Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 2

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“**Основи побудови об’єктно-орієнтованих додатків на Python** ”

Виконала студентка РІ-31

**Панасюк Яна**

Прийняв: Щербак С.С.

Львів – 2024

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

**Завдання лабораторної 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.

Реалізація програми:

from Data.Lab1.Functions.memory\_functions import memory\_save, memory\_clear, memory\_recall

from Data.Lab2.Classes.Calculator import Calculator

from Data.Lab2.Functions.history\_functions import log\_history, clear\_history, view\_history

from Data.Lab2.Functions.memory\_functions import settings

from Shared.Settings.user\_settings import format\_result, set\_decimal\_places, toggle\_auto\_memory\_clear, \

    toggle\_auto\_memory\_save

def display\_menu():

    print("\n🧮  Welcome to the Friendly Calculator!")

    print("Please select an operation:")

    print("1. Add (+)")

    print("2. Subtract (-)")

    print("3. Multiply (\*)")

    print("4. Divide (/)")

    print("5. Exponentiation (x^y)")

    print("6. Square Root (√x)")

    print("7. Remainder (x % y)")

    print("M+. Save to Memory")

    print("MR. Recall from Memory")

    print("MC. Clear Memory")

    print("H. View History")

    print("CH. Clear History")

    print("S1. Set Decimal Places")

    print("S2. Toggle Auto Memory Save")

    print("S3. Toggle Auto Memory Clear")

    print("Q. Quit")

def handle\_choice(choice, calc):

    if choice in ['1', '2', '3', '4', '5', '7']:

        num1 = float(input("🔢 Enter the first number: "))

        #operator = input("✍️  Enter operator (+, -, \*, /, ^, %): ")

        num2 = float(input("🔢 Enter the second number: "))

        if choice == '1':

            result = calc.add(num1, num2)

            print(f"🔍 Result: {num1} + {num2} = {format\_result(result)}")

            log\_history(f"{num1} + {num2}", format\_result(result))

        elif choice == '2':

            result = calc.subtract(num1, num2)

            print(f"🔍 Result: {num1} - {num2} = {format\_result(result)}")

            log\_history(f"{num1} - {num2}", format\_result(result))

        elif choice == '3':

            result = calc.multiply(num1, num2)

            print(f"🔍 Result: {num1} \* {num2} = {format\_result(result)}")

            log\_history(f"{num1} \* {num2}", format\_result(result))

        elif choice == '4':

            result = calc.divide(num1, num2)

            print(f"🔍 Result: {num1} / {num2} = {format\_result(result)}")

            log\_history(f"{num1} / {num2}", format\_result(result))

        elif choice == '5':

            result = calc.exponentiate(num1, num2)

            print(f"🔍 Result: {num1} ^ {num2} = {format\_result(result)}")

            log\_history(f"{num1} ^ {num2}", format\_result(result))

        elif choice == '7':

            result = calc.remainder(num1, num2)

            print(f"🔍 Result: {num1} % {num2} = {format\_result(result)}")

            log\_history(f"{num1} % {num2}", format\_result(result))

        if settings["auto\_memory\_save"]:

            memory\_save(format\_result(result))

    elif choice == '6':

        num = float(input("🔢 Enter number for square root (√): "))

        result = calc.sqrt(num)

        print(f"🔍 Result: √{num} = {format\_result(result)}")

        log\_history(f"√{num}", format\_result(result))

        if settings["auto\_memory\_save"]:

            memory\_save(format\_result(result))

    elif choice == 'M+':

        memory\_save(calc.get\_last\_result())

    elif choice == 'MR':

        memory\_recall()

    elif choice == 'MC':

        memory\_clear()

    elif choice == 'H':

        view\_history()

    elif choice == 'CH':

        clear\_history()

    elif choice == 'S1':

        set\_decimal\_places()

    elif choice == 'S2':

        toggle\_auto\_memory\_save()

    elif choice == 'S3':

        toggle\_auto\_memory\_clear()

    elif choice == 'Q':

        print("👋 Exiting the calculator. Goodbye!")

        return False

    else:

        print("❌ Invalid input. Please enter a valid choice.")

    return True

def run\_calculator():

    calc = Calculator()

    continue\_calculation = True

    while continue\_calculation:

        display\_menu()

        choice = input("🔽 Enter your choice: ")

        continue\_calculation = handle\_choice(choice, calc)

**Висновок**

Виконавши ці завдання, я перетворила 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.

GitHub: https://github.com/yanapanasiuk2005/SMP/tree/master/Data/Lab1