Міністерство освіти і науки України

Національний університет “Львівська політехніка”

**МОДУЛЬНЕ ТЕСТУВАННЯ**

**Звіт**

до лабораторної роботи № 3

**Автоматизація збірки проекту**

Виконав

студент гр. КН-307

Дзиба М.Т.

Перевірив викладач:

Денисюк П.Ю.

Львів – 2021

**Мета роботи** – ознайомитись з принципами модульного тестування проектів.

**1. ТЕОРЕТИЧНІ ВІДОМОСТІ**

**1.1. ОСНОВНИ МОДУЛЬНОГО ТЕСТУВАННЯ**

Модульне тестування, або юніт-тестування (англ. unit testing) - процес у програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми. Це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яку може бути протестованою. У процедурному програмуванні модулем вважають окрему функцію або процедуру. В об'єктно-орієнтованому програмуванні — інтерфейс, клас. Модульні тести, або unit-тести, розробляються в процесі розробки програмістами та, іноді, тестувальниками білої скриньки (white-box testers). Зазвичай Unit-тести застосовують для того, щоб упевнитися, що код відповідає вимогам архітектури та має очікувану поведінку.

2.2. ОСНОВИ JUNIT

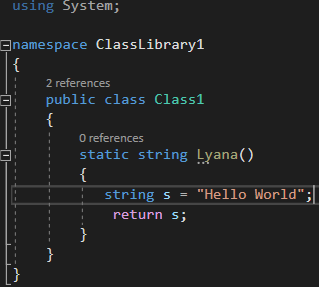
Сьогодні все більшої популярності набуває test-driven development (TDD), техніка розробки ПЗ, при якій спочатку пишеться тест на певний функціонал, а потім пишеться реалізація цього функціоналу. На практиці все, звичайно ж, не настільки ідеально, але в результаті код не тільки написаний і протестований, але тести як би неявно задають вимоги до функціонала, а також показують приклад використання цього функціоналу. Найвідоміший, а тому й самий використовуваний фреймворк для тестування - JUnit. Використовується він у двох варіантах JUnit 3 і JUnit 4 (http://junit.org/). Розглянемо обидві версії, так як в старих проектах досі використовується 3-а, яка підтримує Java 1.4.

2.2.1. JUNIT 3

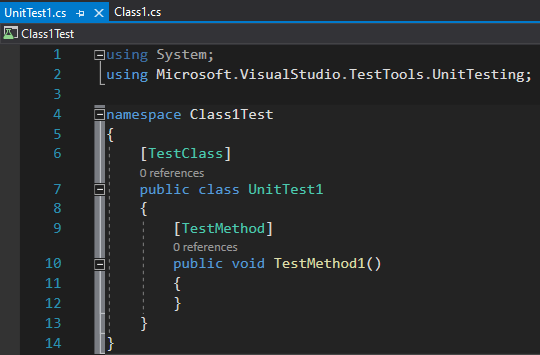
Для створення тесту потрібно успадкувати тест- клас від TestCase , перевизначити методи setUp і tearDown якщо треба , ну і найголовніше - створити тестові методи (повинні починатися з test) . При запуску тесту спочатку створюється екземпляр тест-класу (для кожного тесту в класі окремий екземпляр класу), потім виконується метод setUp, запускається сам тест, ну і на завершення виконується метод tearDown. Якщо який-небудь з методів викидає виключення, тест вважається проваленим. Примітка: тестові методи повинні бути public void, можуть бути static. Самі тести складаються з виконання деякого коду і перевірок. Перевірки найчастіше виконуються за допомогою класу Assert хоча іноді використовують ключове слово assert.

**Практичне завдання**

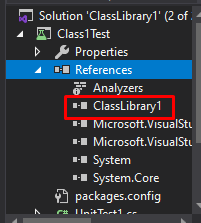
1. Взяв код з попередньої лабораторної роботи. Створив метод з виконанням дії



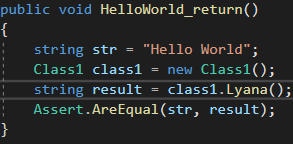
1. Добавив Unit-тест для цього методу



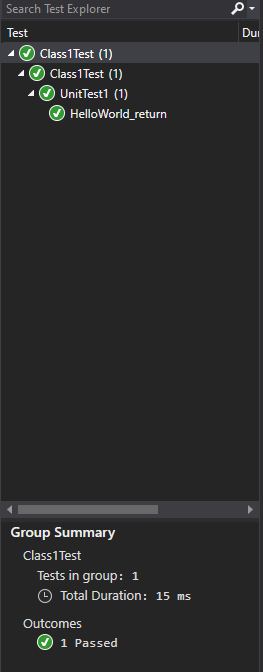
1. Добавив посилання на збірку проекту для Unit-тесту



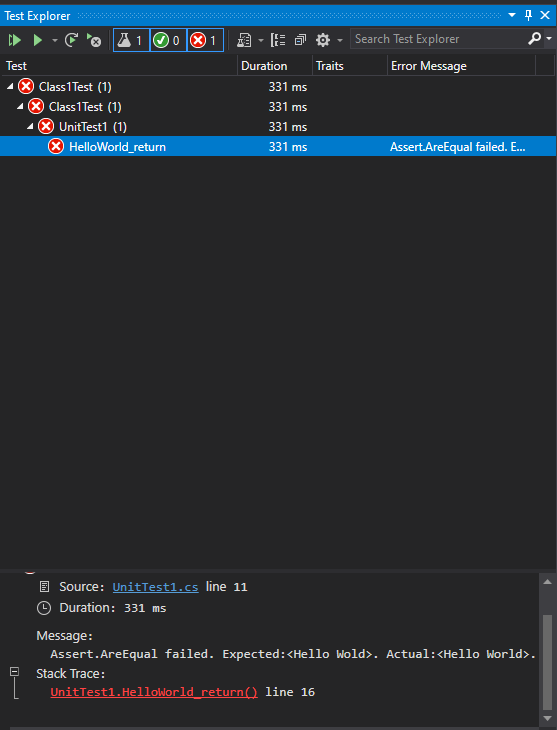
1. Змінимо код Unit-тесту наступним чином:



1. Після збірки та виконання програми, отримуємо наступне вікно:



Це означає, що Unit-тест пройшов успішно. У випадку неуспішності, буде наступне вікно:



**Висновок.** ознайомився з принципами модульного тестування проектів.