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»

Виконав:

ст. гр. РІ-32

Маньянов М.Ю.

Прийняв:

доц. каф. ІСМ

Щербак С. С.

Львів - 2024

**Мета**

Створити простий консольний калькулятор на Python, який може виконувати арифметичні операціїї, обробляти помилки та надавати користувачу зручний інтерфейс. Вивчити основний синтаксис Python і концепції, такі як введення користувача, умовні оператори, цикли та обробка помилок

**Хід виконання роботи**

**Завдання 1: Введення користувача**

Python-програма, яка приймає введення користувача для двох чисел і оператора.

Створено функцію яка запитує в користувача перше число, оператор та інше число, а потім повертає їх.

**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.

Задля цього додаємо перетворення вхідних рядків на цілі числа, огортаємо даний код в try except блок та при ValueError виводимо помилку на екран.

**Завдання 3: Обчислення**

Виконайте обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення) і відобразіть результат.

Створено ще одну функцію, яка отримує операнди та оператор та повертає результат, який згодом виводиться на екран.

**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.

Гортаємо код з запитом користувача в нескінченний цикл та додаємо опцію вийти з програми або продовжити обчислення.

**Завдання 5: Обробка помилок**

Реалізуйте обробку помилок для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідне повідомлення про помилку, якщо виникає помилка.

Додаємо до операцій перевірку чи підходять операнди для операцій, та якщо ні викидаємо помилки. Огортаємо цей код в try except та в разі помилки в термінал пишемо про помилку.

**Завдання 6: Десяткові числа**

Змініть калькулятор так, щоб він обробляв десяткові числа (плаваючу кому) для більш точних обчислень.

Заміняємо перетворення рядка в ціле число на перетворення рядка в число з плаваючою комою.

**Завдання 7: Додаткові операції**

Додайте підтримку додаткових операцій, таких як піднесення до степеня (^), квадратний корінь (√) і залишок від ділення (%).

До match блоку додаємо операції, до квадратного кореня треба додати перевірку чи число не від’ємне.

**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.

Окремо створюємо функції, які приймають пам’ять та значення якщо необхідно, та повертають нове значення пам’яті. Також створюємо змінну пам’яті.

**Завдання 9: Історія обчислень**

Створіть журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозвольте користувачам переглядати історію своїх обчислень.

Ініціалізуємо масив, в який при успішних операціях будуть додаватись об’єкти з числами, оператором та результатом. Додаємо в меню можливість переглядати історію.

**Завдання 10: Налаштування користувача**

Надайте користувачам можливість налаштувати поведінку калькулятора, таку як зміну кількості десяткових розрядів, які відображаються, або налаштування функцій пам'яті.

Створимо додаткову змінну, яка означає кількість знаків після коми, створюємо функцію яка форматує число з плаваючою комою в рядок, в відображення чисел додаємо цю функцію.

Тепер скористаємось програмою.

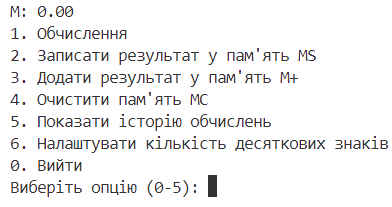
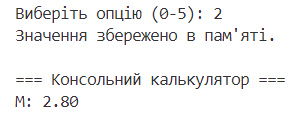
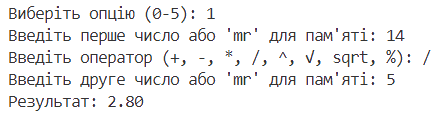


Рис. 1. Початкове меню програми



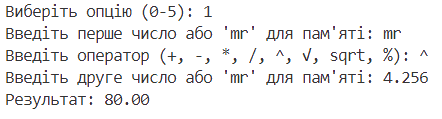
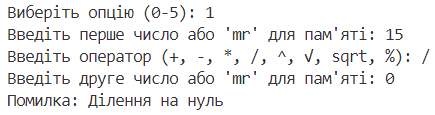


Рис. 2-4. Функції пам’яті



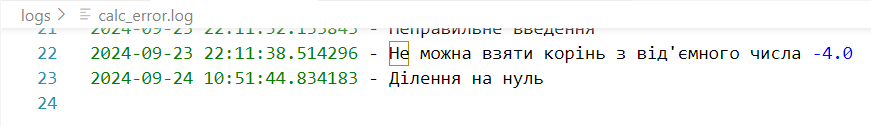
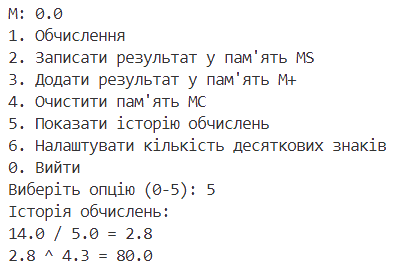
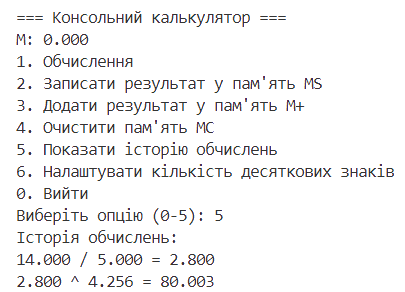


Рис. 5-6. Обробка помилок.



****Рис. 7. Історія обчислень

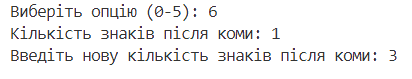


Рис. 8-9. Зміна знаків після коми та результат відображення.

**Висновок :**

Я створив простий консольний калькулятор на Python, котрий також може перевіряти помилки, відображати та записувати їх у файли, реалізує функцію пам’яті. Я вивчив основний синтаксис Python та його концепції введення користувача, обробку помилок, умовні оператори, імпорт модулів, цикли тощо.