

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 2

з дисципліни «**Основи розробки програмного забезпечення мовою
програмування Python**»

Виконала:

Студентка 2 курсу ФІОТ-у
групи ІМ-41

Саламатова Марія Олексіївна

Перевірів:

асистент Фещенко К.Ю.

Київ 2026

Тема роботи

Використання операторів умовного виконання та циклів у мові програмування Python

Мета роботи

Ознайомитися з алгоритмічними структурами керування в мові Python: операторами умовного виконання *if*, *elif*, *else*, а також циклами *for* і *while*. Набути практичних навичок створення програм з розгалуженням та повторенням дій.

Короткі теоретичні відомості

Оператор *if* використовується для перевірки логічної умови. Якщо умова є істинною, виконується відповідний блок коду. У разі, якщо умова хибна, цей блок коду пропускається.

Оператор *else* використовується разом з оператором *if* і визначає блок коду, який виконується у випадку, якщо умова в *if* є хибною.

Оператор *elif* застосовується у випадках, коли необхідно перевірити декілька альтернативних умов. Він дозволяє уникнути вкладених конструкцій *if* і підвищує читабельність програмного коду.

Оператори умовного виконання *if*, *elif*, *else* є базовими інструментами керування логікою виконання програм у мові Python. Їх правильне застосування дозволяє створювати алгоритми, які реагують на різні вхідні дані, забезпечуючи гнучкість та коректність роботи програм.

Цикл *for* використовується для послідовного перебору елементів ітерабельних об'єктів, таких як діапазони чисел, списки, кортежі, рядки, словники.

Цикл *while* використовується для повторення дій доти, доки виконується певна логічна умова.

Цикл *for* доцільно застосовувати тоді, коли відома кількість ітерацій або необхідно перебрати елементи колекції, а цикл *while* – коли повторення залежать від динамічної умови. Цикли *for* і *while* є фундаментальними елементами алгоритмічної структури програм у Python.

Оператор *break* призначений для негайного припинення виконання циклу, в якому він знаходиться. Після виконання *break* програма виходить з тіла циклу та переходить до виконання першої інструкції, що знаходиться після циклу.

Оператор *continue* використовується для пропуску поточної ітерації циклу та негайного переходу до наступної ітерації. При цьому сам цикл не припиняється, а лише пропускається виконання коду, що знаходиться після *continue* в тілі циклу.

Загальне завдання

Написати програму, яка:

- вводить початкові дані;
- виконує розрахунки згідно індивідуального завдання;
- виводить результат;

Індивідуальні завдання

Визначити, чи є введене число додатним або від'ємним.

Алгоритм розв'язання



Текст програми

```
#Ввід даних
number = int(input('Enter your number:'))

#Перевірка чи є число додатнім чи від'ємним
if number > 0:
    print('This number is positive.')
elif number < 0:
    print('This number is negative.')
else:
    print('This number is 0.')
```

Посилання на репозиторій:

<https://github.com/MariaSalamatova/Python-labs>

Результати тестування

```
Enter your number:-11
This number is negative.
```

```
Enter your number:0
This number is 0.
```

```
Enter your number:6
This number is positive.
```

Висновки

В даній лабораторній роботі ми ознайомилися з алгоритмічними структурами керування в мові Python. В даній задачі потрібно було визначити чи є число додатним чи від'ємним: операторами *if*, *elif* ми задали умови якщо число менше чи більше нуля, оператором *else* було задано умову якщо введене число є нулем (адже 0 не є ні додатним ні від'ємним числом).