Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ІСМ



Звіт

про виконання лабораторної роботи № 6

«Розробка ASCII ART генератора для візуалізації 2D-фігур»

з дисципліни

«Розробка та Unit тестування Python додатку»

Виконав:

Студент групи ІТ-32,

Мельничук І.Т

Прийняв:

Щербак С.С

Львів

2023

**Мета роботи:** 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.**

**Результат виконання:**

**Calculator.py**

class Calculator:

    @staticmethod

    def add(num1, num2):

        return num1 + num2

    @staticmethod

    def subtract(num1, num2):

        return num1 - num2

    @staticmethod

    def multiply(num1, num2):

        return num1 \* num2

    @staticmethod

    def divide(num1, num2):

        if num2 == 0:

            raise ValueError("Cannot divide by zero")

        return num1 / num2

**data.py**

defaultroundto = 3

results = []

memory = []

**main.py**

# main.py

from test import TestCalculator

import unittest

def main():

   suite = unittest.TestLoader().loadTestsFromTestCase(TestCalculator)

   runner = unittest.TextTestRunner()

   runner.run(suite)

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

    main()

**test.py**

import unittest

from unittest.mock import MagicMock

from calculator import Calculator

class TestCalculator(unittest.TestCase):

    def test\_input(self):

        self.assertIsInstance(float(5), float)

        self.assertNotIsInstance(int(5), float)  # Ensure int(5) is not an instance of float

    # Завдання 1: Тестування Додавання

    def test\_add(self):

        self.assertEqual(Calculator.add(1, 2), 3)

        self.assertEqual(Calculator.add(-1, -2), -3)

        self.assertEqual(Calculator.add(-1, 1), 0)

    # Завдання 2: Тестування Віднімання

    def test\_subtract(self):

        self.assertEqual(Calculator.subtract(5, 3), 2)

        self.assertEqual(Calculator.subtract(-1, -2), 1)

        self.assertEqual(Calculator.subtract(-1, 1), -2)

        self.assertEqual(Calculator.subtract(1, -1), 2)

    # Завдання 3: Тестування Множення

    def test\_multiply(self):

        self.assertEqual(Calculator.multiply(3, 2), 6)

        self.assertEqual(Calculator.multiply(-1, -2), 2)

        self.assertEqual(Calculator.multiply(-1, 2), -2)

        self.assertEqual(Calculator.multiply(0, 100), 0)

    # Завдання 4: Тестування Ділення

    def test\_divide(self):

        self.assertEqual(Calculator.divide(6, 2), 3)

        self.assertEqual(Calculator.divide(-4, -2), 2)

        self.assertEqual(Calculator.divide(-4, 2), -2)

        with self.assertRaises(ValueError):

            Calculator.divide(5, 0)

    # Завдання 5: Тестування Обробки Помилок

    def test\_divide\_errors(self):

        with self.assertRaises(ValueError):

            Calculator.divide(10, 0)

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

    unittest.main()

**user.py**

from data import defaultroundto

from data import memory

class User:

    def \_\_init\_\_(self,name):

        self.name = name

        self.roundto = defaultroundto

        self.memory = memory

**Висновки: Виконавши ці завдання, я створив 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 додатку, забезпечуючи його надійність і точність.**