Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 6

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“ **Розробка та Unit тестування Python додатку**”

Виконала студентка РІ-31

**Панасюк Яна**

Прийняв: Щербак С.С.

Львів – 2024

**Мета роботи:** Cтворення юніт-тестів для додатка-калькулятора на основі класів

**Завдання лабораторної роботи**

Завдання 1: Тестування Додавання

Напишіть юніт-тест, щоб перевірити, що операція додавання в вашому додатку-калькуляторі працює правильно. Надайте тестові випадки як для позитивних, так і для негативних чисел.

Завдання 2: Тестування Віднімання

Створіть юніт-тести для переконання, що операція віднімання працює правильно. Тестуйте різні сценарії, включаючи випадки з від'ємними результатами.

Завдання 3: Тестування Множення

Напишіть юніт-тести, щоб перевірити правильність операції множення в вашому калькуляторі. Включіть випадки з нулем, позитивними та від'ємними числами.

Завдання 4: Тестування Ділення

Розробіть юніт-тести для підтвердження точності операції ділення. Тести повинні охоплювати ситуації, пов'язані з діленням на нуль та різними числовими значеннями.

Завдання 5: Тестування Обробки Помилок

Створіть юніт-тести, щоб перевірити, як ваш додаток-калькулятор обробляє помилки. Включіть тести для ділення на нуль та інших потенційних сценаріїв помилок. Переконайтеся, що додаток відображає відповідні повідомлення про помилки.

Файл запуску:

from Data.Lab1.Classes.Calculator import Calculator

from Data.Lab1.Functions.history\_functions import log\_history, view\_history, clear\_history

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

from Shared.Settings.user\_settingsLab1 import format\_result, settings, set\_decimal\_places, toggle\_auto\_memory\_save, \

    toggle\_auto\_memory\_clear

def calculator():

    calc = Calculator()

    print("Select 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+. Memory Save")

    print("MR. Memory Recall")

    print("MC. Memory Clear")

    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")

    while True:

        choice = input("Enter choice (1/2/3/4/5/6/7/M+/MR/MC/H/CH/S1/S2/S3): ")

        if choice in ['1', '2', '3', '4', '5', '6', '7']:

            if choice == '6':

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

                result = calc.sqrt(num)

                formatted\_result = format\_result(result)

                print(f"√{num} = {formatted\_result}")

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

            else:

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

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

                if choice == '1':

                    result = calc.add(num1, num2)

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

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

                elif choice == '2':

                    result = calc.subtract(num1, num2)

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

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

                elif choice == '3':

                    result = calc.multiply(num1, num2)

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

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

                elif choice == '4':

                    result = calc.divide(num1, num2)

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

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

                elif choice == '5':

                    result = calc.exponentiate(num1, num2)

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

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

                elif choice == '7':

                    result = calc.remainder(num1, num2)

                    print(f"{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 == 'M+':

            if 'result' in locals():

                memory\_save(format\_result(result))

            else:

                print("No result to save.")

        elif choice == 'MR':

            recalled\_value = memory\_recall()

            if recalled\_value is not None:

                print(f"Recalled value: {recalled\_value}")

        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()

        else:

            print("Invalid input, please select a valid operation.")

        next\_calculation = input("Do you want to perform another calculation? (yes/no): ")

        if next\_calculation.lower() != 'yes':

            break

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

        memory\_clear()

if \_\_name\_\_ == '\_\_main\_\_':

    calculator()

**Висновок**

Виконавши ці завдання, у мене є набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у мому додатку-калькуляторі. Ці тести допомогають виявити та виправити будь-які проблеми або помилки, які можуть виникнути під час розробки чи обслуговування мого додатку, забезпечуючи його надійність і точність

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