Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Лабораторна робота №1

з навчальної дисципліни

«Спеціалізовані мови програмування»

на тему

«Введення в Python»

*Інститут комп’ютерних наук*

*та інформаційних технологій*

*Кафедри інформаційних*

*систем та мереж*

***Виконав:***

*Студент групи РІ-31*

*Олександр ЛУЧКЕВИЧ*

***Прийняла:***

*викладач*

*Cергій ЩЕРБАК*

*Львів – 2024*

**Мета ротоби:** 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.

**Код програми:**

Файл lab\_calculator/main:

from functions import calculator, show\_history, log\_history

memory\_value = 0

decimal\_places = 1

def set\_decimal\_places():

global decimal\_places

while True:

try:

decimal\_places = int(input("How many decimal places do you want to display: "))

print(f"Decimal places set to: {decimal\_places}")

break

except ValueError:

print("Incorrect input! Enter an integer.")

def main():

global memory\_value, decimal\_places

while True:

print("\n--- Calculator ---")

print("1. New calculation")

print("2. Show history")

print("3. Set the number of decimal places")

print("4. Exit")

choice = input("Choose option (1-4): ")

if choice == '1':

result, first\_number, operator, second\_number = calculator(decimal\_places, memory\_value)

if result is not None:

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.")

log\_history(first\_number, operator, second\_number, result)

print(f"Result: {result}")

if input("Do you want to view history? (y/n): ").strip().lower() == 'y':

print(show\_history())

if input('Do you want to make another calculation? (y/n): ').lower() != 'y':

break

elif choice == '2':

print(show\_history())

elif choice == '3':

set\_decimal\_places()

elif choice == '4':

print("Exit.")

break

else:

print("Wrong choice! Try again.")

main()

lab\_calculator/functions:

import math

def log\_history(first\_operand, operator, second\_operand, result):

with open('./lab\_calculator/source/history\_log.txt', 'a') as file:

file.write(f"{first\_operand} {operator} {second\_operand} = {result}\n")

def show\_history():

try:

with open('./lab\_calculator/source/history\_log.txt', 'r') as file:

return file.read()

except FileNotFoundError:

return "History is empty."

def input\_number(prompt, memory\_value):

while True:

user\_input = input(prompt).upper()

if user\_input == 'MR':

print(memory\_value)

return memory\_value

try:

return float(user\_input)

except ValueError:

print("Incorrect input! Please enter a valid number or 'MR'.")

def calculator(decimal\_places, memory\_value):

first\_number = input\_number('Input first operand (or MR for memory recall): ', memory\_value)

operator = input("Enter the operator (+, -, \*, /, ^, sq, %): ")

while operator not in ['+', '-', '\*', '/', '^', 'sq', '%']:

print("Invalid operator. Available operators: +, -, \*, /, ^, sq, %.")

operator = input("Enter operator (+, -, \*, /, ^, sq, %): ")

second\_number = input\_number('Input second operand (or MR for memory recall): ', memory\_value)

result = None

if operator == '+':

result = round(first\_number + second\_number, decimal\_places)

elif operator == '-':

result = round(first\_number - second\_number, decimal\_places)

elif operator == '\*':

result = round(first\_number \* second\_number, decimal\_places)

elif operator == '/':

if second\_number != 0:

result = round(first\_number / second\_number, decimal\_places)

else:

print("Error: division by zero!")

elif operator == '^':

result = round(first\_number \*\* second\_number, decimal\_places)

elif operator == 'sq':

result = round(first\_number \*\* (1 / second\_number), decimal\_places)

elif operator == '%':

if second\_number != 0:

result = round(first\_number % second\_number, decimal\_places)

else:

print("Error: division by zero!")

print(f"{first\_number} {operator} {second\_number} = {result}")

return result, first\_number, operator, second\_number # Return result and operands

**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.