Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №6

з дисципліни «Спеціалізовані мови програмування»

на тему

«Розробка та Unit тестування Python додатку»

Виконав:

Костюк В. О.

Перевірив:

Щербак С. С.

Львів 2024

**Мета:** Cтворення 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.

**Реалізація:**

**Файл calculator.py:**

class Calculator:

    def addition(self, x, y):

        return x + y

    def subtraction(self, x, y):

        return x - y

    def multiplication(self, x, y):

        return x \* y

    def division(self, x, y):

        if y == 0:

            raise ValueError("Cannot divide by zero.")

        return x / y

def main():

    calculator = Calculator()

    print("Ласкаво просимо до калькулятора!")

    print("Доступні операції:")

    print("1. Додавання (+)")

    print("2. Віднімання (-)")

    print("3. Множення (\*)")

    print("4. Ділення (/)")

    while True:

        operation = input("Введіть операцію (або 'exit' для виходу): ").strip()

        if operation == 'exit':

            print("Дякуємо за використання калькулятора!")

            break

        if operation in ['+', '-', '\*', '/']:

            try:

                x = float(input("Введіть перше число: "))

                y = float(input("Введіть друге число: "))

                if operation == '+':

                    result = calculator.addition(x, y)

                elif operation == '-':

                    result = calculator.subtraction(x, y)

                elif operation == '\*':

                    result = calculator.multiplication(x, y)

                elif operation == '/':

                    result = calculator.division(x, y)

                print(f"Результат: {result}")

            except ValueError as e:

                print(f"Помилка: {e}")

        else:

            print("Невідома операція. Спробуйте ще раз.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Файл test\_add.py:**

import unittest

from lab6.calculator import Calculator

class TestAddition(unittest.TestCase):

    def setUp(self):

        self.calculator = Calculator()

    def test\_add(self):

        self.assertEqual(self.calculator.addition(1, 2), 3)

        self.assertEqual(self.calculator.addition(-1, -1), -2)

        self.assertEqual(self.calculator.addition(-1, 1), 0)

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

    unittest.main()

**Файл test\_divide.py:**

import unittest

from lab6.calculator import Calculator

class TestDivision(unittest.TestCase):

    def setUp(self):

        self.calculator = Calculator()

    def test\_divide(self):

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

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

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

        self.assertEqual(self.calculator.division(0, 1), 0)

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

    unittest.main()

**Файл test\_multiply.py:**

import unittest

from lab6.calculator import Calculator

class TestMultiplication(unittest.TestCase):

    def setUp(self):

        self.calculator = Calculator()

    def test\_multiply(self):

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

        self.assertEqual(self.calculator.multiplication(-1, -1), 1)

        self.assertEqual(self.calculator.multiplication(-1, 1), -1)

        self.assertEqual(self.calculator.multiplication(0, 5), 0)

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

    unittest.main()

**Файл test\_subtract.py:**

import unittest

from lab6.calculator import Calculator

class TestSubtraction(unittest.TestCase):

    def setUp(self):

        self.calculator = Calculator()

    def test\_subtract(self):

        self.assertEqual(self.calculator.subtraction(2, 1), 1)

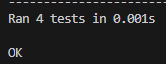
        self.assertEqual(self.calculator.subtraction(-1, -1), 0)

        self.assertEqual(self.calculator.subtraction(1, 2), -1)

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

    unittest.main()

**Результат виконання (test\_calculator.py):**



**Висновок:** на цій лабораторній роботі я додав набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у 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