**Thuộc Môn Học: Kiểm Thử Phần Mềm**

**BÀI TẬP LỚN 01(LÀM NHÓM):**

**UNIT TESTING**

Thông Tin Sinh Viên

MSSV và Họ tên SV1: ……………………………………………………….

MSSV và Họ tên SV2: ……………………………………………………….

MSSV và Họ tên SV3: ……………………………………………………….

MSSV và Họ tên SV4: ……………………………………………………….

**Nội dung:** Báo cáo bài tập lớn. Xem đề bài ở trang bên dưới.

**Hình thức:** Bài tập lớn, làm nhóm, nhóm tối đa 4 người.

**Hướng dẫn:** Các bạn viết file báo cáo theo mẫu này và đính kèm các tài nguyên liên quan như code,database… (nếu có) . Nén lại tất cả thành một file duy nhất để nộp đặt tên là

MSL\_MSSV1\_MSSV2\_MSSV3\_MSSV4\_BaiTapLonUnitTesing.rar

**MÔ TẢ BÀI TẬP LỚN MÔN HỌC:**

Sinh viên thực bài tập lớn môn học theo nhóm (Nhóm theo danh sách đăng ký). Nội dung bài tập lớn là tìm hiểu về công cụ Unit – Testing trong lập trình. Bài tập lớn của môn học phải đáp ứng các điều kiện và tiêu chí sau:

* Ghi nhận kết quả làm được trong mẫu báo cáo này và nộp lại.
* **Lý thuyết về Unit – Testing (3 điểm)**: Khái niệm, vòng đời, thiết kế, ứng dụng và vai trò của Unit – Test. Tìm hiểu bằng bất kỳ nguồn tài liệu nào có trên mạng, đọc hiểu và ghi lại cô động nội dung mình đã hiểu. Bên dưới là link cho nguồn tài liệu tìm hiểu mở đầu:

<https://topdev.vn/blog/unit-test-la-gi/>

* **Demo Unit – Testing (7 điểm)**: Nhóm demo cách sử dụng Unit – testing ở mức cơ bản cho ngôn ngữ mà mình quen thuộc. Lưu ý chỉ làm demo cho một ngôn ngữ mà mình chọn thôi. Bên dưới là 3 hướng dẫn cho 3 ngôn ngữ (Java, C#, Python) đề xuất. Nếu chọn nhóm chỉ làm 1 trong 3 ngôn ngữ này. Ngoài ra nhóm có thể chọn bất kỳ ngôn ngữ khác mà mình thích hoặc nguồn hướng dẫn nào mình thích có trên mạng. Các link hướng dẫn bên dưới chỉ là gợi ý.

Nhóm thực hiện lại các bước cần thực hiện. Chỉ chụp và dán màn hình kết quả vào mỗi bước. Có thể giải thích ngắn ngọn không cần quá chi tiết cho mỗi bước mình làm. Lưu ý source code viết được hay các tài nguyên khác cần thiết để chạy demo cần nén lại thành một tập tin duy nhất và nộp.

Hướng dẫn sử dụng Unit – Testing

* 1. Sử dụng Unit – Testing với Java

Part 01: <https://www.youtube.com/watch?v=0JEkuD7EAtE>

Part 02: <https://www.youtube.com/watch?v=POgl8b7FnrM>

* 1. Unit-Test C#

Part 01: <https://www.youtube.com/watch?v=L-7ux_K809c>

Part 02: <https://www.youtube.com/watch?v=F1W9IaSfOZ8>

* 1. Unit-Test Python:

<https://www.youtube.com/watch?v=6tNS--WetLI>

**NỘI DUNG BÁO CÁO**

1. **Lý thuyết về Unit – Testing (3 điểm)**
   * + - **Mục nội dung 1…**

Nội dung ….

* + - * **Mục nội dung 2…**

Nội dung ….

* + - * **Mục nội dung 3…**

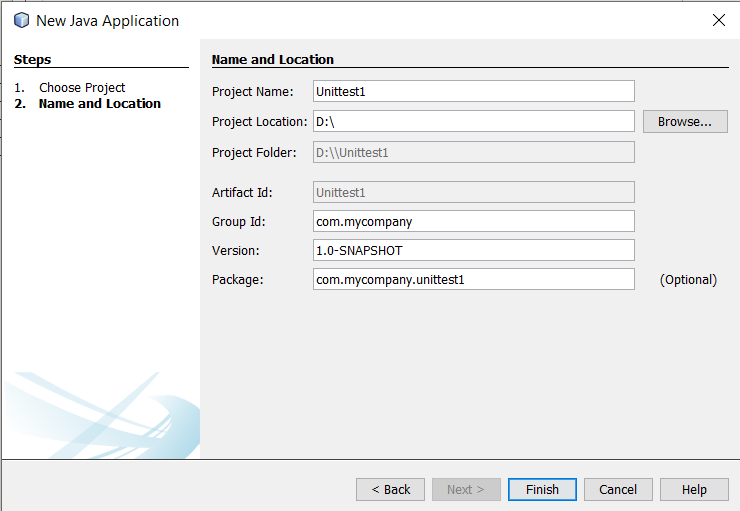
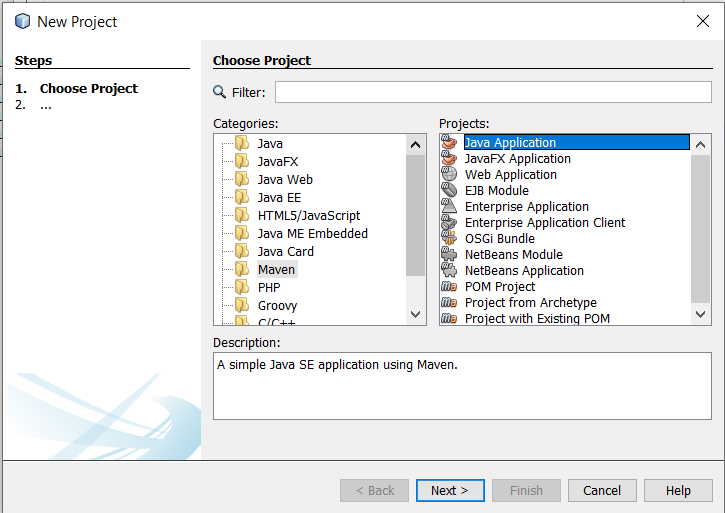
Nội dung ….

* + - * **…**

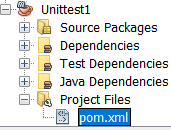
...

1. **Demo về Unit – Testing (7 điểm)**

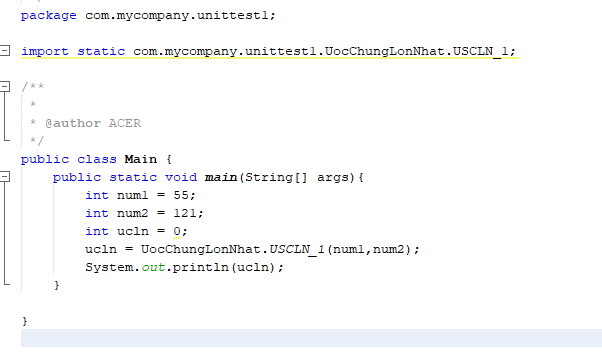
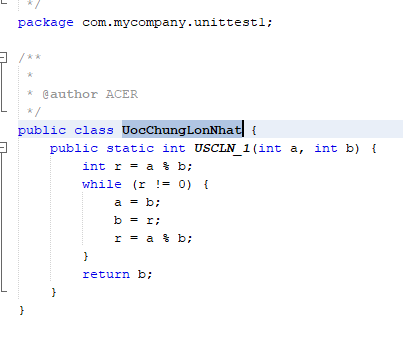
**1/ Tạo 1 hàm Test case căn bản với Junit 5:**

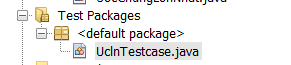
*Tạo 1 Maven Project*

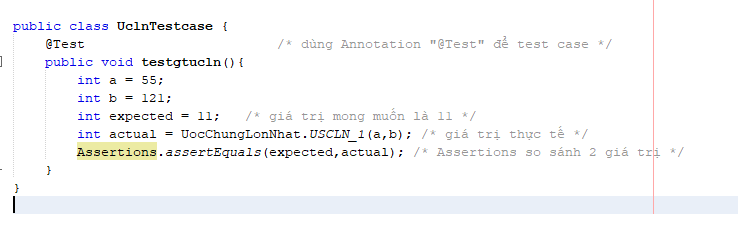
*Thêm đoạn code Framework Junit 5 vào File pom.xml*



*Code tìm ước chung lớn nhất 2 số a = 55 và b = 121, kết quả là 11*



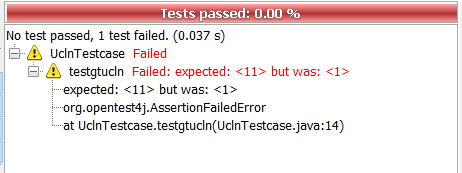
*Tạo 1 File để test*



*Hàm Test case căn bản*



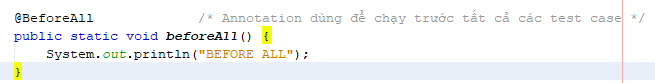
*Test thành công (Gía trị chính xác)*



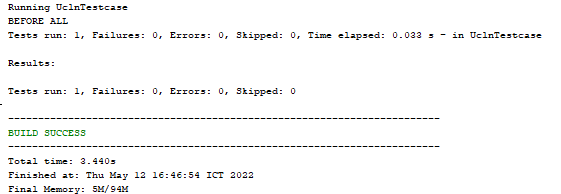
*Test fail*

**1.2/ Before All – After All**

**Before All**

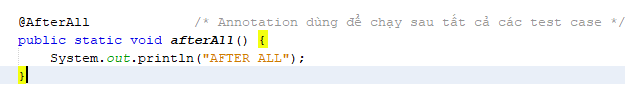


*Hàm thực thi trước khi tất cả test case chạy xong*

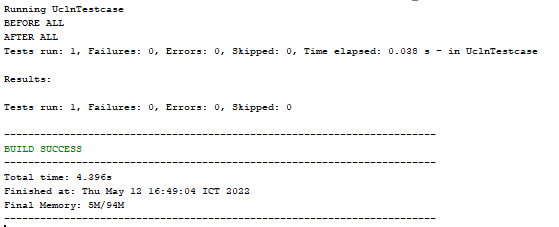


*Kết quả*

**After All**



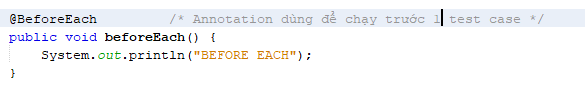
*Hàm thực thi sau khi tất cả test case chạy xong*



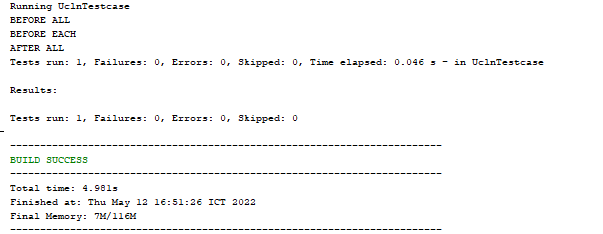
*Kết quả*

**1.3/ Before Each – After Each**

**Before Each**

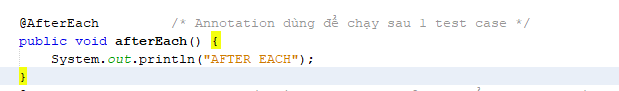


*Hàm thực thi trước khi 1 test case chạy xong*

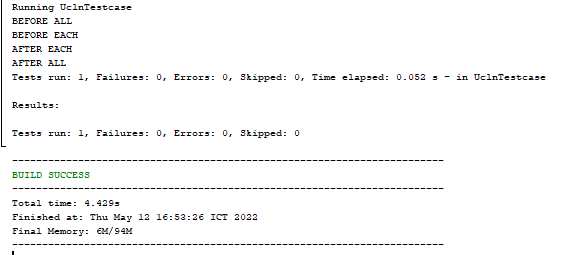


*Kết quả*

**After Each**



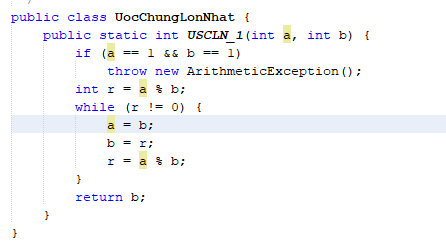
*Hàm thực thi sau khi 1 test case chạy xong*



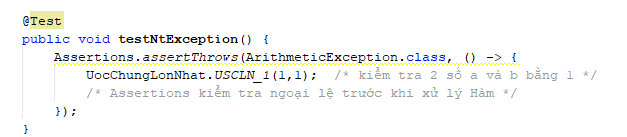
*Kết quả*

**2/ Một số Assertions:**

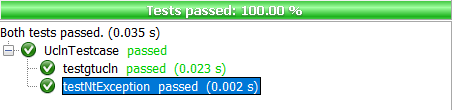
**2.1/ Ngoại lệ (Exception):**

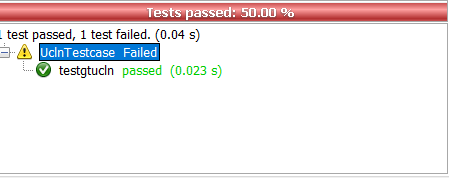


*Thêm 1 ngoại lệ nếu a =1 và b = 1*

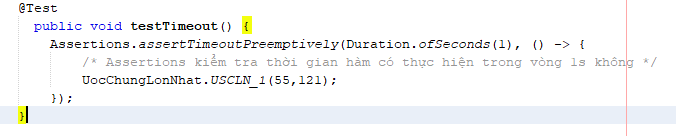


*Ta kiểm tra ngoại lệ 2 nếu số a,b bằng 1*

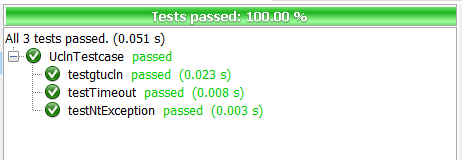




**2.2/ Kiểm tra thời gian chạy:**

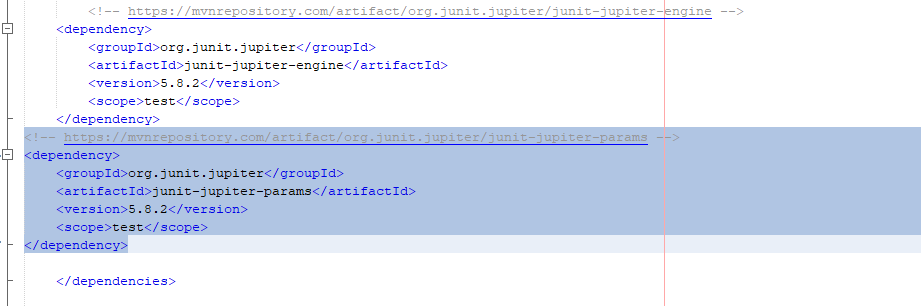


*Ta thực hiện test thời gian hàm thực thi*

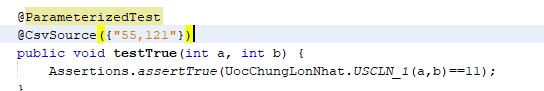


*Kết quả thành công*

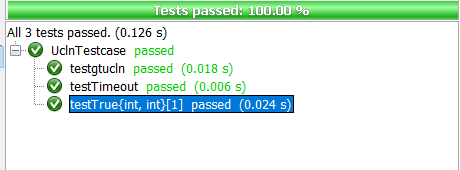
**3/ Parameterized test:**



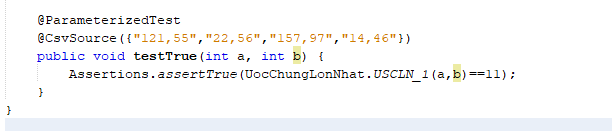
*Ta thêm Parameterized test vào file pom.xml để test nhiều giá trị*



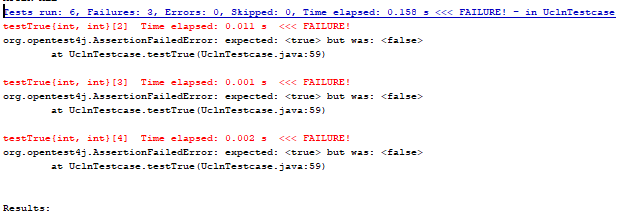
*Ta test thử 2 giá trị (a,b)={55,121} với đầu vào là CsvSource có bằng 11 hay không?*



*Kết quả đúng*



*Ta thêm nhiều giá trị hơn để test*



*Kết quả có 3 biến Fail*

**----- HẾT -----**

**(Chúc các bạn làm tốt!)**