

Й.Я. РИВКІНД, Т.І. ЛИСЕНКО,
Л.А. ЧЕРНІКОВА, В.В. ШАКОТЬКО

ІНФОРМАТИКА

8

Оператори циклу

За навчальною програмою 2017 року



Урок 59

teach-inf.com.ua



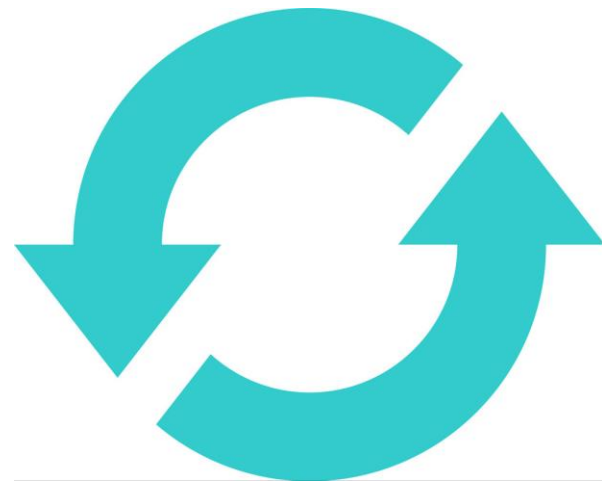
- 1. Які процеси називають циклічними? Наведіть приклади.**
- 2. Що таке цикл в алгоритмі? Що таке тіло циклу?**
- 3. Який вигляд має команда циклу з лічильником у мові програмування? Як вона виконується?**





Ви вже знаєте, що в багатьох алгоритмах одна або кілька команд повинні повторюватися багато разів поспіль. Недоцільно включати в алгоритм ці команди стільки разів, скільки вони повинні повторюватися, тому що кількість разів повторення може бути дуже велика або навіть заздалегідь невідома.

*У таких випадках потрібно в алгоритмах використовувати цикли. Нагадаємо, що **цикл в алгоритмі** – це фрагмент алгоритму, який може повторюватися поспіль більше ніж один раз.*

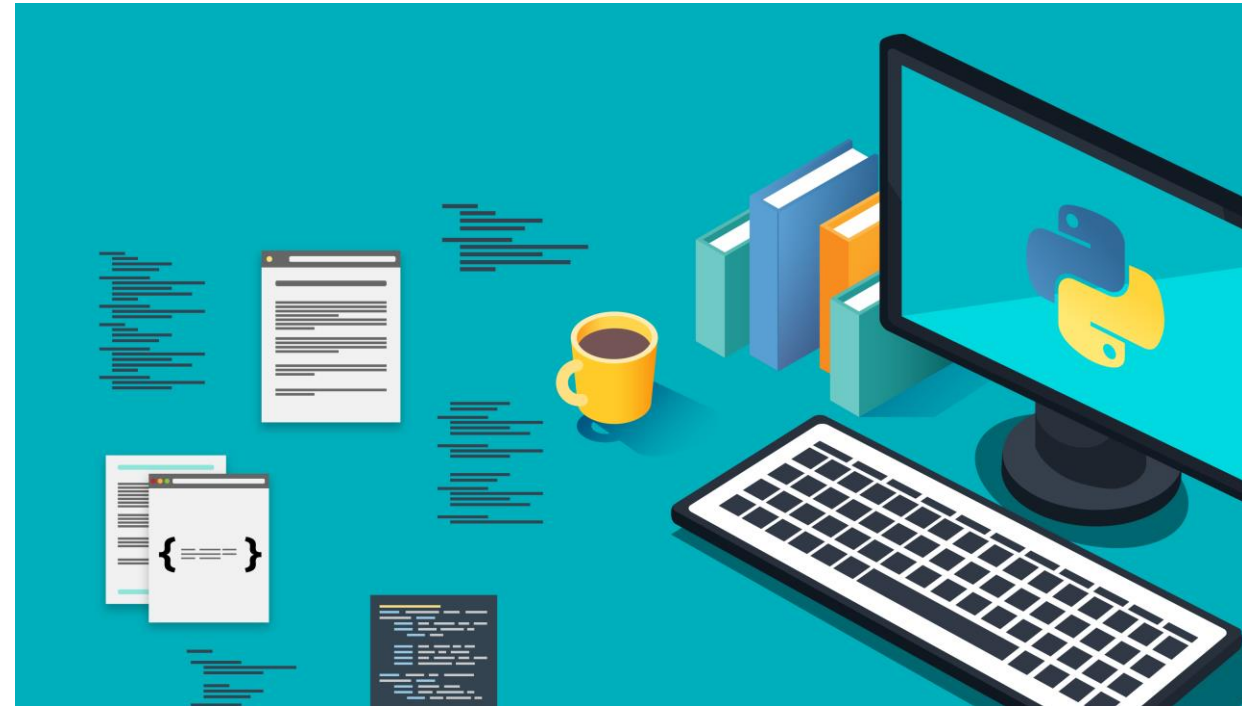




*У мовах програмування є кілька команд, які можуть реалізувати цикл. Одна з них – **команда циклу з лічильником**.*

Її доцільно використовувати в тих випадках:

коли кількість повторень команд тіла циклу відома ще до початку виконання циклу.





Загальний вигляд команди циклу з лічильником у мові *Python* такий:

**Змінна -
лічильником циклу**

**Рядок заголовка команди
циклу з лічильником**

```
for <ім'я змінної> in range (<кількість повторень>):  
<команди тіла циклу>
```

Функція, що утворює (генерує) набір послідовних цілих чисел, перше з яких 0, а останнє - на 1 менше від заданої кількості повторень (усього чисел стільки, скільки задано кількість повторень).



Наприклад, функція:

```
for i in range(3):
```

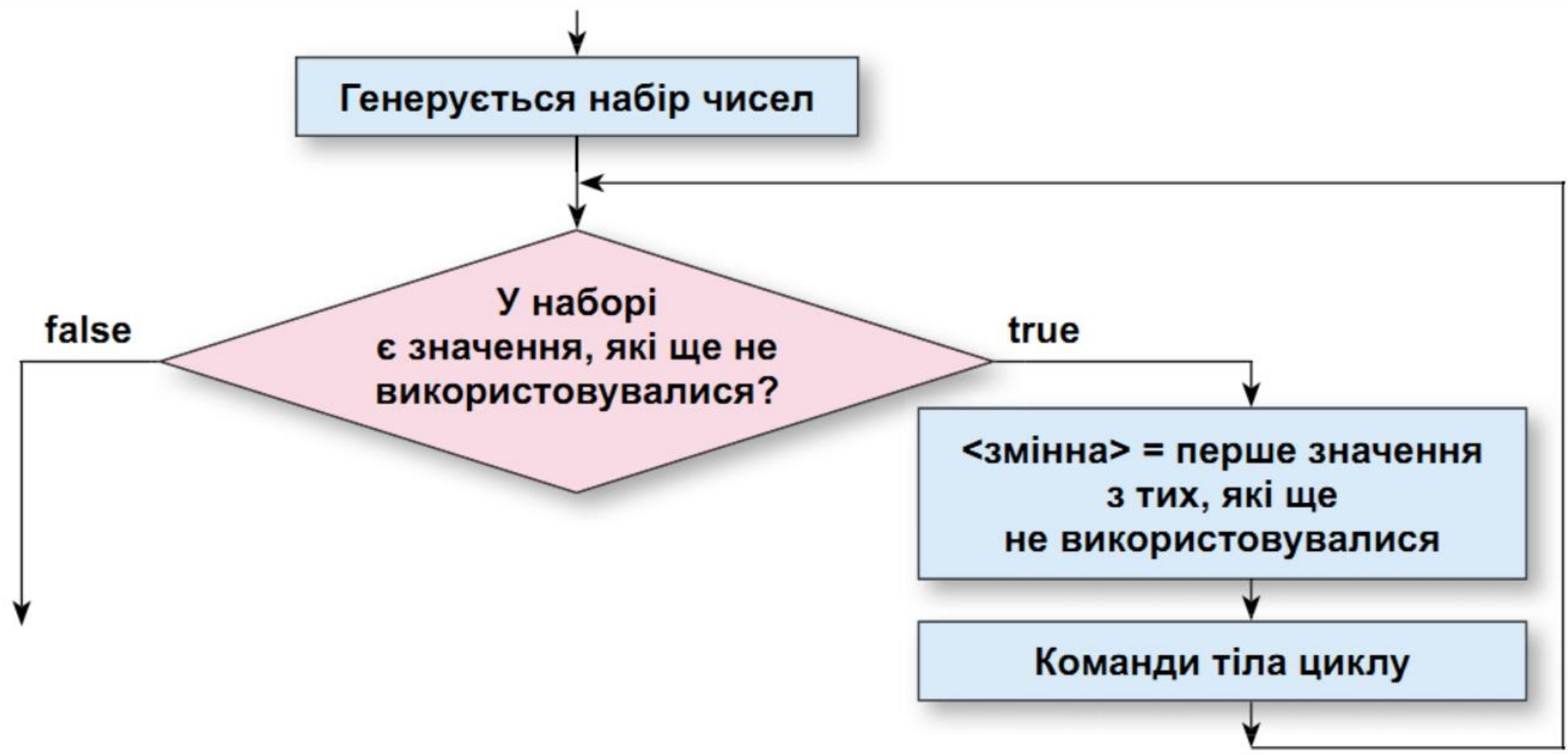
генерує набір чисел:

0, 1, 2, усього 3 числа.

*Змінна-лічильник **i** циклу послідовно набуває значень із цього набору чисел.*



Блок-схему команди циклу з лічильником у Python





Виконується команда циклу з лічильником так:

1. Функція *range* (<кількість повторень>) генерує набір чисел від 0 до (кількість повторень - 1).

2. Виконується перевірка, чи є в наборі числа, які ще не використовувалися і могли б стати новим значенням змінної-лічильника циклу.





Продовження...

3. Якщо результат перевірки **True**, то змінна-лічильник циклу набуває значення першого з набору чисел, які ще не використовувалися, і після цього виконується дія 4; якщо **False**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.



4. Виконуються команди тіла циклу.

5. Виконуються команди тіла циклу.



*У записі команди циклу з лічильником усі команди тіла циклу повинні бути записані з однаковим відступом від першої літери слова **for**.*

Для ілюстрації виконання команди циклу з лічильником розглянемо проєкт, у якому обчислюється сума $1^2 + 2^2 + 3^2 + 4^2$.

```
s=0  
for i in range(4):  
    s = s + (i + 1)*(i + 1)  
print('Сума дорівнює', s)
```



Виконаємо цей проєкт.

Команда

Результат виконання

`s = 0`

`s = 0`

`range(4)`

`0, 1, 2, 3`

У наборі ще є невикористані числа

`True`

*Змінна **i** набуває першого значення з набору невикористаних чисел*

`i = 0`



Продовження...

Команда

Результат виконання

```
s = s + (i + 1) * (i + 1)
```

```
s = 0 + (0 + 1) * (0 + 1) = 1
```

У наборі ще є невикористані числа

True

*Змінна **i** набуває першого значення з набору невикористаних чисел*

```
i = 1
```

```
s = s + (i + 1) * (i + 1)
```

```
s = 1 + (1 + 1) * (1 + 1) = 5
```



Продовження...

Команда

Результат виконання

У наборі ще є невикористані числа

True

*Змінна **i** набуває першого значення з набору невикористаних чисел*

$i = 2$

$s = s + (i + 1) * (i + 1)$

$s = 5 + (2 + 1) * (2 + 1) = 14$

У наборі ще є невикористані числа

True



Продовження...

Команда

Результат виконання

*Змінна **i** набуває першого значення з набору невикористаних чисел*

i = 3

s = s + (i + 1) * (i + 1)

s = 14 + (3 + 1) * (3 + 1) = 30

У наборі ще є невикористані числа

False

print('Сума дорівнює', s)

Сума дорівнює 30



Функція:

range(<початок>, <кінець>, <крок>)

у загальному випадку має три аргументи цілого типу:

початок

перше ціле число, яке потрапить до згенерованого набору

кінець

ціле число, яке не ввійде до згенерованого набору і яке більше за всі числа з набору, якщо крок > 0, і менше від всіх чисел з набору, якщо крок < 0

крок

різниця між кожними двома сусідніми цілими числами в згенерованому наборі чисел.



Наприклад, функція:

`range(3, 9, 2)`

згенерує такий набір чисел:

3

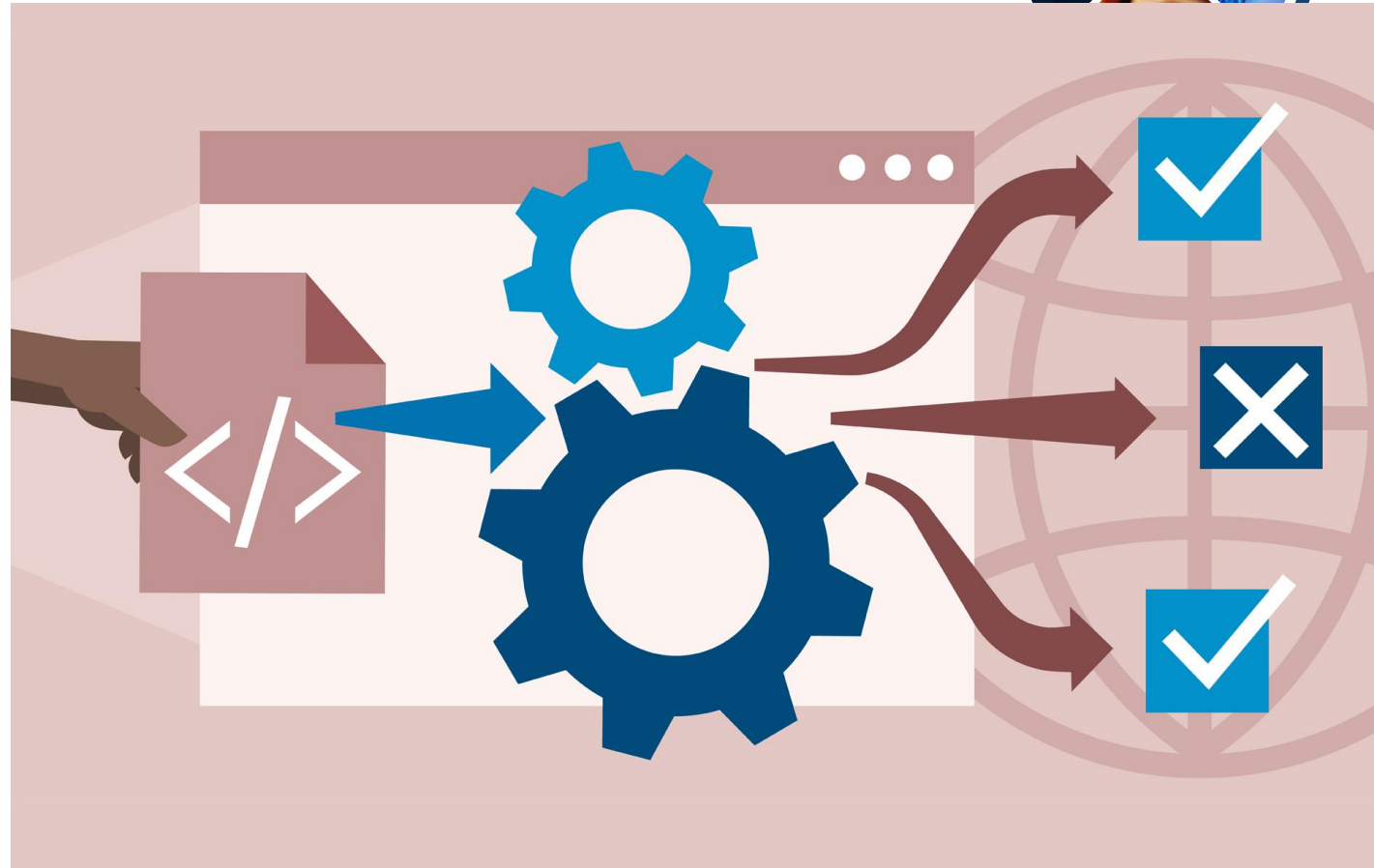
$3 + 2 = 5$

$5 + 2 = 7$

Наступне число $7 + 2 = 9$ не увійде в набір чисел, оскільки воно не менше, ніж другий аргумент функції.



Під час тестування проєкту, що містить цикл з лічильником, доцільно підбирати такі набори вхідних даних, щоб тіло циклу виконувалося різну кількість разів:



1 раз

2 рази

3 рази

5 разів

іншу кількість разів



1. Що таке цикл в алгоритмі?

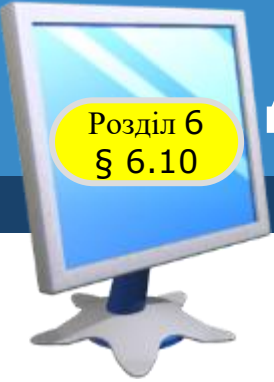
2. Який вигляд має команда циклу з лічильником у мові програмування?

3. Як виконується команда циклу з лічильником у мові програмування?

4. Який вигляд має блок-схема циклу з лічильником? Поясніть виконання такого фрагмента алгоритму.

5. Чи можуть команди тіла циклу з лічильником виконуватися тільки один раз? Поясніть свою відповідь. Наведіть приклад.





***Проаналізувати
§ 6.10, с. 235-240***