Міністерство освіти і науки України Національний університет “Львівська політехніка” Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 1

***“ Введення в Python”***

з дисципліни “***Спеціальні мови програмування*** ”

Виконав студент групи IT-32

Дякун Б.В.

Прийняв: Щербак С.С

# 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:

**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.

**Хід роботи**

**Реалізований код:**

import math

# Історія обчислень

calc\_history = []

class CalculatorSettings:

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

self.decimal\_places = 2 # Кількість десяткових знаків

self.memory = 0.0 # Значення в пам'яті

def set\_decimal\_places(self):

try:

self.decimal\_places = int(input("Введіть кількість десяткових знаків: "))

except ValueError:

print("Помилка введення. Використовується значення за замовчуванням (2).")

def set\_memory(self, value):

self.memory = value

def get\_memory(self):

return self.memory

def add\_to\_history(expression, result):

calc\_history.append((expression, result))

def display\_history():

for i, (expression, result) in enumerate(calc\_history, 1):

print(f"{i}. {expression} = {result}")

def display\_settings\_menu(settings):

print("\nНалаштування:")

print("1. Кількість десяткових знаків")

print("2. Пам'ять")

print("3. Вихід з меню налаштувань")

def calculator():

settings = CalculatorSettings()

while True:

try:

# Введення користувача

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

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

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

# Перевірка оператора

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

print("Помилковий оператор. Будь ласка, введіть дійсний оператор.")

continue

# Обчислення

if operator == '+':

result = num1 + num2

elif operator == '-':

result = num1 - num2

elif operator == '\*':

result = num1 \* num2

elif operator == '/':

if num2 == 0 or num1 == 0:

raise ZeroDivisionError("Ділення на нуль недопустиме.")

result = num1 / num2

elif operator == '^':

result = num1 \*\* num2

elif operator == '√':

result = math.sqrt(num1)

elif operator == '%':

result = num1 % num2

# Округлення результату

result = round(result, settings.decimal\_places)

# Збереження в історію

expression = f"{num1} {operator} {num2}"

add\_to\_history(expression, result)

# Виведення результату

print(f"Результат: {result}")

# Повторення обчислень

another\_calculation = input("Виконати ще одне обчислення? (Так/Ні): ").strip().lower()

if another\_calculation != 'так':

break

except ValueError:

print("Помилка введення. Будь ласка, введіть коректні числа.")

except ZeroDivisionError as e:

print(f"Помилка: {e}")

def settings\_menu(settings):

while True:

display\_settings\_menu(settings)

choice = input("Виберіть налаштування: ")

if choice == '1':

settings.set\_decimal\_places()

elif choice == '2':

try:

value = float(input("Введіть значення для пам'яті: "))

settings.set\_memory(value)

except ValueError:

print("Помилка введення. Будь ласка, введіть коректне число.")

elif choice == '3':

break

else:

print("Недійсний вибір.")

if \_\_name\_\_ == "\_\_main\_\_":

settings = CalculatorSettings()

while True:

main\_menu\_choice = input("Обчислення (1) / Налаштування (2) / Історія (3) / Вихід (4): ").strip()

if main\_menu\_choice == '1':

calculator()

elif main\_menu\_choice == '2':

settings\_menu(settings)

elif main\_menu\_choice == '3':

display\_history()

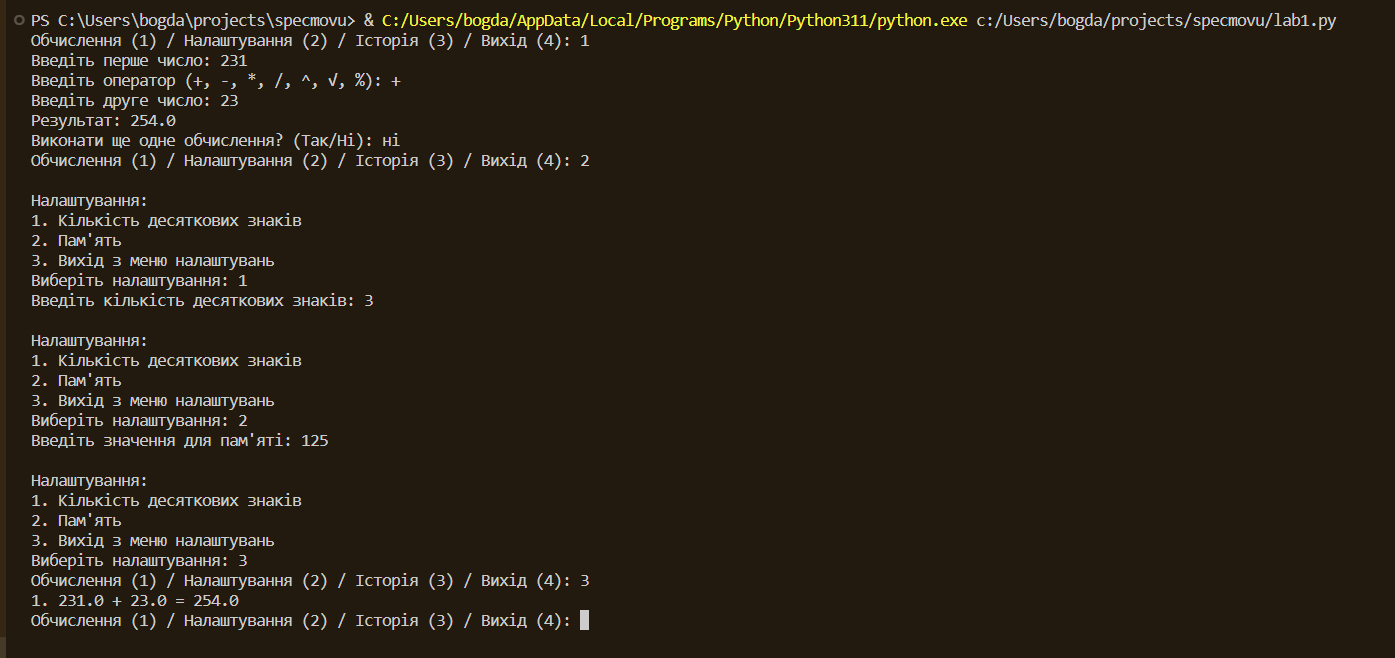
elif main\_menu\_choice == '4':

break

else:

print("Недійсний вибір.")

**Результат виконання программи**:



**Висновок:** Виконавши ці завдання, ми створили простий консольний калькулятор на Python, який може виконувати арифметичні операції, обробляти помилки та надавати користувачу зручний інтерфейс. У цьому проекті ми вивчили основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок.