Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 1

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“**Введення в Python**”

Виконав студент РІ-31

**Гуледза Тарас**

Прийняв: Щербак С.С.

Львів – 2024

***Лабораторна робота № 1***

**Тема роботи:** ​“ **Введення в Python** ”

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

**Завдання лабораторної 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.

def calculator():  
 *"""  
 Function to run the calculator. It handles user input, performs operations,  
 stores results in memory, and shows history.  
 """* memory\_value = global\_value.memory\_value  
 while True:  
 try:  
 operator = input("Enter the operator (+, -, \*, /, ^, sq, %): ")  
 while operator not in ['+', '-', '\*', '/', '^', '%', 'sq']:  
 print("Invalid operator. Available operators: +, -, \*, /, ^, sq, %.")  
 operator = input("Enter operator (+, -, \*, /, ^, sq, %): ")  
  
 user\_input = input('Input first operand (or MR for memory recall): ').upper()  
 if user\_input == 'MR':  
 first\_operand = memory\_value  
 print(f"Recalled from memory: {first\_operand}")  
 else:  
 first\_operand = float(user\_input)  
  
 user\_input = input('Input second operand (or MR for memory recall): ').upper()  
 if user\_input == 'MR':  
 second\_operand = memory\_value  
 print(f"Recalled from memory: {second\_operand}")  
 else:  
 second\_operand = float(user\_input)  
  
 result = 0  
 result = functions.choose\_operator(first\_operand, second\_operand, operator)  
 print('Result: ', round(result, global\_value.round\_number))  
  
 functions.log\_history(first\_operand, operator, second\_operand, round(result, global\_value.round\_number))  
 choice\_memory = input(  
 'Would you like to store result in memory (MS), add to memory (M+), clear memory (MC), or skip? '  
 ).upper()  
  
 match choice\_memory:  
 case 'MS':  
 memory\_value = result  
 print(f"Stored {result} in memory.")  
 case 'M+':  
 memory\_value += result  
 print(f"Added {result} to memory. New memory value: {memory\_value}.")  
 case 'MC':  
 memory\_value = 0  
 print("Memory cleared.")  
  
 if input("Do you want to view history? (yes/no): ").strip().lower() == 'yes':  
 print(functions.show\_history())  
  
 if input('Do you want to make another calculation? (yes/no): ').lower() != 'yes':  
 break  
  
 except ValueError as e:  
 print(f"Error: {e}")  
  
  
def run():  
 *"""  
 Main function to display menu and call corresponding functions based on user's choice.  
 """* while True:  
 functions.show\_menu()  
  
 choice = input('Enter your choice: ').strip()  
  
 match choice:  
 case '1':  
 calculator()  
 case '2':  
 app\_settings.setting()  
 case '0':  
 print("Exiting the program. Goodbye!")  
 break

**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.