# **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА” ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ** **Кафедра ІСМ**

# **Звіт**

# **до лабораторної роботи №1**

# **З дисципліни “Спеціальні мови програмування”**

**Виконав:  
ст.гр.ІТ-31  
Шельвах Максим  
  
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 # Імпортуємо бібліотеку для обчислення квадратного кореня, піднесення до степеня та знаходження відсотка

# Налаштування користувача

decimal\_places = 2 # Кількість десяткових розрядів за замовчуванням

memory = 0 # Ініціалізуємо пам'ять значенням 0

history = [] # Ініціалізуємо список для збереження історії обчислень

while True:

try:

if history:

print("Історія обчислень:")

for i, entry in enumerate(history, 1):

print(f"{i}. Вираз: {entry['expression']}, Результат: {entry['result']}")

# Просимо користувача ввести перше число

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

# Просимо користувача ввести друге число

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

while True:

# Просимо користувача ввести оператор (+, -, \*, /, ^, √, %, M - для збереження в пам'ять, D - для зміни десяткових розрядів)

operator = input("Введіть оператор (+, -, \*, /, ^, √, %, M - для збереження в пам'ять, D - для зміни десяткових розрядів): ")

# Перевіряємо чи оператор є допустимим (+, -, \*, /, ^, √, %, M, D)

if operator in ('+', '-', '\*', '/', '^', '√', '%', 'M', 'D'):

break # Виходимо з циклу, якщо оператор коректний

else:

print("Помилка! Введіть дійсний оператор (+, -, \*, /, ^, √, %, M, D).")

# Змінюємо кількість десяткових розрядів за бажанням користувача

if operator == "D":

decimal\_places = int(input("Введіть кількість десяткових розрядів: "))

continue # Перейти до наступної ітерації циклу

# Проводимо обчислення залежно від обраного оператора

if operator == "+":

result = num1 + num2

elif operator == "-":

result = num1 - num2

elif operator == "\*":

result = num1 \* num2

elif operator == "/":

# Перевіряємо, щоб уникнути ділення на нуль

if num2 == 0:

raise ZeroDivisionError("Ділення на нуль неможливе.")

result = num1 / num2

elif operator == "^":

result = num1 \*\* num2

elif operator == "√":

# Використовуємо функцію math.sqrt для обчислення квадратного кореня

result = math.sqrt(num1)

elif operator == "%":

result = num1 % num2

elif operator == "M":

memory = result = 0

print(f"Значення {memory} збережено в пам'яті.")

# Форматуємо результат з вказаною кількістю десяткових розрядів

formatted\_result = round(result, decimal\_places)

# Виводимо результат обчислення

print(f"Результат обчислення: {formatted\_result}")

# Додаємо запис до історії обчислень

history\_entry = {'expression': f"{num1} {operator} {num2}", 'result': formatted\_result}

history.append(history\_entry)

except ValueError:

print("Помилка! Введено недійсне число.")

except ZeroDivisionError as e:

print(f"Помилка: {e}")

# Питаємо користувача, чи він хоче виконати ще одне обчислення

repeat = input("Виконати ще одне обчислення? (Так/Ні): ").lower()

if repeat != "так":

break # Виходимо з головного циклу, якщо користувач відповів "ні"

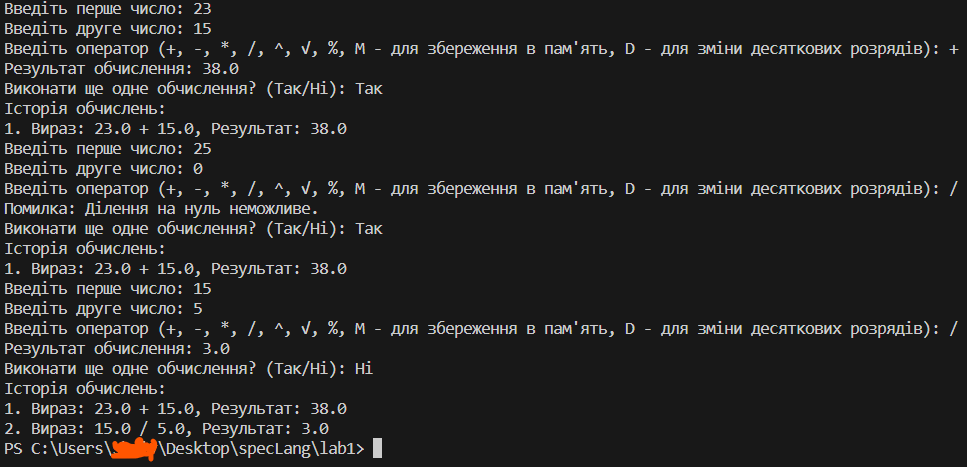
# Виводимо історію обчислень при завершенні програми

print("Історія обчислень:")

for i, entry in enumerate(history, 1):

print(f"{i}. Вираз: {entry['expression']}, Результат: {entry['result']}")

Результат виконання програми:



**Висновок:** виконавши ці завдання, ми створили консольну програму-калькулятор за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.