МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №1

з дисципліни “Спеціалізовані мови програмування”

Виконала:

студентка групи ІТ-31

Анастасія КАМІНОВСЬКА

Прийняв:

доцент кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

**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:

Хід 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 :

"""

Simple Calculator Module

This module provides a basic Calculator class for performing arithmetic operations.

Classes:

Calculator: A class that allows users to perform simple arithmetic calculations.

Functions:

None

Usage:

from calculator import Calculator

# Create an instance of the Calculator class

calc = Calculator()

# Run the calculator program

calc.run()

"""

import math

import logging

class Calculator:

"""

Simple calculator class for basic arithmetic operations.

Attributes:

num1 (float): The first operand.

num2 (float): The second operand.

operator (str): The arithmetic operator.

"""

def \_\_init\_\_(self):

"""

Initializes an instance of the Calculator class.

This method sets the initial values for num1, num2, and operator,

and logs the initiation of the calculator.

"""

self.num1 = None

self.num2 = None

self.operator = None

logging.info("Калькулятор ініційовано")

def get\_user\_input(self):

"""

Gets input from the user for the operands and operator.

This method prompts the user to enter the first number, the

arithmetic operator, and the second number (if applicable). It

also performs input validation and logs the input data.

"""

try:

self.num1 = float(input("Введіть перше число: "))

self.operator = input("Введіть операцію (+, -, \*, /, ^, sq, %): ")

if self.operator not in ('+', '-', '\*', '/', '^', 'sq', '%'):

logging.warning("Невідомий оператор введено")

print("Помилка: Невідомий оператор.")

return

if self.operator in ('+', '-', '\*', '/', '^', 'sq', '%'):

if self.operator != 'sq':

self.num2 = float(input("Введіть друге число: "))

else:

self.num2 = None

logging.info("Отримано вхідні дані: %s, %s, %s", self.num1, self.operator, self.num2)

except ValueError:

logging.error("Неправильне введення числа")

print("Помилка: Введіть коректні числа.")

def calculate(self):

"""

Performs the calculation based on the user's input.

This method calculates the result based on the entered operands

and operator. It logs the calculated result and handles errors.

"""

try:

result = None

if self.operator is not None:

num1, num2 = self.num1, self.num2 if self.num2 is not None else 0

match self.operator:

case '+':

result = num1 + num2

case '-':

result = num1 - num2

case '\*':

result = num1 \* num2

case '/':

if num2 != 0:

result = num1 / num2

else:

logging.error("Спроба ділення на нуль")

print("Помилка: Ділення на нуль.")

return

case '^':

result = num1 \*\* num2

case 'sq':

result = math.sqrt(num1)

case '%':

result = num1 % num2

case \_:

print("Помилка: Невідомий оператор.")

return

print(f"Результат: {self.num1} {self.operator} "

f"{self.num2 if self.num2 is not None else ''} = {result:.2f}")

logging.info("Обраховано результат: %s", result)

except Exception as e:

logging.exception("Виникла помилка при обрахунку")

print(f"Помилка: {e}")

def run(self):

"""

Runs the main loop of the calculator program.

This method continuously prompts the user for input, performs

calculations, and asks if the user wants to continue. It logs the

termination of the calculator when the user chooses to exit.

"""

while True:

self.get\_user\_input()

self.calculate()

choice = input("Бажаєте продовжити (так/ні)? ").lower()

if choice != 'так':

logging.info("Завершення роботи калькулятора")

break

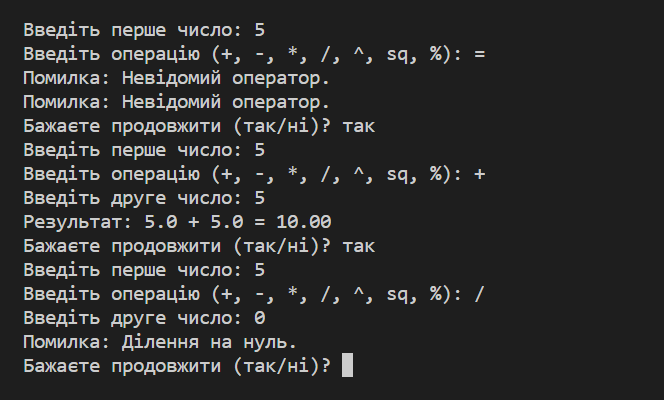


Рис.1 Результат виконання роботи

**Висновок**: Виконуючи лабораторну роботу я створила простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. Цей проект допоміг вам вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.