

# Програмування на мові Паскаль

## Тема 3. Цикли з умовою

# Цикл з невідомою кількістю кроків

---

**Приклад:** Відрізати поліно від колоди. Скільки разів потрібно зробити рух пилкою?

**Задача:** Ввести ціле число ( $<2000000$ ) і визначити кількість цифр у ньому.

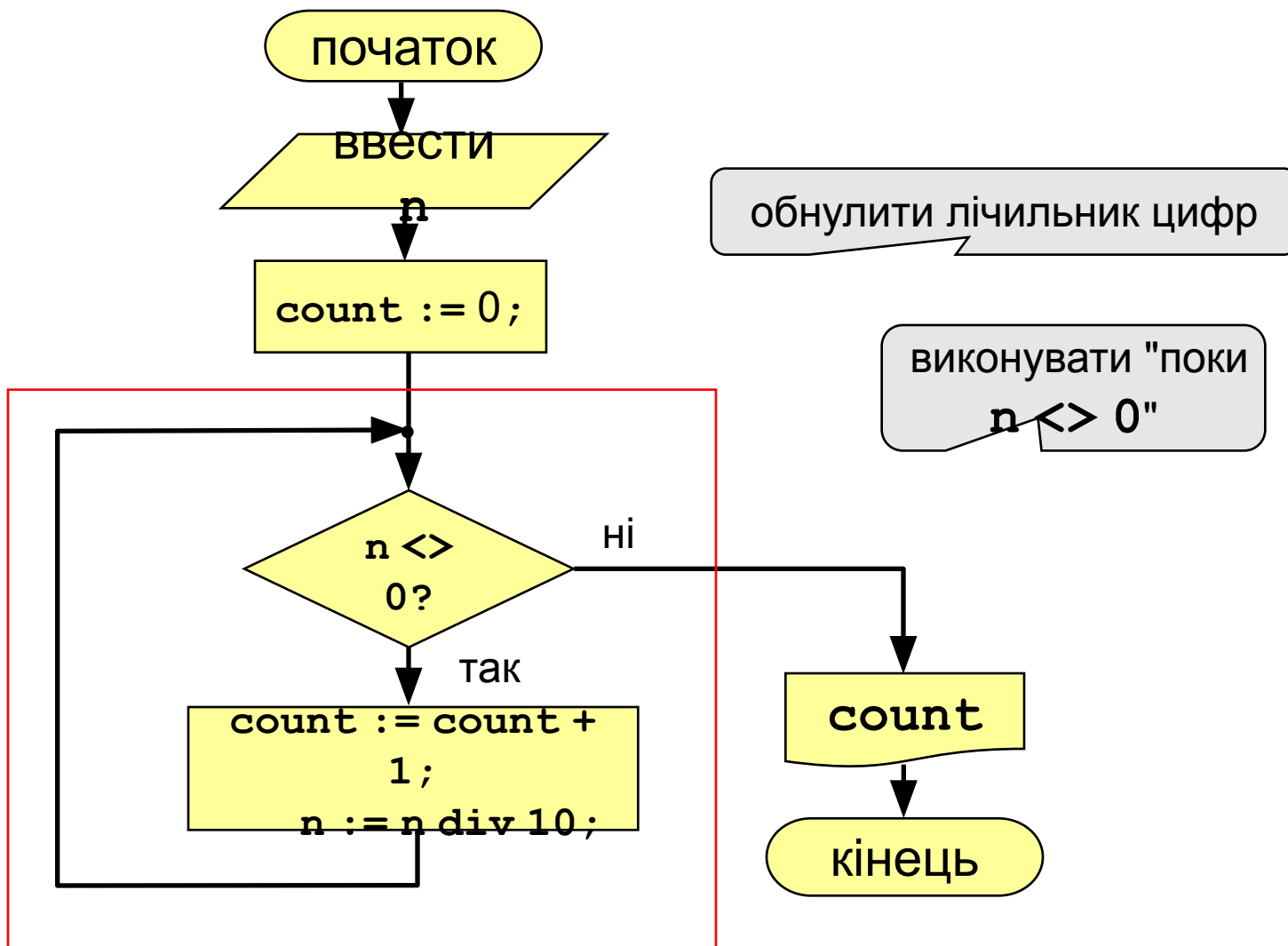
**Ідея розв'язання:** Відрізаємо послідовно останню цифру, збільшуємо лічильник.

n	count
123	0
12	1
1	2
0	3

**Проблема:** Невідомо, скільки кроків потрібно зробити.

**Розв'язання:** Потрібно зупинитися, коли  $n = 0$ , тобто потрібно робити "поки  $n \neq 0$ ".

# Алгоритм



# Програма

---

```
program qq;  
var n, count, n1: integer;  
begin  
  writeln('Ввести ціле число');  
  read(n); n1 := n;  
  count := 0;  
  while n <> 0 do begin  
    count := count + 1;  
    n := n div 10;  
  end;  
  writeln('В числі ', n1, ' знайшли ',  
        count, ' цифр');  
end.
```

ВИКОНУВАТИ "ПОКИ

$n \neq 0$ "



Що погано?

# Цикл с умовою

---

```
while <умова> do begin
    {тіло циклу}
end;
```

## Особливості:

- можна використовувати складені умови:

```
while (a<b) and (b<c) do begin
    {тіло циклу}
end;
```

- якщо в тілі циклу тільки один оператор, слова **begin** і **end** можна не писати:

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

# Цикл з умовою

---

## Особливості:

- умова перевіряється **кожен раз** при вході в цикл
- якщо умова на вході в цикл хибна, цикл не виконується жодного разу

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

- Якщо умова ніколи не стане хибною, програма **зациклиться**

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

# Скільки разів виконується цикл?

```
a := 4; b := 6;  
while a < b do a := a + 1;
```

2 рази

a = 6

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

1 раз

a = 10

```
a := 4; b := 6;  
while a > b do a := a + 1;
```

0 разів

a = 4

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

1 раз

b = -2

```
a := 4; b := 6;  
while a < b do a := a - 1;
```

зациклювання

# Заміна for на while і навпаки

---

```
for i:=1 to 10 do begin
  {тіло циклу}
end;
```

```
i := 1;
while i <= 10 do begin
  {тіло циклу}
  i := i + 1;
end;
```

```
for i:=a downto b do
begin
  {тіло циклу}
end;
```

```
i := a;
while i >= b do begin
  {тіло циклу}
  i := i - 1;
end;
```

Заміна циклу **for** на **while** можлива **завжди**.

Заміна **while** на **for** можлива тільки тоді, коли можна наперед **розрахувати кількість кроків циклу**.



# Завдання

---

**"4"**: Ввести ціле число і знайти суму його цифр.

**Приклад:**

Ввести ціле число:

**1234**

Сума цифр числа 1234 рівна 10.

**"5"**: Ввести ціле число і визначити, чи правда, що в його записі є дві однакові цифри.

**Приклад:**

Ввести ціле число:

**1234**

Ні.

Ввести ціле число:

**1224**

Так.

# Послідовності

## Приклади:

• 1, 2, 3, 4, 5, ...

$$a_n = n$$

$$a_1 = 1, a_{n+1} = a_n + 1$$

• 1, 2, 4, 7, 11, 16, ...

$$a_1 = 1, a_{n+1} = a_n + n$$

• 1, 2, 4, 8, 16, 32, ...

$$a_n = 2^{n-1}$$

$$a_1 = 1, a_{n+1} = 2a_n$$

•  $\frac{1}{2}, \frac{1}{2}, \frac{3}{8}, \frac{1}{4}, \frac{5}{32}, \dots$

$$\frac{1}{2}, \frac{2}{4}, \frac{3}{8}, \frac{4}{16}, \frac{5}{32}, \dots$$

$$a_n = \frac{b_n}{c_n}$$

$$b_1 = 1, b_{n+1} = b_n + 1$$

$$c_1 = 2, c_{n+1} = 2c_n$$

# Послідовності

**Задача:** знайти суму всіх елементів послідовності,

$$1, -\frac{1}{2}, \frac{2}{4}, -\frac{3}{8}, \frac{4}{16}, -\frac{5}{32}, \dots$$

які по модулю більші 0,001:

$$S = 1 - \frac{1}{2} + \frac{2}{4} - \frac{3}{8} + \frac{4}{16} - \frac{5}{32} + \dots$$

**Елемент послідовності (починаючи з №2):**

$$a = z \frac{b}{c}$$

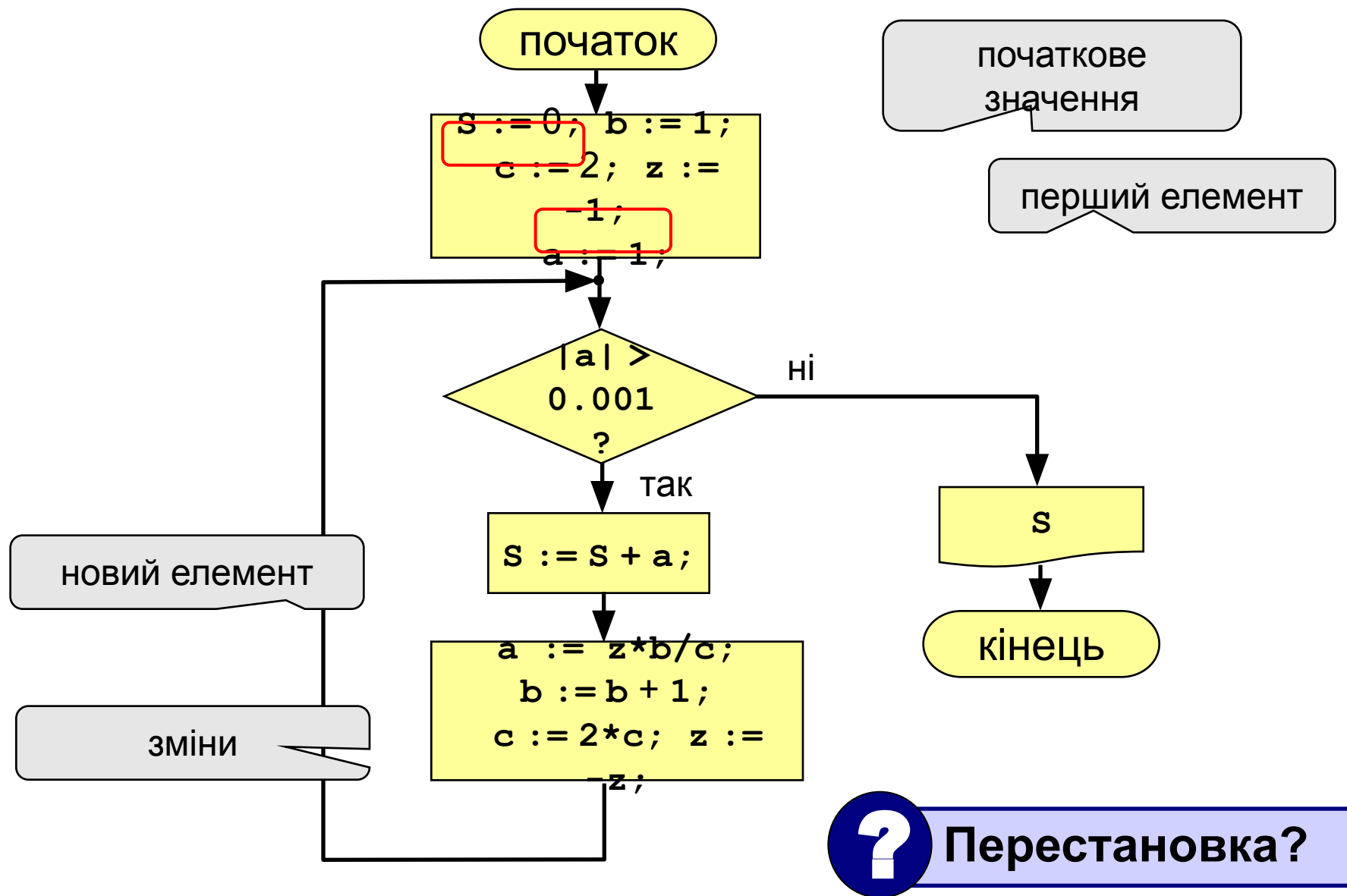
n	1	2	3	4	5	...
b	1	2	3	4	5	...
c	2	4	8	16	32	...
z	-1	1	-1	1	-1	...

`b := b+1;`

`c := 2*c;`

`z := -z;`

# Алгоритм



# Програма

---

```
program qq;  
var b, c, z: integer;  
    S, a: real;  
begin  
    S := 0; z := -1;  
    b := 1; c := 2; a := 1;  
    while abs(a) > 0.001 do begin  
        S := S + a;  
        a := z * b / c;  
        z := - z;  
        b := b + 1;  
        c := c * 2;  
    end;  
    writeln('S =', S:10:3);  
end.
```

початкове  
значення

збільшення  
суми

розрахунок елемента  
послідовності

перехід до  
наступного  
доданка

# Завдання

---

**"4":** Знайти суму елементів послідовності з точністю 0,001:

$$S = 1 + \frac{2}{3 \cdot 3} - \frac{4}{5 \cdot 9} + \frac{6}{7 \cdot 27} - \frac{8}{9 \cdot 81} + \dots$$

**Відповідь:**

$$S = 1.157$$

**"5":** Знайти суму елементів послідовності з точністю 0,001:

$$S = 1 + \frac{2}{2 \cdot 3} - \frac{4}{3 \cdot 9} + \frac{6}{5 \cdot 27} - \frac{8}{8 \cdot 81} + \frac{10}{13 \cdot 243} - \dots$$

**Відповідь:**

$$S = 1.220$$

# Цикл з післяумовою

---

**Задача:** Ввести ціле **додатне** число ( $<2000000$ ) і визначити кількість цифр в ньому.

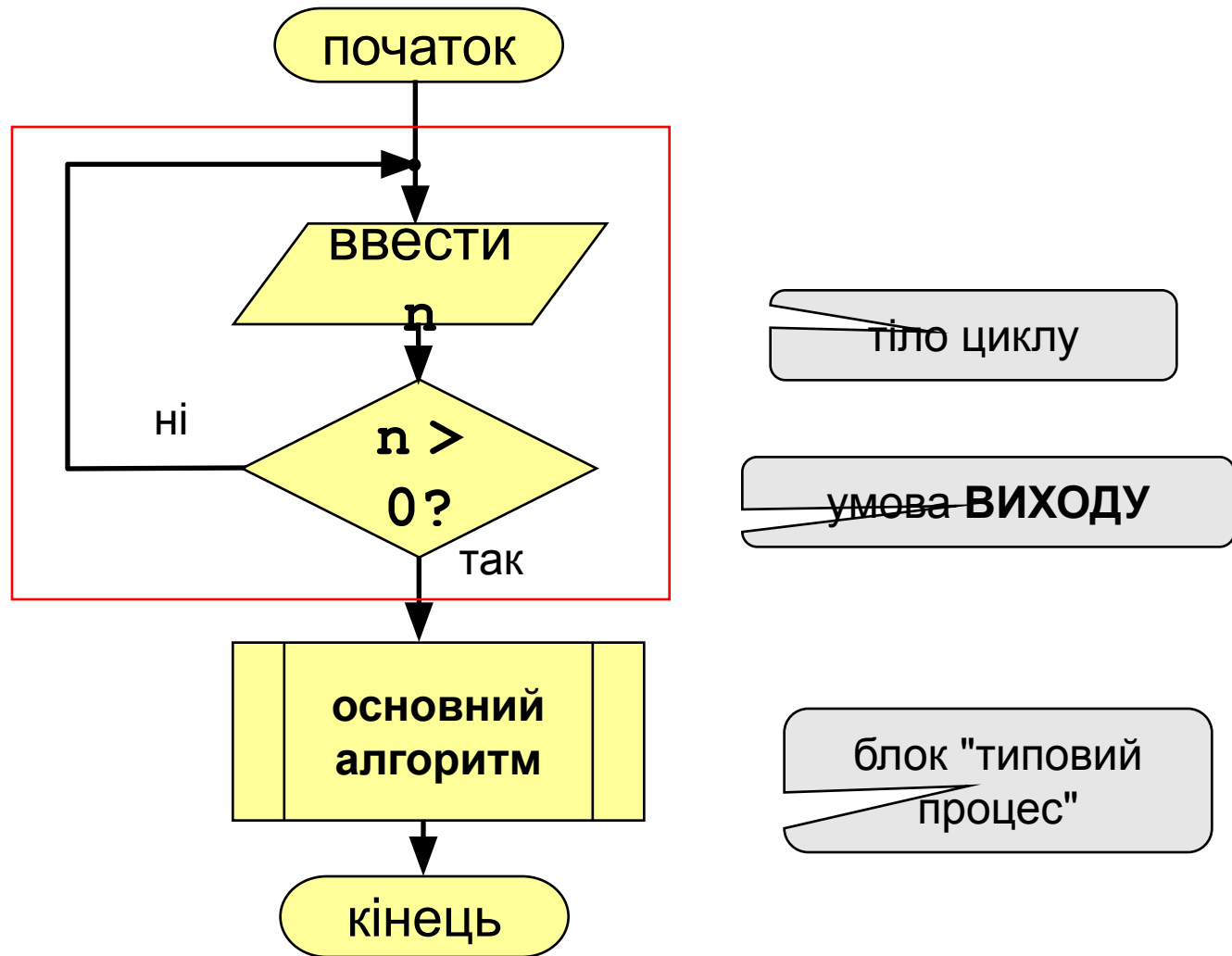
**Проблема:** Як не дати ввести від'ємне число або нуль?

**Розв'язання:** Якщо вводиться неправильне число, повернутися назад до введення даних (цикл!).

**Особливості:** Один раз тіло циклу потрібно виконати в будь-якому випадку => перевірку умови циклу потрібно виконувати в кінці циклу (цикл с **післяумовою**).

**Цикл с післяумовою** – це цикл, в якому перевірка умови виконується в кінці циклу.

# Цикл з післяумовою: алгоритм





# 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

# Скільки разів виконується цикл?

```
a := 4; b := 6;  
repeat a := a + 1; until a > b;
```

3 рази

a = 7

```
a := 4; b := 6;  
repeat a := a + b; until a > b;
```

1 раз

a = 10

```
a := 4; b := 6;  
repeat a := a + b; until a < b;
```

зациклення

```
a := 4; b := 6;  
repeat b := a - b; until a < b;
```

2 рази

b = 6

```
a := 4; b := 6;  
repeat a := a + 2; until a < b;
```

зациклення