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

Кобленц Д.А.

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.

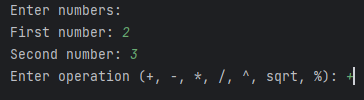
**План роботи**

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.

def get\_operator():

while True:

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

if value not in ('+', '-', '\*', '/', '^', '%', '√'):

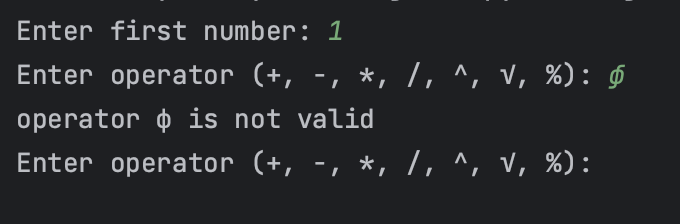
print(f"operator {value} is not valid")

else:

return value

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.

def calculate(num1, operator, num2):

if operator == '+':

return num1 + num2

elif operator == '-':

return num1 - num2

elif operator == '\*':

return num1 \* num2

elif operator == '/':

if num2 == 0:

print("Error: you can't divide by zero")

return None

else:

return num1 / num2

elif operator == '^':

return num1 \*\* num2

elif operator == '√':

return math.sqrt(num1)

elif operator == '%':

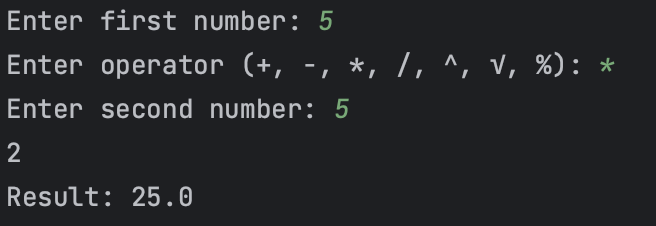
return num1 % num2

На рисунку 3 зображено обчислення.

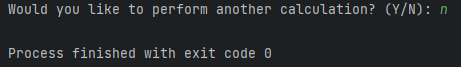
*Рис.3 Виконання програми*

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*

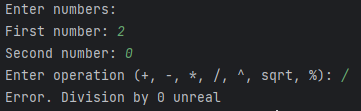
Завдання 5. Обробка помилок.

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.



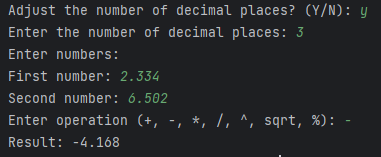
*Рис.5 Результат програм*

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.



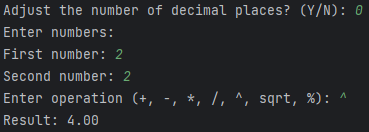
*Рис.6 Вивід виконаного завдання*

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.



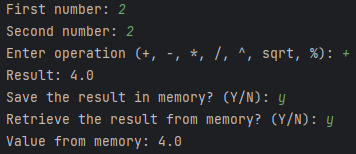
*Рис.7 Результат виконання програми*

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*

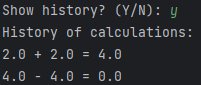
Завдання 9. Історія обчислень.

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.

Рисунок 22

*Рис.10 Виконання завдання*

Загальний вигляд коду програми:

import math

memory = None

history = []

settings = {

"decimal\_places": 2,

"use\_memory": True

}

def show\_history():

global history

print("History:")

for item in history:

print(item)

def save(value):

global memory

memory = value

def get\_operator():

while True:

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

if value not in ('+', '-', '\*', '/', '^', '%', '√'):

print(f"operator {value} is not valid")

else:

return value

def get\_number(message):

global memory

global settings

if memory is not None and settings["use\_memory"] :

print(f"Value in memory {memory}")

use\_from\_memory = input("Do you want to use value from memory? (yes/no)")

if use\_from\_memory.lower() == 'yes':

return memory

while True:

try:

return float(input(message))

except:

print("Incorrect number! Enter valid")

def calculate(num1, operator, num2):

if operator == '+':

return num1 + num2

elif operator == '-':

return num1 - num2

elif operator == '\*':

return num1 \* num2

elif operator == '/':

if num2 == 0:

print("Error: you can't divide by zero")

return None

else:

return num1 / num2

elif operator == '^':

return num1 \*\* num2

elif operator == '√':

return math.sqrt(num1)

elif operator == '%':

return num1 % num2

def change\_settings():

global settings

print("1. Change decimal places.")

print("2. Toggle memory function.")

choice = input("Enter your choice: ")

if choice == '1':

try:

places = int(input("Enter number of decimal places (0-10): "))

if 0 <= places <= 10:

settings["decimal\_places"] = places

print(f"Set decimal places to {places}.")

else:

print("Invalid number of decimal places.")

except ValueError:

print("Invalid input.")

elif choice == '2':

settings["use\_memory"] = not settings["use\_memory"]

status = "enabled" if settings["use\_memory"] else "disabled"

print(f"Memory function is now {status}.")

def get\_choice():

while True:

print("\nCalculator Menu:")

print("1. Continue.")

print("2. View history.")

print("3. Change settings.")

print("4. Quit.")

choice = input("Enter your choice: ")

if choice == "1":

break

elif choice == "2":

show\_history()

continue\_choice = input("Do you want to continue? (yes/no)")

if continue\_choice.lower() != 'yes':

return 'close'

elif choice == "3":

change\_settings()

continue\_choice = input("Do you want to continue? (yes/no)")

if continue\_choice.lower() != 'yes':

return 'close'

elif choice == "4":

return 'close'

else:

print("Enter valid option")

while True:

first\_num = get\_number("Enter first number: ")

operation = get\_operator()

second\_num = get\_number("Enter second number: ")

result = calculate(first\_num, operation, second\_num)

if result is None:

continue

print(settings['decimal\_places'])

result = round(result, settings["decimal\_places"])

print(f"Result: {result}")

history.append(f"{first\_num} {operation} {second\_num} = {result}")

if settings["use\_memory"]:

save\_choice = input("Do you want to save result? (yes, no)")

if save\_choice == 'yes':

save(result)

print(f"Value {result} is saved")

else:

memory = None

option = get\_choice()

if option == "close":

break

**Висновок:** під час виконання лабораторної роботи я створив 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.