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

Пазюк С.А.

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

# Функція для обчислення виразу

def calculate(num1, operator, num2):

    try:

        if operator == '+':

            result = num1 + num2

        elif operator == '-':

            result = num1 - num2

        elif operator == '\*':

            result = num1 \* num2

        elif operator == '/':

            if num2 == 0:

                raise ZeroDivisionError("Ділення на нуль неможливе.")

            result = num1 / num2

        elif operator == '^':

            result = num1 \*\* num2

        elif operator == '√':

            result = math.sqrt(num1)

        elif operator == '%':

            result = num1 % num2

        else:

            raise ValueError("Невідомий оператор")

        return result

    except Exception as e:

        return str(e)

# Функція для збереження та відновлення результату

memory = None

def save\_to\_memory(result):

    global memory

    memory = result

def recall\_memory():

    global memory

    return memory

# Функція для виведення історії обчислень

history = []

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

    history.append((expression, result))

def display\_history():

    if not history:

        print("Історія обчислень порожня.")

    else:

        print("Історія обчислень:")

        for i, (expression, result) in enumerate(history, 1):

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

# Основний цикл програми

while True:

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

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

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

    result = calculate(num1, operator, num2)

    if isinstance(result, float):

        result\_str = f"{result:.6f}"

    else:

        result\_str = result

    print(f"Результат: {result\_str}")

    add\_to\_history(f"{num1} {operator} {num2}", result\_str)

    option = input("Бажаєте зберегти результат в пам'яті? (y/n): ")

    if option.lower() == 'y':

        save\_to\_memory(result\_str)

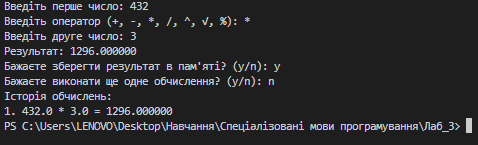
    option = input("Бажаєте виконати ще одне обчислення? (y/n): ")

    if option.lower() != 'y':

        break

display\_history()

**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.