

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
Факультет прикладної математики  
Кафедра прикладної математики

Звіт  
із лабораторної роботи №2  
із дисципліни «Програмування»  
на тему  
«Програмування циклічних алгоритмів»

Виконав:  
студент групи КМ-83  
Касіч Б.В.

Керівник:  
ст.вик.  
Дрозденко О.М.

Київ — 2018

# 1. ПОСТАНОВКА ЗАДАЧІ

## 1.1 Мета роботи

Вивчення засобів реалізації циклічних конструкцій мови Python і отримання навичок складання програм з використанням операторів циклу.

## 1.2 Що потрібно знати

- поняття про циклічну структуру в програмі;
- інструкція *for*;
- функція *range*;
- інструкція *while*;
- вихід з циклу *break* і *continue*.

## 1.3 Узагальнене формулювання завдання до лабораторної роботи

- 1) Вивчити принципи побудови програм циклічної структури.
- 2) Розробити програми з операторами циклу відповідно до варіанта завдання.
- 3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.
- 4) Показати розроблену програму викладачеві.
- 5) Письмово відповісти на Питання для самоперевірки.
- 6) Оформити звіт відповідно до вимог.

Завдання на виконання лабораторної роботи складається з 2-х частин:

- програма, яка реалізує цикл з допомогою оператора *for*;
- програма, яка реалізує цикл через ітераційний процес – оператор *while*.

## 2. ОПИС ПРОГРАМ

### 2.1 Перша програма

За допомогою команди **“print”** ми вітаємо користувача та надаємо йому інформацію про дану програму. За допомогою двох циклів **“while True”** та двох методів **“try – except”** ми налагодили валідацію даних, на випадок, коли користувач введе їх не вірно. Йому запропонують ввести дані заново.

Користувач вводить значення числа **“x”** та натурального числа **“n”** за допомогою команди **“input”**

Далі ми задаємо початкове значення нашого майбутнього результату (**sum=0**)

Потім ми вже реалізуємо саму програму за допомогою ітератора **“for”**.

Для кожного **“i”** з проміжку (**0;n**) ми збільшуємо його на 1 та додаємо всі значення виразів за допомогою операції **“+=”**. Виводимо результат за допомогою команди **“print”**

Далі за допомогою команди **“input”** ми запитуємо у користувача, хоче він завершити програму чи ні. Якщо хоче, то він має написати **“Not”** і програма завершить свою роботу за допомогою **“break”**. Якщо введено щось інше, то програма починає працювати заново за допомогою **“continue”**.

### 2.2 Друга програма

За допомогою команди **“print”** ми вітаємо користувача та надаємо йому інформацію про дану програму. За допомогою циклу **“while True”** ми налагодили валідацію даних, на випадок, коли користувач введе їх не вірно. Йому запропонують ввести дані заново.

Користувач вводить значення числа **“n”** за допомогою команди **“input”**.

Далі ми задаємо початкове значення змінної **“i”**.

Потім ми вже реалізуємо саму програму за допомогою циклу **“while”**.

Допоки **“i\*i <= n”** програма буде виводити значення **“i\*i”** за допомогою команди **“print”**, збільшуючи **“i”** на 1 кожен раз за допомогою операції **“+=”**.

Далі за допомогою команди **“input”** ми запитуємо у користувача, хоче він завершити програму чи ні. Якщо хоче, то він має написати **“Not”** і програма завершить свою роботу за допомогою **“break”**. Якщо введено щось інше, то програма починає працювати заново за допомогою **“continue”**.

### 3. РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

3.1 Усі можливі результати першої програми продемонстровано на Рисунок 3.1

```
=====
                ЛАБОРАТОРНА РОБОТА №2
                ПРОГРАМУВАННЯ ЦИКЛІЧНИХ АЛГОРИТМІВ
                АВТОР: КАСІЧ БОГДАН , КМ-83 (6 ВАРІАНТ)
=====
Добрий день!!! Дана програма знайде суму значень виразу  $(x-i)/i^{**2}$  в залежності від введеного вами числа x
та значень натурального числа i , яке належить проміжку від 1 до введеного вами n
Програма написана за допомогою оператор for
Введіть будь-яке число x: 34
Введіть натуральне число n: 23
50.746876433229396
Print 'Not' if u want to break:
Введіть будь-яке число x: gr
Введіть коректні дані
Введіть будь-яке число x: 34
Введіть натуральне число n: gr
Введіть коректні дані
Введіть натуральне число n: 23
50.746876433229396
Print 'Not' if u want to break: Not
>>>
```

Рисунок 3.1

3.2 Усі можливі результати другої програми продемонстровано на Рисунок 3.2

```
=====
                ЛАБОРАТОРНА РОБОТА №2
                ПРОГРАМУВАННЯ ЦИКЛІЧНИХ АЛГОРИТМІВ
                АВТОР: КАСІЧ БОГДАН , КМ-83 (6 ВАРІАНТ)
=====
Добрий день!!! Дана програма виведе квадрати усіх натуральних чисел,
які не перевищують введеного вами числа n
Програма написана за допомогою цикла while
Введіть ваше число n: 34
1
4
9
16
25
Print 'Not' if u want to break:
Введіть ваше число n: -24
Print 'Not' if u want to break:
Введіть ваше число n: ger
Введіть коректні дані
Введіть ваше число n: 3
1
Print 'Not' if u want to break: Not
>>>
```

Рисунок 3.2

## Текст першої програми:

```
print("=====")
print("          ЛАБОРАТОРНА РОБОТА №2")
print("    ПРОГРАМУВАННЯ ЦИКЛІЧНИХ АЛГОРИТМІВ")
print("    АВТОР: КАСІЧ БОГДАН , КМ-83 (6 ВАРІАНТ)")
print("=====")
print(" Добрий день!!! Дана програма знайде суму значень виразу (x-i)/i**2 в залежності від введеного вами числа x")
print(" та значень натурального числа i , яке належить проміжку від 1 до введеного вами n")
print(" Програма написана за допомогою оператор for")

while True:
    try:
        x=float(input("Введіть будь-яке число x:  "))
    except ValueError:
        print("Введіть коректні дані")
        continue
    while True:
        try:
            n=int(input("Введіть натуральне число n:  "))
        except ValueError:
            print("Введіть коректні дані")
            continue

        sum=0

        for i in range(0,n):
            i += 1
            sum += (x-i)/i**2
        if n >= 1:
            print(sum)
            break
        else:
            print("Введіть коректні дані")
            continue

    answer=input("Print 'Not' if u want to break:  ")
    if answer == "Not":
        break
    else:
        continue
```

## Текст другої програми:

```
print("=====")
print("          ЛАБОРАТОРНА РОБОТА №2")
print("    ПРОГРАМУВАННЯ ЦИКЛІЧНИХ АЛГОРИТМІВ")
print("    АВТОР: КАСІЧ БОГДАН , КМ-83 (6 ВАРІАНТ)")
print("=====")
print(" Добрий день!!! Дана програма виведе квадрати усіх натуральних чисел, ")
print(" які не перевищують введеного вами числа n ")
print(" Програма написана за допомогою цикла while ")

while True:
    try:
        n=float(input("Введіть ваше число n:  "))
    except ValueError:
        print("Введіть коректні дані")
        continue

    i=1

    while (i*i) <= n:
        print(i*i)
        i += 1

    answer=input("Print 'Not' if u want to break:  ")
    if answer=="Not":
        break
    else:
        continue
```

## Відповіді до контрольних запитань

1. Який обчислювальний процес називається циклічним?

Це процес який повторюється визначену або нескінченну кількість разів

2. Чим відрізняється арифметичний цикл від ітераційного?

В арифметичних циклах відразу відома кількість повторів (for)

В ітераційних кількість повторів стає відома тільки після завершення (while)

3. Скільки операторів циклу існує в мові Python?

for і while

4. Що називається параметром циклу, тілом циклу?

Параметр циклу – умова для його виконання

Тіло циклу - блок команд , що виконується в ході циклу

5. Як працює оператор циклу з передумовою?

Якщо умова для виконання циклу є істинною , то цикл буде виконуватися

6. Як можна вийти з циклу примусово?

Для цього існує команда break та continue

7. Які правила пунктуації необхідно дотримувати при записі операторів циклу?

**while** condition:

while condition2:

body

if (enough):

break

else:

continue

**for** var in something/range(x,y):

for var2 in something/range(x,y):

body

## ЗМІСТ

1.Постановка задачі.....	3
2.Опис програм.....	4
3.Результати випробувань.....	5
Текст програм.....	6
Відповіді до контрольних запитань.....	7