

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

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

Виконав:  
студент групи РІ-32  
Гусак Віктор  
Прийняв:  
Щербак С.С.

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

### План роботи

Завдання 1: Введення користувача

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

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.

Завдання 3: Обчислення

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

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.

Завдання 5: Обробка помилок. Реалізуйте обробку помилок для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідне повідомлення про помилку, якщо виникає помилка.

Завдання 6: Десяткові числа

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

Завдання 7: Додаткові операції

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

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.

Завдання 9: Історія обчислень

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

Завдання 10: Налаштування користувача

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

## Хід роботи

### **calculator\_functions.py:**

```
import math

def get_user_input():
    try:
        num1 = float(input("Введіть перше число: "))
        operator = input("Введіть оператор (+, -, *, /, ^, %,
sqrt): ")
        if operator != 'sqrt':
            num2 = float(input("Введіть друге число: "))
        else:
            num2 = None
        return num1, operator, num2
    except ValueError:
        print("Помилка! Введіть правильне число.")
        return get_user_input()

def is_valid_operator(operator):
    if operator in ['+', '-', '*', '/', '^', '%', 'sqrt']:
        return True
    else:
        print("Помилка! Невірний оператор.")
        return False

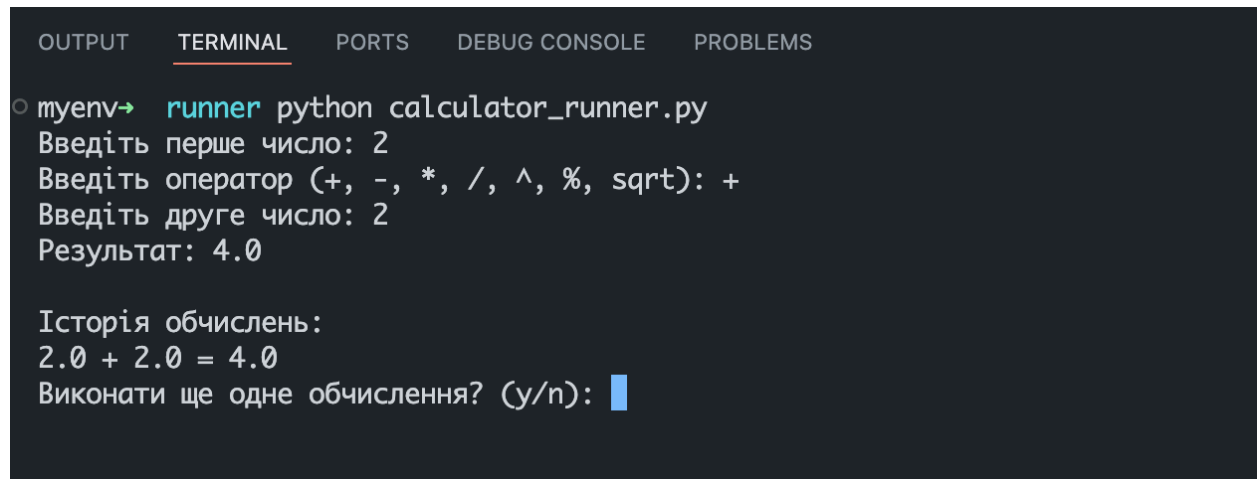
def perform_calculation(num1, operator, num2=None):
    try:
        if operator == '+':
            return num1 + num2
        elif operator == '-':
            return num1 - num2
        elif operator == '*':
            return num1 * num2
        elif operator == '/':
            if num2 == 0:
                return "Помилка! Ділення на нуль."
            return num1 / num2
        elif operator == '^':
            return num1 ** num2
        elif operator == '%':
            return num1 % num2
        elif operator == 'sqrt':
            return math.sqrt(num1)
    except Exception as e:
        return f"Помилка обчислення: {str(e)}"
```

### calculator\_runner.py:

```
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(
__file__))))

from function.calculator_function import run

run()
```



```
myenv - runner python calculator_runner.py
Введіть перше число: 2
Введіть оператор (+, -, *, /, ^, %, sqrt): +
Введіть друге число: 2
Результат: 4.0

Історія обчислень:
2.0 + 2.0 = 4.0
Виконати ще одне обчислення? (y/n):
```

Рис. 1. Приклад роботи програми

**Висновки:** Виконавши ці завдання, було створено простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс.