

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра САПР



ЗВІТ

Лабораторна робота № 3

на тему: “Модульне тестування ”
з дисципліни: “Управління ІТ-проектами”

Виконав студент групи КН-406

Назарій ШКВАРЛА

Прийняв

Іван ШПАКЕВИЧ

Львів 2023

Мета роботи: ознайомитись з принципами модульного тестування проєктів. Набути практичних навичок роботи з Junit.

Лабораторне завдання:

1. Ознайомитись з принципами модульного тестування.
2. Організувати модульне тестування проєкту зробленого у попередній лабораторній роботі, інтегрувати процес модульного тестування до автоматизованої збірки.

Хід виконання роботи

Для модульного тестування на мові програмування Python, створив новий проєкт в який імпортував бібліотеку для модульного тестування, а саме unittest, та функцію з своєї лабораторної роботи, яку буду тестувати. Далі використовуючи документацію створюю клас для тестування, який наслідуватиме клас для тестування з бібліотеки unittest, та створюю тести, для точної перевірки роблю один тест правильним, інший ні.

```
from lab3 import transport_min_cost
import unittest

import numpy as np

class TestClass(unittest.TestCase):
    def test_correct(self):
        # Ввід початкових даних
        supply = np.array([10, 20, 30]) # Постачальники
        demand = np.array([16, 12, 11, 10]) # Споживачі
        costs = np.array([[17, 16, 15, 14],
                          [12, 13, 11, 12],
                          [15, 18, 19, 17]]) # Вартість перевезення
        routes = transport_min_cost(supply, demand, costs)
        # Вартість доставки
        total = np.sum(routes * costs)
        self.assertEqual(690, total)

    def test_failed(self):
        # Ввід початкових даних
        supply = np.array([10, 20, 30]) # Постачальники
        demand = np.array([16, 12, 11, 10]) # Споживачі
        costs = np.array([[17, 16, 15, 14],
                          [12, 13, 11, 12],
                          [15, 18, 19, 17]]) # Вартість перевезення
        routes = transport_min_cost(supply, demand, costs)
        # Вартість доставки
        total = np.sum(routes * costs)
        self.assertEqual(60, total)

if __name__ == "__main__":
    unittest.main()
```

Рис. 1 Підключення бібліотеки та тести

Далі запускаю проєкт змінивши конфігурацію запуску для кращого виводу інформації про тести.

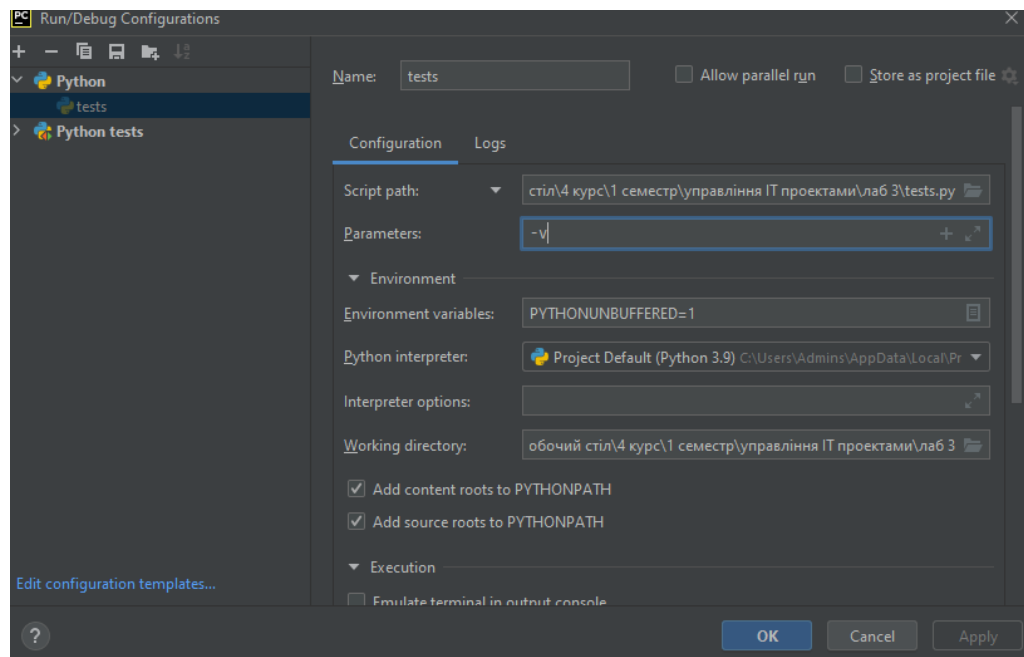


Рис. 2 Зміна конфігурації запуску

Далі запускаю тестування та отримую детальний результат по тестах, а саме які виконали перевірку, які ні, та де він знаходиться.

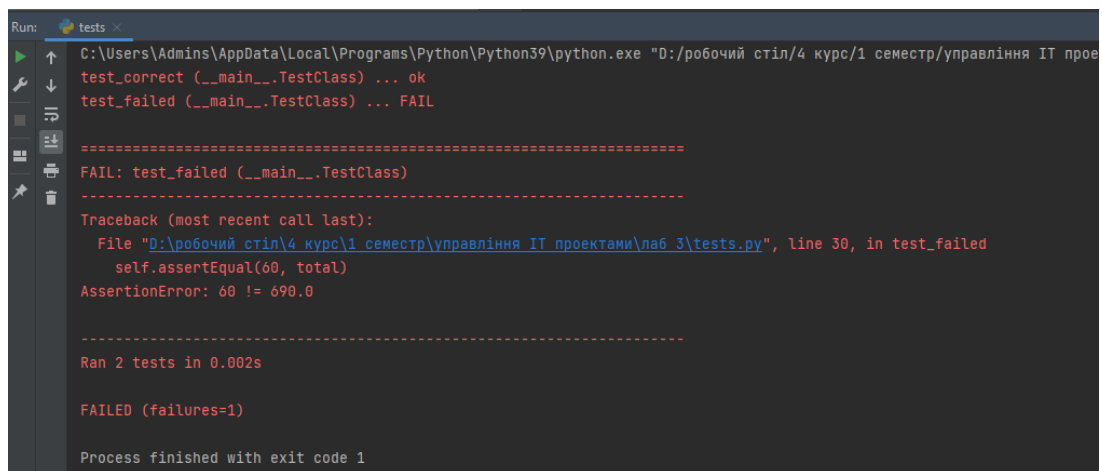


Рис. 3 Результат тестування

Висновок: під час виконання лабораторної роботи я ознайомився з модульним тестуванням. Зрозумів принципи його роботи. Набув навичок модульного тестування на прикладі бібліотеки unittest на мові програмування Python.