Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Звіт про виконання лабораторної роботи №6

з дисципліни «Спеціалізовані мови програмування»

на тему «Розробка та Unit тестування Python додатку»

Виконала:  
студентка групи РІ-31  
Заплетнюк Ю.І.

Прийняв:  
Щербак С.С.

**Мета роботи:** с творення 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 програмної реалізації**

**test.py**

from io import StringIO

import sys

import os

sys.path.append(os.path.join(os.path.dirname(os.path.abspath(\_\_file\_\_)), '../..'))

import unittest

from lab2.classes.calculator import Calculator

class TestCalculator(unittest.TestCase):

def setUp(self):

self.calc = Calculator()

def test\_addition(self):

self.assertEqual(self.calc.calculate(-1, 4, '+'), 3)

def test\_subtraction(self):

self.assertEqual(self.calc.calculate(5, -3, '-'), 8)

self.assertEqual(self.calc.calculate(3, 5, '-'), -2)

def test\_multiplication(self):

self.assertEqual(self.calc.calculate(0, 5, '\*'), 0)

self.assertEqual(self.calc.calculate(-3, 4, '\*'), -12)

def test\_division(self):

self.assertEqual(self.calc.calculate(10, 2, '/'), 5)

self.assertEqual(self.calc.calculate(-10, 2, '/'), -5)

def test\_error\_handling(self):

with self.assertRaises(ValueError):

self.calc.get\_number('Enter a number: ')

with self.assertRaises(ValueError):

self.calc.calculate(-4, None, '√')

with self.assertRaises(ZeroDivisionError):

self.calc.calculate(10, 0, '/')

unittest.main()

На рис. 1 показано результат виконання завдання.

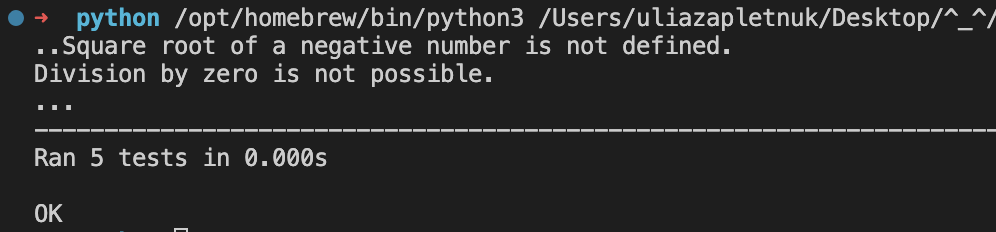


Рис. 1. Результат виконання завдання

**Висновки:** виконавши лабораторну роботу, було створено набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у моєму додатку-калькуляторі. Ці тести допомогли виявити та виправити будь-які проблеми.