МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ

******

Звіт

до лабораторної роботи №6

На тему “Розробка та Unit тестування Python додатку”

З дисципліни “Спеціалізовані мови програмування”

*Виконав:*

*ст. гр. ІТ-31*

*Володимир Лисецький*

*Прийняв:*

*Щербак С. С.*

*Львів - 2023*

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

**Хід роботи**

**Програмний код:**

/calculator/tests/tests

import sys

import unittest

import logging

# Add the parent directory to the sys.path if not already present

sys.path.insert(0, '../lab2')

# Now you can import your Calculator and other classes

from operations.calculator import Calculator, Sum, Subtraction, Multiplication, Division, Power, SquareRoot

# Set up logging configuration

logging.basicConfig(level=logging.INFO)

logger = logging.getLogger(\_\_name\_\_)

class TestCalculator(unittest.TestCase):

def test\_valid\_input(self):

logger.info("Running test\_valid\_input")

calculator = Calculator("+", 2, 3)

self.assertEqual(calculator.operation, "+")

self.assertEqual(calculator.first\_number, 2)

self.assertEqual(calculator.second\_number, 3)

def test\_invalid\_operator(self):

logger.info("Running test\_invalid\_operator")

with self.assertRaises(ValueError):

calculator = Calculator("&", 2, 3)

def test\_invalid\_first\_number(self):

logger.info("Running test\_invalid\_first\_number")

with self.assertRaises(ValueError):

calculator = Calculator("+", "abc", 3)

# Similar modifications for other test classes...

class TestSum(unittest.TestCase):

def test\_sum\_positive\_numbers(self):

logger.info("Running test\_sum\_positive\_numbers")

calculator = Sum(2, 3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 5)

def test\_sum\_negative\_numbers(self):

logger.info("Running test\_sum\_negative\_numbers")

calculator = Sum(-2, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, -5)

def test\_sum\_mixed\_numbers(self):

logger.info("Running test\_sum\_mixed\_numbers")

calculator = Sum(2, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, -1)

class TestSubtraction(unittest.TestCase):

def test\_subtract\_positive\_numbers(self):

logger.info("Running test\_subtract\_positive\_numbers")

calculator = Subtraction(5, 3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 2)

def test\_subtract\_negative\_numbers(self):

logger.info("Running test\_subtract\_negative\_numbers")

calculator = Subtraction(-5, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, -2)

def test\_subtract\_mixed\_numbers(self):

logger.info("Running test\_subtract\_mixed\_numbers")

calculator = Subtraction(2, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 5)

class TestMultiplication(unittest.TestCase):

def test\_multiply\_positive\_numbers(self):

logger.info("Running test\_multiply\_positive\_numbers")

calculator = Multiplication(2, 3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 6)

def test\_multiply\_negative\_numbers(self):

logger.info("Running test\_multiply\_negative\_numbers")

calculator = Multiplication(-2, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 6)

def test\_multiply\_mixed\_numbers(self):

logger.info("Running test\_multiply\_mixed\_numbers")

calculator = Multiplication(2, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, -6)

class TestDivision(unittest.TestCase):

def test\_divide\_positive\_numbers(self):

logger.info("Running test\_divide\_positive\_numbers")

calculator = Division(6, 3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 2)

def test\_divide\_negative\_numbers(self):

logger.info("Running test\_divide\_negative\_numbers")

calculator = Division(-6, -3)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, 2)

def test\_divide\_mixed\_numbers(self):

logger.info("Running test\_divide\_mixed\_numbers")

calculator = Division(8, -4)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertEqual(result, -2)

def test\_divide\_by\_zero(self):

logger.info("Running test\_divide\_by\_zero")

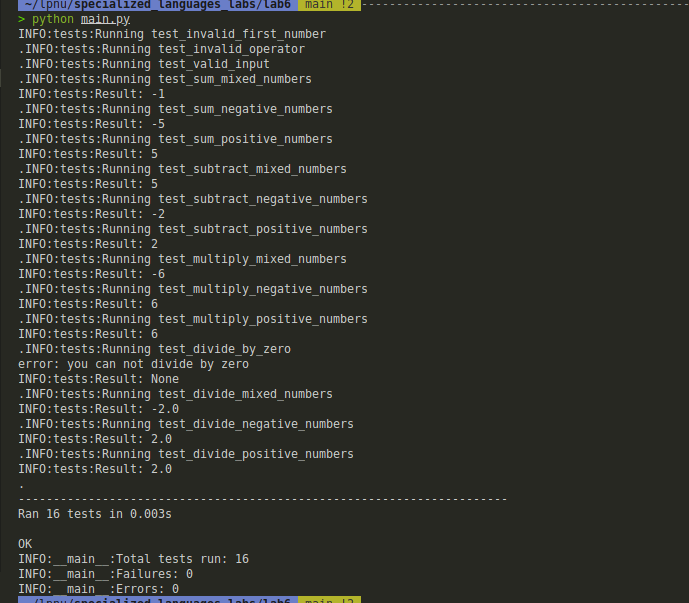
calculator = Division(5, 0)

result = calculator.calculate()

logger.info(f"Result: {result}")

self.assertIsNone(result)

Результат виконання програми:



**Висновок:** Під час виконання даної лабораторної роботи було створено юніт-тести для додатка-калькулятора на основі класів.