

Лабораторна робота №4

Мета роботи: Отримати навички роботи з циклом типу `while` у Python

1. Теоретичні відомості

Типи операторів циклів

При виконанні програми часто виникає необхідність неодноразового повторення однотипних обчислень над різними даними. Для цих цілей використовують так звані цикли.

Цикл представляє собою частину програми, у якій одні й ті самі обчислення реалізуються неодноразово над різними значеннями одних й тих самих змінних (об'єктів).

Для організації циклів в Python використовуються наступні оператори: **while** та **for**.

Цикл типу **while**

Цикл типу **while** є циклом з передумовою. Він використовується у випадку, коли, по-перше, невідома точна кількість повторів і, по-друге, при цьому немає необхідності, щоб цикл неодмінно був виконаний хоча б один раз. Цикл типу **while** має наступну форму запису:

```
while (вираз):
```

```
<Tab>      оператор1
```

```
<Tab>      оператор2
```

```
оператор після циклу
```

В якості виразу зазвичай використовуються умовні вирази. На місці оператора може стояти простий оператор або сукупність операторів

Якщо вираз істинний – True, то тіло циклу виконується один раз, далі вираз перевіряється знову. Ітерації (перевірка умови та тіло циклу) виконуються до тих пір, поки вираз не стане хибним – False.

При організації циклу типу **while** в його тіло повинні бути включені конструкції, які б змінювали вираз, що перевіряється, так, щоб все ж таки він став хибним. В протилежному випадку виконання циклу ніколи не закінчиться.

Приклад 1.

```
temperature = 115
while temperature > 112: # first while loop code
    print(temperature)
    temperature = temperature - 1

print('The tea is cool enough.')
```

Приклад 2.

```
i = 4
while i < 9:
    print(i)
    i = i+2
```

Приклад 3.

Користувачу дається 10 спроб щоб вгадати задане число (скрипт поруч з лабою FindTheNumber.py).

```
num = 15 #Число яке ми загадали
count = 0 #Тут будемо зберігати кількість спроб
test = False #Вводимо додаткову змінну щоб знати, чи вгадали ми
число чи ні
while (count < 10) and (not test):
    print 'You have ', 10 - count, ' attempts' #Виводиться кількість
спроб що лишилися
    ourNum = int(raw_input("Please enter x: "))
    if num == ourNum:
        test = True # Якщо вгадали число, то робимо змінну test рівну
True, що преведе до припинення роботи циклу
        count = count + 1 # Рахуємо кількість використаних спроб

if test == True:
    print 'You win! The number was: ', num, '. You have used ', count,
' attempts'
else:
    print 'Sorry, you have used all attempts'
```

2. Варіанти завдань

1. Ознайомитися з наведеними вище теоретичними відомостями.
2. Виконати приклади, які приводяться в теоретичних відомостях.
3. Виконати наступні завдання:
 - 3.1. Написати три окремі скрипти для кожної з частин завдання
 - 3.2. На початку, у вигляді коментаря, буде містити назву курсу та номер лабораторної роботи (4.1, 4.2, 4.3), а також ваше ім'я та прізвище, та номер Вашої заліковки
 - 3.3. у першому рядку буде виводити назву курсу та номер лабораторної роботи (4.1, 4.2, 4.3)
 - 3.4. у другому – буде виводити ваше ім'я та прізвище, а також номер Вашої заліковки.

I частина

Використовуючи оператор циклу **while**, розв'язати наступні задачі (за варіантом):

1. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{(-1)^{n-1}}{n^n}$$

2. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{1}{2^n} + \frac{1}{3^n}$$

3. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{1}{((3n-2)(3n+1))}$$

4. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{(2n-1)^3}{2^n}$$

5. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{10^n}{n!}$$

6. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{n!}{(2n)!}$$

7. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{n!}{n^n}$$

8. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{2^n n!}{n^n}$$

9. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{3^n n!}{(3n)!}$$

10. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \frac{n!}{3n^n}$$

11. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член

$$a_n = \frac{(n!)^2}{2^{n^2}}$$

якого

12. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого

$$a_n = \lg(n!)e^{-n\sqrt{n}}$$

13. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого $a_n = 10^{-n}(n-1)!$

14. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого $a_n = \frac{n^3}{(3n-3)!}$

15. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого $a_n = \frac{n}{(n-1)^2}$

16. Знайти суму ряду з точністю $\varepsilon=10^{-4}$, загальний член якого $a_n = e^n \cdot 100^{-n^2}$

17. Знайти суму 13 членів ряду, у якому $a_n = \frac{\ln(n!)}{n^2}$

18. Знайти суму 15 членів ряду, у якому $a_n = \frac{n^{\ln n}}{(\ln n)^n}$

II частина

Написати програму, що підраховує кількість цифр у введеному з клавіатури цілому числі n (не використовуєте оператор для визначення довжини рядка, користуйтесь математичними операціями).

III частина

Написати програму, яка обчислює квадратний корінь за ітераційною формулою Герона, з точністю $\varepsilon=10^{-4}$:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right)$$

$$\lim_{n \rightarrow \infty} x_n = \sqrt{a}$$

де a — це фіксоване додатне число, корінь якого ми хочемо обчислити,

а x_1 — будь-яке початкове додатне число.

4. Зберегти скрипти у файлах з наступними назвами `<Surname>_Task4_1.py`, `<Surname>_Task4_2.py`, `<Surname>_Task4_3.py` (наприклад `Rodionov_Task4_1.py`)

5. Запустити даний скрипт за допомогою інтерпретатора та переконатись у його працездатності та правильності результатів.

6. Спробувати здати та захистити лабораторну роботу у викладача практичних занять

Література:

- Числові ряди http://ru.wikipedia.org/wiki/Числовой_ряд