НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №6

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав:

ст. гр. РІ-31

Назар МРАКА

Прийняв:

Сергій ЩЕРБАК

Львів-2024

**Мета** 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.Код програми:

*class TestCalculator(unittest.TestCase):*

*def setUp(self):*

*self.calculator = Calculator()*

*def test\_addition(self):*

*self.assertEqual(self.calculator.calcOperations(5, 3, '+'), 8)*

*self.assertEqual(self.calculator.calcOperations(-5, -3, '+'), -8)*

*self.assertEqual(self.calculator.calcOperations(-5, 3, '+'), -2)*

*def test\_subtraction(self):*

*self.assertEqual(self.calculator.calcOperations(10, 3, '-'), 7)*

*self.assertEqual(self.calculator.calcOperations(-5, -3, '-'), -2)*

*self.assertEqual(self.calculator.calcOperations(5, 10, '-'), -5)*

*def test\_multiplication(self):*

*self.assertEqual(self.calculator.calcOperations(5, 3, '\*'), 15)*

*self.assertEqual(self.calculator.calcOperations(5, 0, '\*'), 0)*

*self.assertEqual(self.calculator.calcOperations(-5, -3, '\*'), 15)*

*self.assertEqual(self.calculator.calcOperations(5, -3, '\*'), -15)*

*def test\_division(self):*

*self.assertEqual(self.calculator.calcOperations(10, 2, '/'), 5)*

*self.assertEqual(self.calculator.calcOperations(-10, -2, '/'), 5)*

*self.assertEqual(self.calculator.calcOperations(-10, 2, '/'), -5)*

*with self.assertRaises(ZeroDivisionError):*

*self.calculator.calcOperations(10, 0, '/')*

*def test\_error\_handling(self):*

*error = errorChecker(10, '#')*

*self.assertEqual(error, "Invalid operator")*

*error = errorChecker(0, '/')*

*self.assertEqual(error, "Division by zero")*

**Висновок:**

У рамках лабораторної роботи було розроблено юніт-тести для додатка-калькулятора на основі класів, що дозволяє перевіряти коректність роботи калькулятора, реалізованого в об'єктно-орієнтованому стилі на мові Python. Метою цієї роботи було створення тестів для автоматичного перевіряння функціональності основних операцій калькулятора, таких як додавання, віднімання, множення та ділення.