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

Виконала:

ст. гр. ІТ-21сп

Калінка Ю.В.

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.

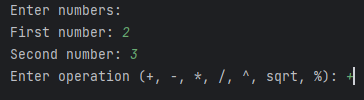
**План роботи**

Завдання 1. Введення користувача

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

print("Enter numbers:")  
 try:  
 num1 = float(input("First number: "))  
 num2 = float(input("Second number: "))

На рисунку 1 зображено код виконання програми.



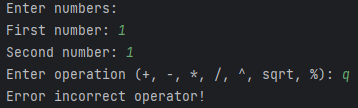
*Рис.1 Код виконання програми*

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.

while True:  
 result = ''  
 print("Enter numbers:")  
 try:  
 num1 = float(input("First number: "))  
 num2 = float(input("Second number: "))  
 except ValueError:  
 print("Error incorrect input!")  
 continue  
  
 # Визначте функції для операцій  
 operations = {  
 '+': summ,  
 '-': substraction,  
 '\*': multiplication,  
 '/': division,  
 '^': power,  
 'sqrt': square,  
 '%': remains  
 }  
  
 # Отримайте оператор від користувача  
 operation = input("Enter operation (+, -, \*, /, ^, sqrt, %): ")  
  
 # Перевірте, чи оператор існує у словнику функцій  
 if operation in operations:  
 if operation == 'sqrt':  
 result = operations[operation](num1)  
 else:  
 result = operations[operation](num1, num2)  
 else:  
 print("Error incorrect operator!")

На рисунку 2 зображено результат виконання задачі.



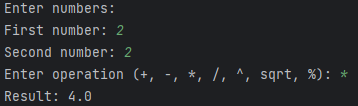
*Рис.2 Результат виконання задачі*

Завдання 3. Обчислення.

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

while True:  
 result = ''  
 print("Enter numbers:")  
 try:  
 num1 = float(input("First number: "))  
 num2 = float(input("Second number: "))  
 except ValueError:  
 print("Error incorrect input!")  
 continue  
  
 # Визначте функції для операцій  
 operations = {  
 '+': summ,  
 '-': substraction,  
 '\*': multiplication,  
 '/': division,  
 '^': power,  
 'sqrt': square,  
 '%': remains  
 }  
  
 # Отримайте оператор від користувача  
 operation = input("Enter operation (+, -, \*, /, ^, sqrt, %): ")

На рисунку 3 зображено обчислення.



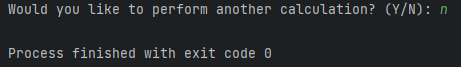
*Рис.3 Виконання програми*

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.

continueCalc = input("Would you like to perform another calculation? (Y/N): ")  
if continueCalc.lower() != 'y':  
 break

На рисунку 4 вивід роботи програми.



*Рис.4 Вивід роботи програми*

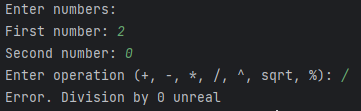
Завдання 5. Обробка помилок.

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

def division(x, y):  
 if y == 0:  
 return "Error. Division by 0 unreal"  
 return x / y

def square(x):  
 if x < 0:  
 return "The square root of a negative number is impossible"  
 return math.sqrt(x)

На рисунку 5 зображено результат програми.



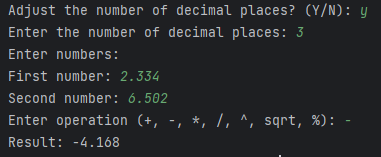
*Рис.5 Результат програм*

Завдання 6. Десяткові числа.

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

num1 = float(input("First number: "))  
num2 = float(input("Second number: "))

На рисунку 6 вивід виконаного завдання.



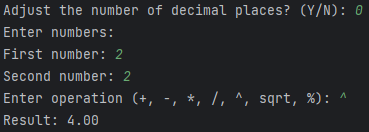
*Рис.6 Вивід виконаного завдання*

Завдання 7. Додаткові операції.

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

def power(x, y):  
 return pow(x, y)  
  
def remains(x, y):  
 return x % y  
  
def square(x):  
 if x < 0:  
 return "The square root of a negative number is impossible"  
 return math.sqrt(x)

На рисунку 7 відображення результату програми.



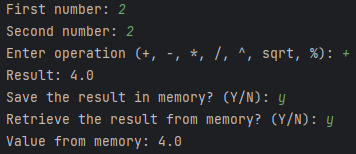
*Рис.7 Результат виконання програми*

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.

save = input("Save the result in memory? (Y/N): ")  
if save.lower() == 'y':  
 memory = result  
  
retrieve = input("Retrieve the result from memory? (Y/N): ")  
if retrieve.lower() == 'y' and memory is not None:  
 print("Value from memory:", memory)

На рисунку 8 зображено виконання програми.



*Рис.8 Код виконання програми*

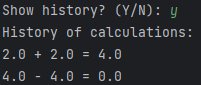
Завдання 9. Історія обчислень.

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

calculation = f"{num1} {operation} {num2} = {result}"  
history.append(calculation)

view\_history = input("Show history? (Y/N): ")  
if view\_history.lower() == 'y':  
 print("History of calculations:")  
 for entry in history:  
 print(entry)

На рисунку 9 зображено роботу калькулятора.



*Рис.9 Вивід роботи програми*

Завдання 10. Налаштування користувача.

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

configure\_decimal = input("Adjust the number of decimal places? (Y/N): ")  
if configure\_decimal.lower() == 'y':  
 try:  
 decimal\_places = int(input("Enter the number of decimal places: "))  
 except ValueError:  
 print("Error incorrect input!")

На рисунку 10 вивід роботи програми.



*Рис.10 Виконання завдання*

Загальний вигляд коду програми:

import math  
  
def summ(x, y):  
 return x + y  
  
def substraction(x, y):  
 return x - y  
  
def multiplication(x, y):  
 return x \* y  
  
def division(x, y):  
 if y == 0:  
 return "Error. Division by 0 unreal"  
 return x / y  
  
def power(x, y):  
 return pow(x, y)  
  
def remains(x, y):  
 return x % y  
  
def square(x):  
 if x < 0:  
 return "The square root of a negative number is impossible"  
 return math.sqrt(x)  
  
memory = None  
history = []  
decimal\_places = 2  
  
configure\_decimal = input("Adjust the number of decimal places? (Y/N): ")  
if configure\_decimal.lower() == 'y':  
 try:  
 decimal\_places = int(input("Enter the number of decimal places: "))  
 except ValueError:  
 print("Error incorrect input!")  
  
while True:  
 result = None # Initialize result as None  
 print("Enter numbers:")  
 try:  
 num1 = float(input("First number: "))  
 num2 = float(input("Second number: "))  
 except ValueError:  
 print("Error incorrect input!")  
 continue  
  
 operations = {  
 '+': summ,  
 '-': substraction,  
 '\*': multiplication,  
 '/': division,  
 '^': power,  
 'sqrt': square,  
 '%': remains  
 }  
  
 operation = input("Enter operation (+, -, \*, /, ^, sqrt, %): ")  
  
 if operation in operations:  
 if operation == 'sqrt':  
 result = operations[operation](num1)  
 else:  
 result = operations[operation](num1, num2)  
 else:  
 print("Error incorrect operator!")  
  
 if isinstance(result, (float, int)): # Check if result is a number  
 rounded\_result = round(result, decimal\_places)  
 print(f"Result: {rounded\_result:.{decimal\_places}f}")  
 else:  
 print(result) # Print the error message  
  
 calculation = f"{num1} {operation} {num2} = {result}"  
 history.append(calculation)  
  
 save = input("Save the result in memory? (Y/N): ")  
 if save.lower() == 'y':  
 memory = result  
  
 retrieve = input("Retrieve the result from memory? (Y/N): ")  
 if retrieve.lower() == 'y' and memory is not None:  
 print("Value from memory:", memory)  
  
 view\_history = input("Show history? (Y/N): ")  
 if view\_history.lower() == 'y':  
 print("History of calculations:")  
 for entry in history:  
 print(entry)  
  
 continueCalc = input("Would you like to perform another calculation? (Y/N): ")  
 if continueCalc.lower() != 'y':  
 break

**Висновок:** під час виконання лабораторної роботи я створила простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проєкт допоміг мені вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.