МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №6

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконала студентка

групи ІТ-21сп

**Романишин.Д.О.**

Прийняв

**Щербак С.С.**

Львів - 2023

**Мета:** створення 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.

Код виконаних завдань представлено нижче.

**\_\_init\_\_.py**

from .main import execute\_lab6

**Calculator.py**

class Calculator:

def add(self, x, y):

return x + y

def subtract(self, x, y):

return x - y

def multiply(self, x, y):

return x \* y

def divide(self, x, y):

if y == 0:

raise ValueError("Cannot divide by zero")

return x / y

**Main.py**

import unittest

def execute\_lab6():

test\_suite = unittest.TestLoader().loadTestsFromName('test\_calc.TestCalculator')

test\_runner = unittest.TextTestRunner(verbosity=2)

test\_runner.run(test\_suite)

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

execute\_lab6()

**Test\_calc.py**

import unittest

from Lab6.calculator import Calculator

class TestCalculator(unittest.TestCase):

def setUp(self):

self.calculator = Calculator()

def test\_addition(self):

self.assertEqual(self.calculator.add(2, 3), 5)

self.assertEqual(self.calculator.add(-2, 3), 1)

self.assertEqual(self.calculator.add(-2, -3), -5)

def test\_subtraction(self):

self.assertEqual(self.calculator.subtract(5, 3), 2)

self.assertEqual(self.calculator.subtract(-2, 3), -5)

self.assertEqual(self.calculator.subtract(-2, -3), 1)

def test\_multiplication(self):

self.assertEqual(self.calculator.multiply(2, 3), 6)

self.assertEqual(self.calculator.multiply(-2, 3), -6)

self.assertEqual(self.calculator.multiply(-2, -3), 6)

self.assertEqual(self.calculator.multiply(0, 3), 0)

def test\_division(self):

self.assertEqual(self.calculator.divide(6, 3), 2)

self.assertEqual(self.calculator.divide(-6, 3), -2)

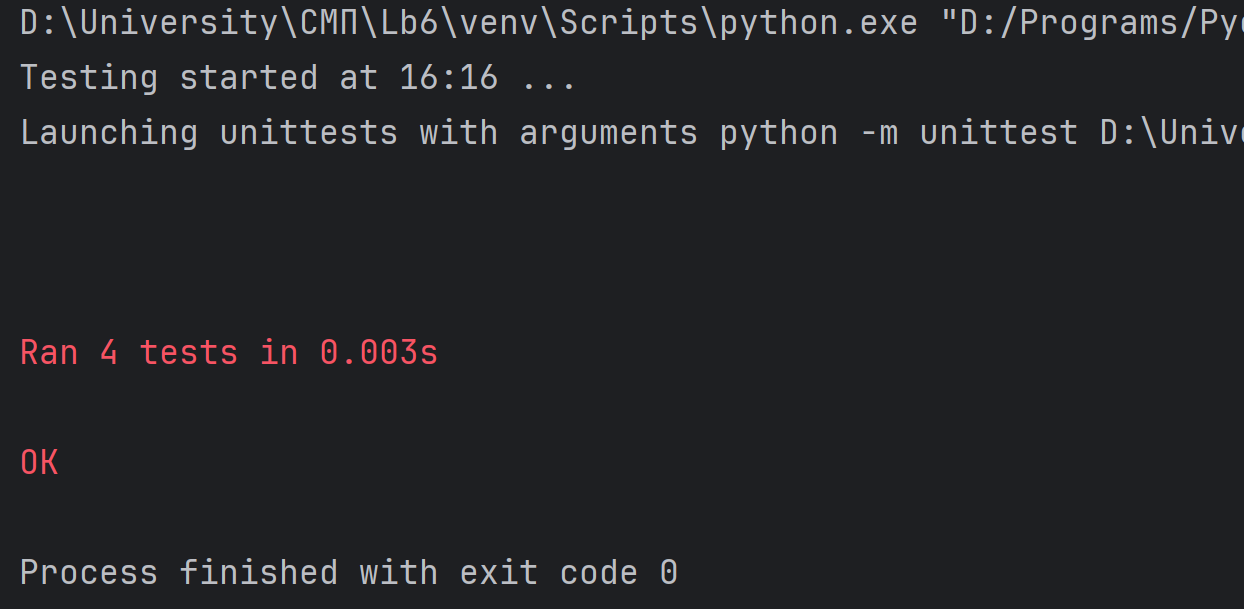
self.assertEqual(self.calculator.divide(6, -3), -2)

self.assertEqual(self.calculator.divide(-6, -3), 2)

with self.assertRaises(ValueError):

self.calculator.divide(5, 0)

**Результат виконання програми**



**Висновок:** під час виконання лабораторної роботи я навчився виявляти та виправляти будь-які проблеми або помилки, які можуть виникнути під час розробки чи обслуговування додатку, забезпечуючи його надійність і точність.