

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра ІСМ



Звіт  
про виконання лабораторної роботи № 1  
«Введення в Python»  
з дисципліни  
«Спеціалізовані мови програмування»

Виконав:  
Студент групи ІТ-32,  
Вольвенко І. Р.

Прийняв:  
Щербак С.С

Львів 2023

**Мета роботи:** створення консольної програми-калькулятора за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.

## Завдання

### Завдання 1: Введення користувача

Створіть Python-програму, яка приймає введення користувача для двох чисел і оператора (наприклад, +, -, \*, /).

```
Введіть перше число: 4
Введіть оператор (+, -, *, /, ^, √, %,): +
Введіть друге число: 5
```

### Завдання 2: Перевірка оператора

Перевірте чи введений оператор є дійсним (тобто одним із +, -, \*, /). Якщо ні, відобразіть повідомлення про помилку і попросіть користувача ввести дійсний оператор.

```
Введіть перше число: 4
Введіть оператор (+, -, *, /, ^, √, %,): 1
Помилка: Недійсний оператор. Введіть один із +, -, *, /, ^, √, %
```

### Завдання 3: Обчислення

Виконайте обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення) і відобразіть результат.

```
if operator == '+':
    result = num1 + num2
elif operator == '-':
    result = num1 - num2
elif operator == '*':
    result = num1 * num2
elif operator == '/':
    if num2 == 0:
        raise ZeroDivisionE
    result = num1 / num2
```

### Завдання 4: Повторення обчислень

Запитайте користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозвольте йому ввести нові числа і оператор. Якщо ні, вийдіть з програми.

```
Бажаєте виконати ще одне обчислення?(y/n):
```

### Завдання 5: Обробка помилок

Реалізуйте обробку помилок для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідне повідомлення про помилку, якщо виникає помилка.

```
elif operator == '/':  
    if num2 == 0:  
        raise ZeroDivisionError("Ділення на нуль недопустимо.")  
    result = num1 / num2
```

### Завдання 6: Десяткові числа

Змініть калькулятор так, щоб він обробляв десяткові числа (плаваючу кому) для більш точних обчислень.

```
result = round(result, 2)
```

### Завдання 7: Додаткові операції

Додайте підтримку додаткових операцій, таких як піднесення до степеня (^), квадратний корінь (√) і залишок від ділення (%).

```
elif operator == '^':  
    result = num1 ** num2  
elif operator == '√':  
    result = num1 ** 0.5  
elif operator == '%':  
    result = num1 % num2
```

### Завдання 8: Функція пам'яті

Реалізуйте функцію пам'яті, яка дозволяє користувачам зберігати і відновлювати результати. Додайте можливості для зберігання та отримання значень з пам'яті.

```
Введіть перше число: 5  
Введіть оператор (+, -, *, /, ^, √, %): +  
Введіть друге число: 2  
Хочете зберегти результат?(y/n): y  
Результат: 7.0  
Бажаєте виконати ще одне обчислення?(y/n): y  
Для відновлення результату натисніть R. Для виходу - Q.  
  
результат збережених обчислень:  
5.0 + 2.0 = 7.0
```

## Завдання 9: Історія обчислень

Створіть журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозвольте користувачам переглядати історію своїх обчислень.

```
Введіть перше число: 5
Введіть оператор (+, -, *, /, ^, √, %): +
Введіть друге число: 6
Хочете зберегти результат?(y/n): n
Результат: 11.0
Бажаєте виконати ще одне обчислення?(y/n): y
Для відновлення результату натисніть R. Для перегляду історії натисніть H, для переходу на калькулятор введіть K: H

Історія обчислень:
5.0 + 6.0 = 11.0
```

## Завдання 10: Налаштування користувача

Надайте користувачам можливість налаштувати поведінку калькулятора, таку як зміну кількості десяткових розрядів, які відображаються, або налаштування функцій пам'яті.

```
decimal = int(input("Введіть скільки десяткових розрядів після коми показувати: "))
max_history_size = int(input("Введіть скільки обчислень зберігати у історії: "))
```

```
result = round(result, decimal)
```

```
if len(history) >= max_history_size:
    del history[0]
```

**Висновки:** Виконавши ці завдання, ми створили простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проект допоможе нам вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.