

Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Кафедра інформаційних систем та мереж

ЗВІТ
про виконання лабораторної роботи № 6
“Розробка та Unit тестування Python додатку ”
з дисципліни **“ Спеціалізовані мови програмування”**

Виконала студентка групи ІТ-32

ДЕНИСЯК С.-М. Т.

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 Помилки

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 програми:

```
import unittest
from src.lab1.calculator_operations import *
from src.lab1.console_interaction import *

class TestCalculatorOperations(unittest.TestCase):

    def test_addition(self):
        result = add(2, 3)
        self.assertEqual(result, 5)

        result = add(-2, 3)
        self.assertEqual(result, 1)

        result = add(-2, -3)
        self.assertEqual(result, -5)

    def test_subtraction(self):
        result = subtract(5, 3)
        self.assertEqual(result, 2)

        result = subtract(-2, 3)
        self.assertEqual(result, -5)
```

```

        result = subtract(-2, -3)
        self.assertEqual(result, 1)

    def test_multiplication(self):
        result = multiply(2, 3)
        self.assertEqual(result, 6)

        result = multiply(-2, 3)
        self.assertEqual(result, -6)

        result = multiply(-2, -3)
        self.assertEqual(result, 6)

    def test_division(self):
        result = divide(6, 3)
        self.assertEqual(result, 2)

        result = divide(-6, 3)
        self.assertEqual(result, -2)

        with self.assertRaises(ZeroDivisionError):
            divide(6, 0)

    def test_error_handling(self):
        with self.assertRaises(ArithmeticError):
            compute_square_root(-4)

        with self.assertRaises(ArithmeticError):
            change_decimal_places(0)

if __name__ == '__main__':
    unittest.main()

```

Висновки. Виконавши ці завдання, у вас буде набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у вашому додатку-калькуляторі. Ці тести допоможуть виявити та виправити будь-які проблеми або помилки, які можуть виникнути під час розробки чи обслуговування вашого додатку, забезпечуючи його надійність і точність