

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №1  
з дисципліни  
СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ  
на тему  
ВВЕДЕННЯ В PYTHON

Виконав:

ст. гр. РІ-31

ЛАЗАР В.С.

Прийняв:

ЩЕРБАК С.С.

Львів-2024

## **Мета роботи:**

Створення консольної програми-калькулятора за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.

## **Хід роботи:**

### **Завдання № 1: Введення користувача**

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

### **Завдання 2: Перевірка оператора**

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

### **Завдання 3: Обчислення**

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

### **Завдання 4: Повторення обчислень**

Запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозволити йому ввести нові числа і оператор. Якщо ні, вийти з програми.

### **Завдання 5: Обробка помилок**

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

### **Завдання 6: Десяткові числа**

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

### **Завдання 7: Додаткові операції**

Додати підтримку додаткових операцій, таких як піднесення до степеня (^),

квадратний корінь ( $\sqrt{\phantom{x}}$ ) і залишок від ділення (%).

### **Завдання 8: Функція пам'яті**

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

### **Завдання 9: Історія обчислень**

Створити журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозволити користувачам переглядати історію своїх обчислень.

### **Завдання 10: Налаштування користувача**

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

### **Код програми:**

```
from UI.functions import try_again, validate_operator, validate_num, validate_memory,
num_prompt, calculator_result
from DAL.functions import history_write
from GlobalVariables import memory_operations

#False - continue; True - break
def perform():
    num1 = validate_num.validate(num_prompt.prompt(1))

    operator = validate_operator.validate()
    if operator in memory_operations:
        validate_memory.validate(operator, num1)
        return False

    num2 = validate_num.validate(num_prompt.prompt(2))

    result = calculator_result.find(num1, num2, operator)

    if not result:
        return False

    history_write.write(num1, num2, operator, result)

    return try_again.parse(result)
```

На рис. 1-3 зображено результат виконання програми:

```
1 - Calculate a number, 2 - View history, 3 - Additional settings: 1
Enter first number (or MR / MC): 3.14159
Enter operator (or MS / M+ / M-): MS
Memory value stored! Current value: 3.142
1 - Calculate a number, 2 - View history, 3 - Additional settings: 1
Enter first number (or MR / MC): 4
Enter operator (or MS / M+ / M-): *
Enter second number (or MR / MC): MR
Recovered value: 3.142
Result : 12.568
The operation was saved into history
Would you like to try again? (Y / N) // Store a value into memory? (MS / M+ / M-): |
```

*Рис. 1. Приклад роботи калькулятора*

```
1 - Calculate a number, 2 - View history, 3 - Additional settings: 2
Your history:
54.0 + 21.0 = 75.0
3.0 * 7.0 = 21.0
30.0 + 24.0 = 54.0
12.0 * 2.0 = 24.0
5.0 * 6.0 = 30.0
4.0 * 3.142 = 12.568
1 - Calculate a number, 2 - View history, 3 - Additional settings: |
```

*Рис. 2. Приклад роботи функції історії обчислень*

```
1 - Calculate a number, 2 - View history, 3 - Additional settings: 3
1 - Change the amount of digits after a dot in a number, 2 - Clear history:
History cleared successfully
1 - Calculate a number, 2 - View history, 3 - Additional settings: 2
Your history:
1 - Calculate a number, 2 - View history, 3 - Additional settings:
```

*Рис. 3. Приклад роботи додаткових налаштувань*

*(у цьому прикладі функція очистки історії обчислень)*

Посилання на Github: [PaperGlit/Python\\_Lab\\_1](https://github.com/PaperGlit/Python_Lab_1)

### **Висновок:**

Виконавши ці завдання, я створив простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проект допоміг мені вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.