Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра інформаційних систем та мереж

******

**ЗВІТ**

**ДО ЛАБОРАТОРНОЇ РОБОТИ №6  
з дисципліни  
«Спеціалізовані мови програмування»  
на тему:**  
«Розробка та Unit тестування Python додатку»

***Виконала:*** *студентка групи РІ-31  
 Діана ВІЙЧУК*

***Прийняв:****к. т. н., доцент* *Сергій ЩЕРБАК*

## МЕТА РОБОТИ

## Cтворення юніт-тестів для додатка-калькулятора на основі класів.

## ЗАВДАННЯ

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.

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Файл */lab2/src/tests/tests.py* .

import unittest

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.classes.Calculator import Calculator

class TestCalculator(unittest.TestCase):

def setUp(self):

self.calculator = Calculator()

def test\_add(self):

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

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

self.assertEqual(self.calculator.calculate(0, "+", 0), 0)

def test\_subtract(self):

self.assertEqual(self.calculator.calculate(10, "-", 5), 5)

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

self.assertEqual(self.calculator.calculate(5, "-", 10), -5)

def test\_multiply(self):

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

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

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

def test\_divide(self):

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

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

self.assertAlmostEqual(

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

)

with self.assertRaises(ZeroDivisionError):

self.calculator.calculate(10, "/", 0)

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

unittest.main()

## ВИСНОВКИ

На цій лабораторній роботі я створила Unitt тести для додатка калькулятора на основі класів.

Посилання на GitHub: <https://github.com/Dinasi4ka/Calculator.git>