МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ



ЗВІТ

Про виконання лабораторної роботи №6

«Розробка та Unit тестування Python додатку»

З дисципліни

«Спеціалізовані мови програмування»

Студентки групи РІ – 31

Нащецької Яни Вікторівни

Прийняв викладач

Щербак С.С.

**Лабораторна робота № 6. Розробка та Unit тестування Python додатку**

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

**План 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.

**Основний код програми:**

# tests/test\_calculator.py

import os

import sys

lab2\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab2\_root)

main\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", "..", ".."))

sys.path.append(main\_root)

# from src.BLL.main import main

import unittest

from shared.classes.calculator.сalculator\_сonsole import CalculatorConsole

class TestCalculator(unittest.TestCase):

    def setUp(self):

        self.calculatorConsole = CalculatorConsole()

        self.calculator = self.calculatorConsole.calculator

    # Завдання 1: Тестування Додавання

    def test\_add(self):

        self.assertEqual(self.calculatorConsole.calculator.calc(5, "+", 3), 8)

        self.assertEqual(self.calculatorConsole.calculator.calc(-2, "+", -3), -5)

        self.assertEqual(self.calculatorConsole.calculator.calc(0, "+", 0), 0)

    # Завдання 2: Тестування Віднімання

    def test\_subtract(self):

        self.assertEqual(self.calculatorConsole.calculator.calc(10, "-", 5), 5)

        self.assertEqual(self.calculatorConsole.calculator.calc(-5, "-", -3), -2)

        self.assertEqual(self.calculatorConsole.calculator.calc(5, "-", 10), -5)

    # Завдання 3: Тестування Множення

    def test\_multiply(self):

        self.assertEqual(self.calculatorConsole.calculator.calc(4, "\*", 3), 12)

        self.assertEqual(self.calculatorConsole.calculator.calc(-2, "\*", 3), -6)

        self.assertEqual(self.calculatorConsole.calculator.calc(0, "\*", 5), 0)

    # Завдання 4: Тестування Ділення

    def test\_divide(self):

        self.assertEqual(self.calculatorConsole.calculator.calc(10, "/", 2), 5)

        self.assertEqual(self.calculatorConsole.calculator.calc(-6, "/", 3), -2)

        self.assertAlmostEqual(

            self.calculatorConsole.calculator.calc(1, "/", 3), 0.333333, places=5

        )

        with self.assertRaises(ValueError):

            self.calculatorConsole.calculator.calc(10, "/", 0)

    # Завдання 5: Тестування Обробки Помилок

    def test\_error\_handling(self):

        with self.assertRaises(ValueError):

            self.calculatorConsole.calculator.calc(10, "/", 0)(

                10, 0

            )

if \_\_name\_\_ == "\_\_main\_\_":

    unittest.main()

**Висновок:** Виконавши ці завдання, я написала набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у моєму додатку-калькуляторі. Ці тести допомогли виявити та виправити будь-які проблеми або помилки, які виникнули під час розробки додатку, забезпечуючи його надійність і точність.