МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №6

з дисципліни “Спеціалізовані мови програмування”

Виконав:

студент групи ІТ-31

Олександр ГАМАЮНОВ

Прийняв:

доцент кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

**Тема роботи:** Розробка та Unit тестування Python додатку

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

Хід 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.

Код:

Main.py

""" Main file"""

import unittest

from .classes.test import TestCalculator

def main():

    """ Main method"""

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

    runner = unittest.TextTestRunner()

    runner.run(suite)

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

    main()

Test.py:

""" Test"""

import unittest

from Shared.calculator import Calculator

class TestCalculator(unittest.TestCase):

    """ Test for unittesting calculator"""

    def test\_add(self):

        """Add test """

        calculator = Calculator()

        num1 = 2.

        num2 = 2.

        result = calculator.test\_plus(num1,num2)

        self.assertEqual(result, 4)

    def test\_multiply(self):

        """ Mulitply test"""

        calculator = Calculator()

        num1 = 2.

        num2 = 2.

        result = calculator.test\_multiply(num1,num2)

        self.assertEqual(result, 4)

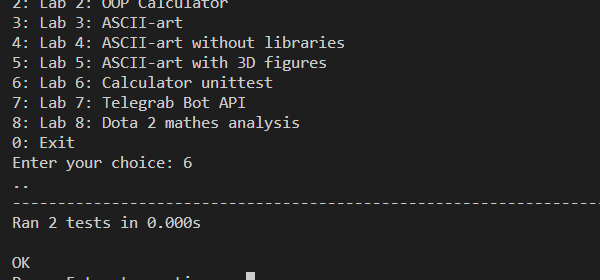


Рис.1 Результат виконання роботи

**Висновок**: Виконуючи лабораторну роботу було створено набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у калькуляторі.