : Integer;  
Begin  
  For i:=1 to 3 do  
    begin  
      k:=2*i;  
      WriteLn(k);  
    end;  
end.
```



2  
4  
6

Результат виконання програми



# 1. Цикл з параметром

Якщо значення лічильника повинно змінюватись від **більшого** числа до **меншого**, то використовують такий варіант команди повторення з параметром (**for...**).

**FOR** лічильник := поч.знач. **DOWNTO** кінц.знач. **DO**  
**BEGIN**  
    серія команд, що повторюються;  
**END;**

# Цикл із зменшенням змінної

---

**Задача.** Вивести на екран квадрати і куби цілих чисел від 8 до 1 (в зворотному порядку).

**Особливості:** змінна циклу повинна зменшуватися.

**Розв'язання:**

```
...  
for i:=8 downto 1 do begin  
    i2 := i*i;  
    i3 := i2*i;  
    writeln(i:4, i2:4, i3:4);  
end;  
...
```

# Скільки разів виконається цикл?

---

```
a := 1;  
for i := 1 to 3 do a := a + 1;
```

a = 4

```
a := 1;  
for i := 3 to 1 do a := a + 1;
```

a = 1

```
a := 1;  
for i := 1 downto 3 do a := a + 1;
```

a = 1

```
a := 1;  
for i := 3 downto 1 do a := a + 1;
```

a = 4

## 2. Цикл з передумовою

Коли виконання команд циклу залежить від того **виконується якась умова чи ні**, то використовують повторення з передумовою (**While...**).

**WHILE** умова **DO**

**BEGIN**

серія команд, що повторюються;

**END;**

Якщо умова є істинною , то серія команд виконується і виконавець повертається назад до перевірки умови.

При невиконанні умови – не виконуються команди.

## 2. Цикл з передумовою

*Написати програму, яка дозволяє надати цілочисловій змінній X тільки додатного значення.*

```
Program NoName;  
Var X : Integer;  
Begin  
    ReadLn (X) ;  
    While X<=0 do ReadLn (X) ;  
end.|
```

-1  
0  
1

Результат  
виконання  
програми

# Складові циклу з передумовою

До складу конструкції циклу з передумовою входять:

- 1) **While** – службове слово, з якого починається запис конструкції і яке в перекладі означає «Поки».
- 2) **X<=0** – умова виконання тіла циклу. Коли вона має значення **True**, управління передається тілу циклу, коли **False** – управління передається на наступну за циклом вказівку.
- 3) Вказівка **ReadLn(X)** є тією вказівкою, яка буде виконуватися. Вона є **тілом циклу**.

умова виконання  
циклу

```
While X<=0 do ReadLn(X);
```

тіло циклу

В цьому прикладі вказівка введення **ReadLn(X)**, виконуватиметься до тих пір, поки вираз **X<=0** не отримає значення **False**.

Оскільки умова передуює тілу циклу, то значення змінної **X** повинно існувати до передачі управління циклу.

# Як бути, коли в тіло циклу треба включити декілька вказівок

- За правилами **тілом циклу може бути лише одна вказівка** – та, яка записана відразу за словом **do**.

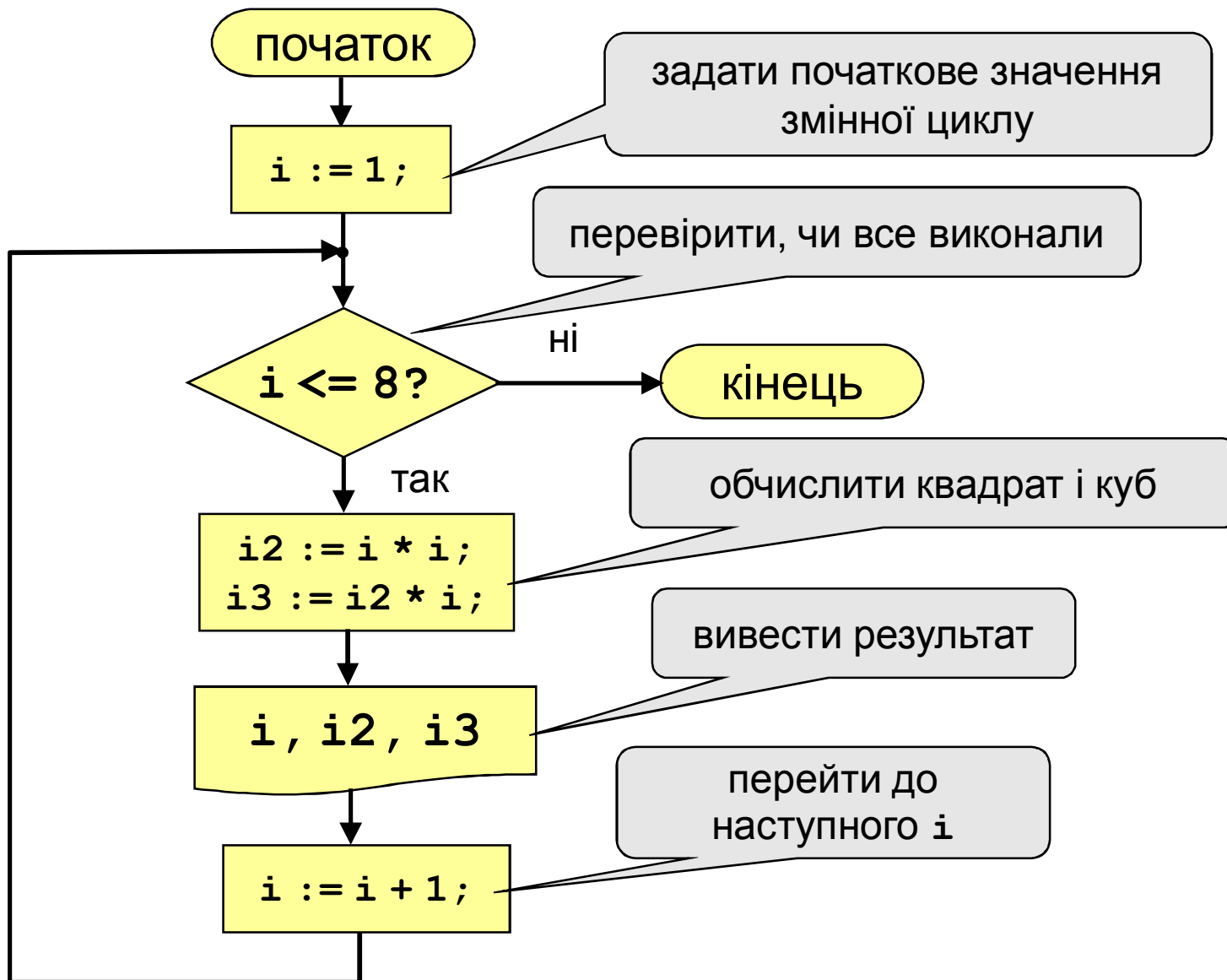
Якщо є необхідність (а це буває часто) в тіло циклу включити декілька вказівок, то їх необхідно помістити між операторними дужками **begin** та **end**, тоді вони сприйматимуться, як одна вказівка, наприклад:

```
Program NoName;  
Var X : Integer;  
Begin  
  WriteLn('Введіть додатне значення: ');  
  ReadLn(X);  
  While X<=0 do  
    begin  
      WriteLn('Введіть додатне значення: ');  
      ReadLn(X);  
    end  
end.
```

```
Введіть додатне значення:  
-1  
Введіть додатне значення:  
0  
Введіть додатне значення:  
1
```

Результат виконання програми

**Задача.** Вивести на екран квадрати і куби цілих чисел від 1 до 8 (від **a** до **b**).





## 3. Цикл з післяумовою

Коли виконання команд циклу потрібно робити до тих пір, поки не виконається поставлена умова, то використовують повторення з післяумовою (**Repeat...**).

### **REPEAT**

серія команд, що повторюються;

**UNTIL** умова;

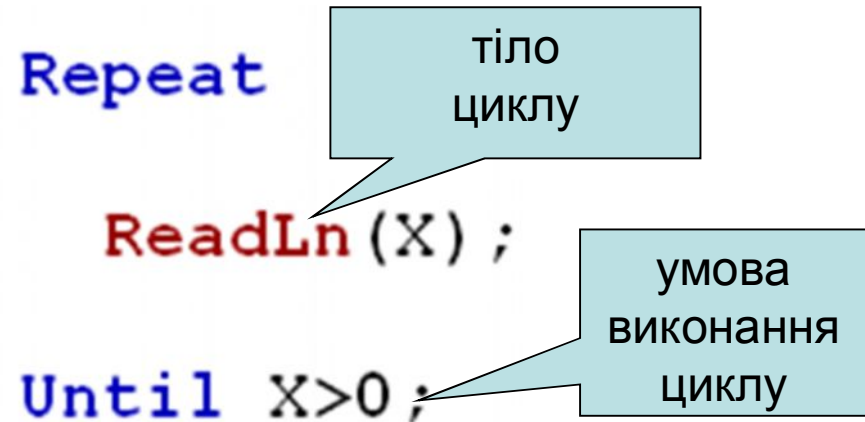
Якщо умова є хибною, то після виконання серії команд виконавець повертається назад для повторного їх виконання.

При виконанні умови – цикл закінчується.

# Складові циклу з післяумовою

До складу конструкції циклу з передумовою входять:

- 1) **Repeat** – службове слово, з якого починається запис конструкції і яке в перекладі означає «Повторити».
- 2) **Until** - службове слово, після якого записується умова і яке в перекладі означає «До».
- 3) **X>0** – умова виконання тіла циклу. Коли вона має значення **False**, управління передається тілу циклу, коли **True** – управління передається на наступну за циклом вказівку.
- 3) Вказівка **ReadLn(X)** є тією вказівкою, яка буде виконуватися. Вона є **тілом циклу**.



В цьому прикладі вказівка введення **ReadLn(X)**, виконуватиметься до отримання виразом **X>0** значення **True**.

Оскільки умова записана після тіла циклу, то значення змінної **X** може не існувати до передачі управління циклу.

# Програма

---

```
program qq;  
var n: integer;  
begin
```

```
  repeat  
    writeln('Ввести додатне число');  
    read(n);
```

```
  until n > 0;
```

умова ВИХОДУ

```
  ... { основний алгоритм }
```

```
end.
```

## Особливості:

- тіло циклу завжди виконується хоча б один раз
- після слова **until** ("до тих пір, поки не...") ставиться умова ВИХОДУ із циклу

## Вказати значення змінної **s** після виконання вказівки повторення:

1)  $s:=1; i:=2; \text{while } i<5 \text{ do begin } i:=i+1; s:=s+i; \text{end}$   
 $s=$  \_\_\_\_\_

2)  $s:=5; i:=0; \text{while } i<4 \text{ do } i:=i+1; s:=s*i$   
 $s=$  \_\_\_\_\_

3)  $s:=-2; i:=1; \text{repeat } s:=s+i; i:=i+2 \text{ until } (i>3);$   
 $s=$  \_\_\_\_\_

4)  $s:=10; i:=5; \text{repeat } s:=s+i; i:=i-1 \text{ until } (i\leq 3);$   
 $s=$  \_\_\_\_\_

5)  $s:=2; \text{for } i:=5 \text{ to } 7 \text{ do } s:=s+i;$   
 $s=$  \_\_\_\_\_

6)  $s:=1; \text{for } i:=5 \text{ downto } 2 \text{ do } s:=s-2*i;$   
 $s=$  \_\_\_\_\_ -

# Завдання

---

«1»: Ввести  $a$  і  $b$  і вивести квадрати і куби чисел від  $a$  до  $b$ .

Приклад:

Введіть границі інтервалу:

4 6

4 16 64

5 25 125

6 36 216

«2»: Вивести квадрати і куби 10 чисел наступної послідовності: 1, 2, 4, 7, 11, 16, ...

Приклад:

1 1 1

2 4 8

4 16 64

...

46 2116 97336

## Домашнє завдання:

- Скласти конспект
- Підготуватися до тестової перевірки
- Виконати завдання №1-2

**Дякую за роботу на уроці!  
До нових зустрічей!!!**