мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

|  |
| --- |
| 2023 |



Звіт

Про виконання лабораторної роботи №2

З предмету «Спеціалізовані мови програмування»

Виконав

ст.гр ІТ-21

Капанайко А.Т

Прийняв:

доц.каф

Щербак С.С

**Мета роботи:** Розробка консольного калькулятора в об’єктно орієнтованому стилі з використанням класів

**План роботи:**

**Завдання 1:** Створення класу Calculator

Створіть клас Calculator, який буде служити основою для додатка калькулятора.

**Завдання 2:** Ініціалізація калькулятора

Реалізуйте метод \_\_init\_\_ у класі Calculator для ініціалізації необхідних атрибутів або змінних.

**Завдання 3:** Введення користувача

Перемістіть функціональність введення користувача в метод у межах класу Calculator. Метод повинен приймати введення для двох чисел і оператора.

**Завдання 4:** Перевірка оператора

Реалізуйте метод у класі Calculator, щоб перевірити, чи введений оператор є дійсним (тобто одним із +, -, \*, /). Відобразіть повідомлення про помилку, якщо він не є дійсним.

**Завдання 5:** Обчислення

Створіть метод у класі Calculator, який виконує обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення).

**Завдання 6**: Обробка помилок

Реалізуйте обробку помилок у межах класу Calculator для обробки ділення на нуль або інших потенційних помилок. Відобразіть відповідні повідомлення про помилку.

**Завдання 7:** Повторення обчислень

Додайте метод до класу Calculator, щоб запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозвольте йому ввести нові числа і оператор. Якщо ні, вийдіть з програми.

**Завдання 8:** Десяткові числа

Модифікуйте клас Calculator для обробки десяткових чисел (плаваюча кома) для більш точних обчислень.

**Завдання 9:** Додаткові операції

Розширте клас Calculator, щоб підтримувати додаткові операції, такі як піднесення до степеня (^), квадратний корінь (√) та залишок від ділення (%).

**Завдання 10:** Інтерфейс, зрозумілий для користувача

Покращте інтерфейс користувача у межах класу Calculator, надавши чіткі запити, повідомлення та форматування виводу для зручності читання.

**Код програми:**

import math

memory = []

history = []

decimals =

print("Starting...")

while True:

    value1 = float(input("Enter your first value: "))

    value2 = float(input("Enter your second value: "))

    while True:

        print("1. +")

        print("2. -")

        print("3. \*")

        print("4. /")

        print("5. ^")

        print("6. √")

        print("7. %")

        chooseOperator = input("Enter an arithmetic operator you prefer: ")

        if chooseOperator not in ["+", "-", "\*", "/", "^", "√", "%"]:

            print(

                "Invalid operator has been chosen. Please, follow the list and try one more time!. Please, reboot the program."

            )

            break

        else:

            if chooseOperator == "+":

                result = value1 + value2

            elif chooseOperator == "-":

                result = value1 - value2

            elif chooseOperator == "\*":

                result = value1 \* value2

            elif chooseOperator == "/":

                if value1 == 0 or value2 == 0:

                    print("You can not divide by zero")

                else:

                    result = value1 / value2

            elif chooseOperator == "^":

                result = value1\*\*value2

            elif chooseOperator == "√":

                print("Alert! the square root working only with first number!")

                result = math.sqrt(value1)

            elif chooseOperator == "%":

                result = (value1 / value2) \* 100

            changeDecimals = input(

                f"Enter a number of decimals if you want to change default parameter: "

            )

            if changeDecimals:

                decimals = int(changeDecimals)

                result = f"{result:.{decimals}f}"

                print(f"Result after changing: {result}")

        saveIntoMemory = input("Do you want to save a result in memory? (Y/N): ")

        if saveIntoMemory.lower() == "y":

            memory.append(result)

            history.append((value1, chooseOperator, value2, result))

            break

    moreCalc = input("Do you want make more calculations? (Y/N): ")

    if moreCalc.lower() != "y":

        print("Disposing...")

        break

saveIntoHistory = input("Do you want to view history of your calculation? (Y/N): ")

if saveIntoHistory.lower() == "y":

    print("History of calculations:")

    for historyItem in history:

        (

            value1,

            chooseOperator,

            value2,

            result,

        ) = historyItem

        print(f"{value1} {chooseOperator} {value2} = {result}")

    if memory is not None:

        print(f"Memory: {memory}")

**Посилання на GitHub репозиторій:** [**https://github.com/Senichkaa/lab1-on-python**](https://github.com/Senichkaa/lab1-on-python)

**Висновок:** Виконавши ці завдання, я перетворив консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг мені вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.