**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

**Кафедра САП**



**Звіт**

**до лабораторної роботи №3**

з дисципліни: “Управління ІТ-проектами”

**на тему:**

**“МОДУЛЬНЕ ТЕСТУВАННЯ”**

Виконала студентка групи КН-408  
КИЧУК О. О.

Прийняв асистент

ШПАКОВИЧ І. М.

ЛЬВІВ – 2023

**Мета роботи:** Ознайомитись з принципами модульного тестування проектів. Набути практичних навичок роботи з JUnit.

**Хід роботи:**

Код нашої програми, який використовувався у минулій лабораторній роботі:

package gradle\_lab;

public class App {

    public static void main(String[] args) {

        double number1 = 15;

        double number2 = 3;

        double sum = add(number1, number2);

        double difference = subtract(number1, number2);

        double product = multiply(number1, number2);

        double quotient = divide(number1, number2);

        System.out.println("Summary: " + sum);

        System.out.println("Difference: " + difference);

        System.out.println("Multiply: " + product);

        System.out.println("Division: " + quotient);

    }

    public static double add(double a, double b) {

        return a + b;

    }

    public static double subtract(double a, double b) {

        return a - b;

    }

    public static double multiply(double a, double b) {

        return a \* b;

    }

    public static double divide(double a, double b) {

        if (b == 0) {

            System.out.println("Error: Division by zero!");

            return Double.NaN;

        }

        return a / b;

    }

}

Код Unit тестів:

package gradle\_lab;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

public class AppTest {

    @Test

    public void testAdd() {

        double result = App.add(5.0, 3.0);

        assertEquals(8.0, result, 0.0001); // Перевіряємо, чи вірно додався результат

    }

    @Test

    public void testSubtract() {

        double result = App.subtract(10.0, 4.0);

        assertEquals(6.0, result, 0.0001); // Перевіряємо, чи вірно віднявся результат

    }

    @Test

    public void testMultiply() {

        double result = App.multiply(6.0, 5.0);

        assertEquals(30.0, result, 0.0001); // Перевіряємо, чи вірно перемножився результат

    }

    @Test

    public void testDivide() {

        double result = App.divide(12.0, 4.0);

        assertEquals(3.0, result, 0.0001); // Перевіряємо, чи вірно поділився результат

    }

    @Test

    public void testDivideByZero() {

        double result = App.divide(10.0, 0.0);

        assertTrue(Double.isNaN(result)); // Перевіряємо, чи вірно обробляється ділення на нуль

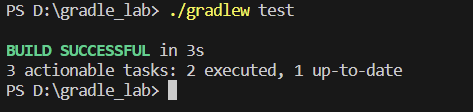
    }

}

**Результати:**

Загалом всі тести у моєму випадку однакові за принципом їхньої роботи, тому ми розглянемо тільки декілька з них.

Спершу перевіримо, чи виконуються наші тести:



1. Тест testSubtract і testMultiply*:*

Змінюємо в коді очікуванні результати, щоб перевірити роботу тестів:

 @Test

    public void testSubtract() {

        double result = App.subtract(10.0, 4.0);

        assertEquals(5.0, result, 0.0001);

    }

    @Test

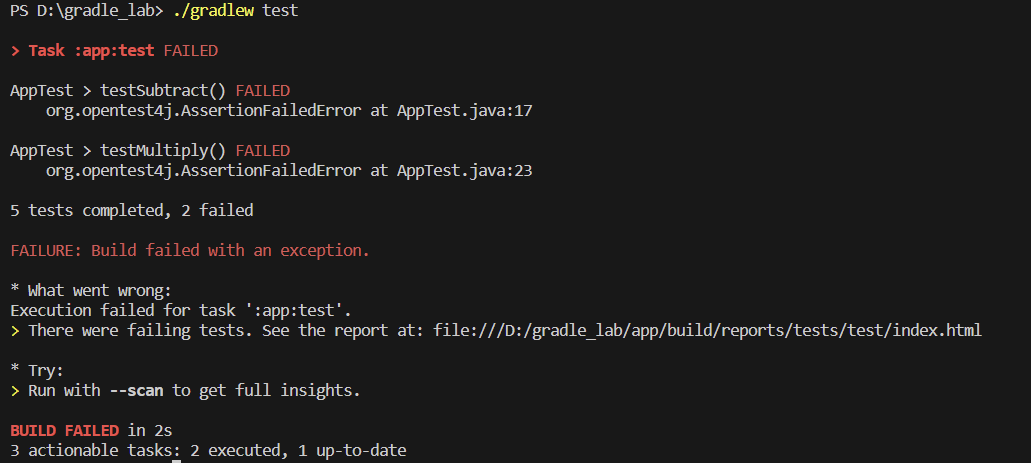
    public void testMultiply() {

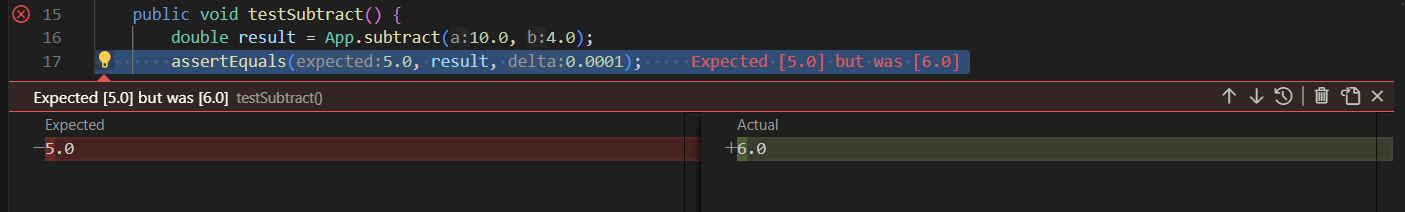
        double result = App.multiply(6.0, 5.0);

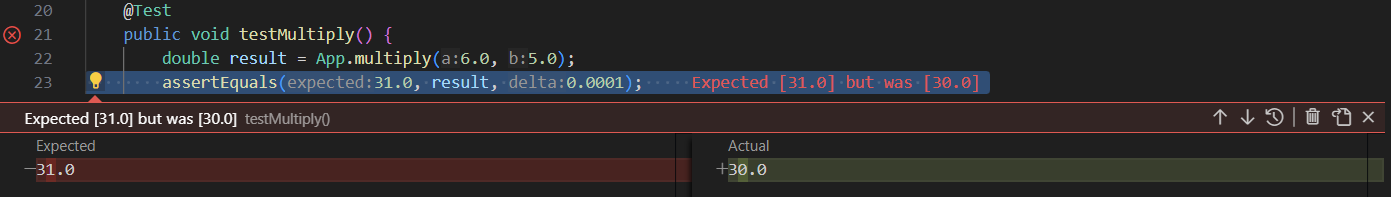
        assertEquals(31.0, result, 0.0001);

    }

Запускаємо тест:







**Висновки:** Ознайомилася з принципами модульного тестування проектів. Набула практичних навичок роботи з JUnit.