



KIỂM THỬ NÂNG CAO

BÀI 3: KIỂM THỬ ĐƠN VỊ

- ◎ Tầm quan trọng kiểm thử đơn vị (Unit Test)
- ◎ Tìm hiểu Junit
- ◎ Ứng dụng Junit



Phần I: Tầm quan trọng kiểm thử đơn vị

 Khái niệm Unit Testing

 Kỹ thuật viết test case cho Unit Testing

 Công cụ Junit

Phần II: Ứng dụng Junit

 Cấu hình Junit

 Junit và Eclipse





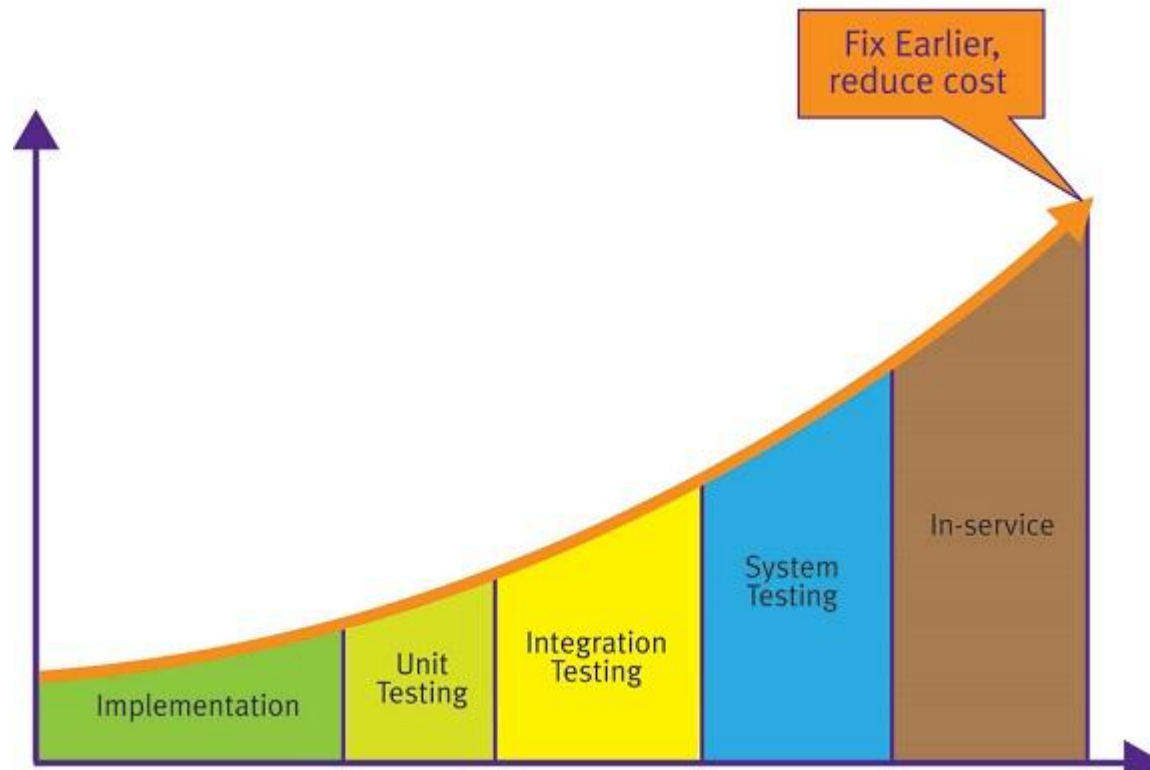
**Quá trình kiểm thử từng đơn vị phần mềm
xem có hoạt động đúng như thiết kế hay
không?**

- ☐ Một dòng lệnh
- ☐ Một hàm, phương thức
- ☐ Một module
- ☐ Một chương trình

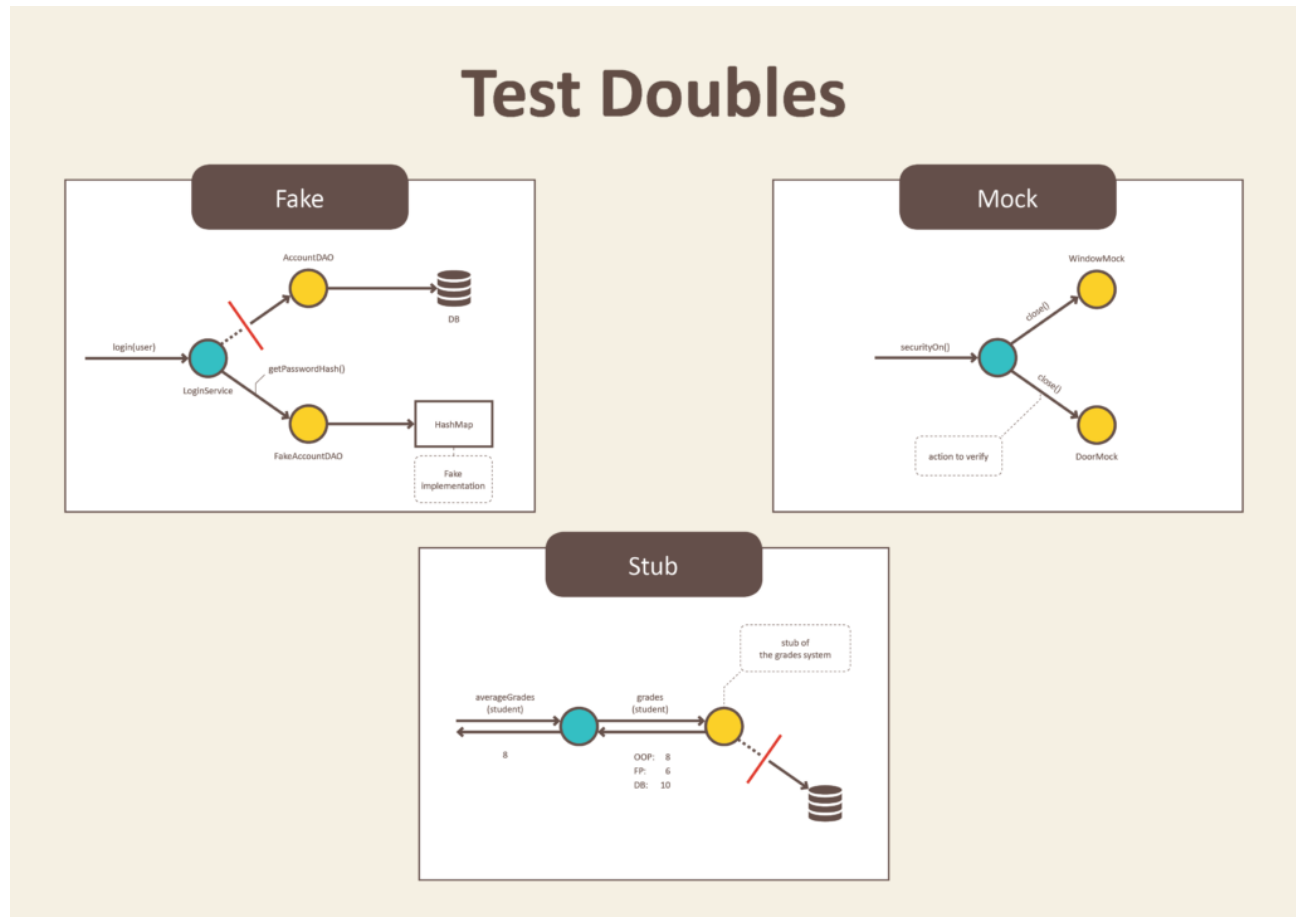


- ❑ Phát hiện lỗi nhanh
- ❑ Cải tiến quy trình và thiết kế sản phẩm
- ❑ Giảm chi phí sửa lỗi (money và time)

The Cost of Defects



- ❑ Unit test có thể cần nhiều thành phần để thực hiện.
- ❑ Các đối tượng giả định
 - ❖ Mock
 - ❖ Stub
 - ❖ Fake



- ❑ Thực hiện automated hoặc manual
- ❑ Quá trình automated:
 - ❖ Viết đoạn code test trong ứng dụng để kiểm tra các chức năng
 - ❖ Có thể tạo ra các đối tượng giả lập hỗ trợ unit test

Unit Tests



- ☐ Hãy làm cho mỗi test độc lập với tất cả những phần khác
- ☐ Đừng làm những assertion không cần thiết
- ☐ Kiểm thử chỉ một unit code tại một thời điểm
- ☐ Giả lập tất cả những dịch vụ và trạng thái bên ngoài
- ☐ Tránh những điều kiện tiên quyết không cần thiết
- ☐ Đặt tên các unit test của bạn một cách rõ ràng và nhất quán

- ❑ JUnit là 1 framework mã nguồn mở được sử dụng để viết và chạy thử nghiệm
- ❑ JUnit là một framework dùng để unit test cho ngôn ngữ lập trình Java



- ❑ Cung cấp Annotation để xác định xác thực phương thức

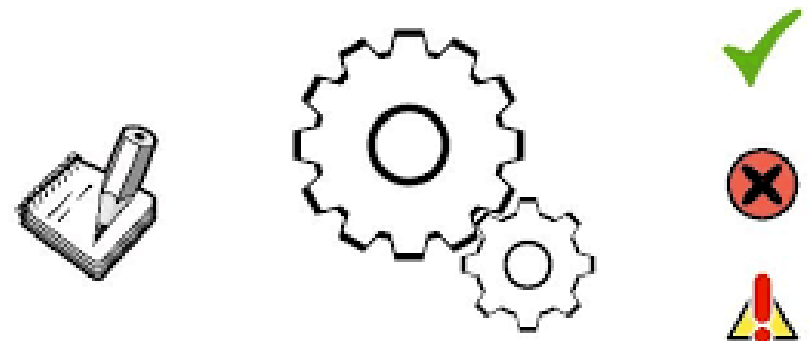


- ❑ Cung cấp Assertion thể hiện kết quả mong muốn

Assertion Methods

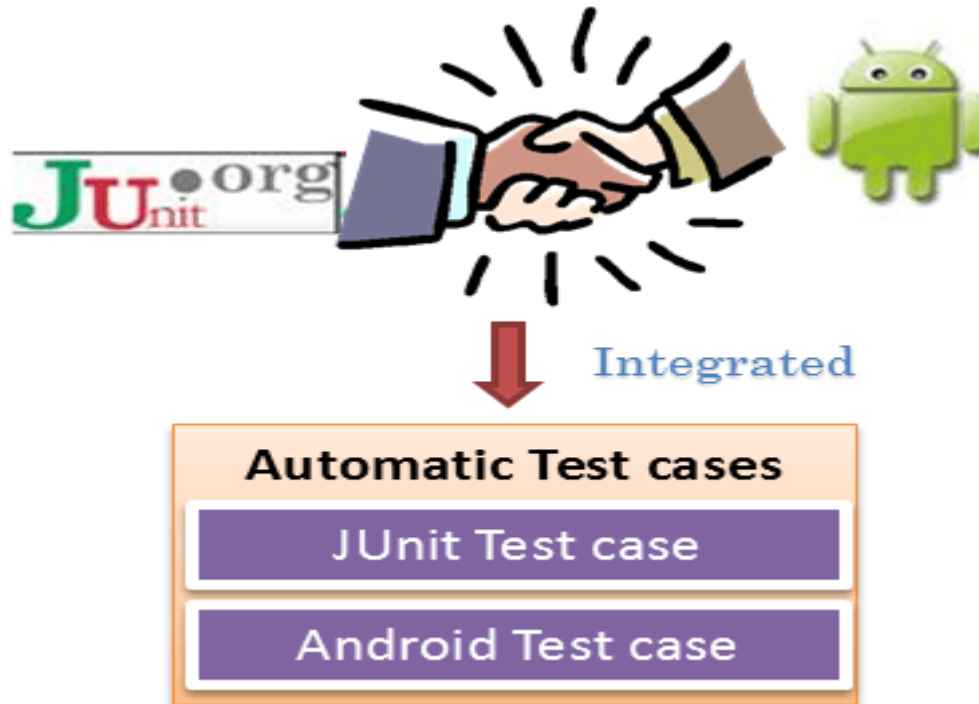
```
assertTrue(String message, boolean condition)
assertFalse(String message, boolean condition)
assertEquals(String message, double expected, double actual, double delta)
assertEquals(String message, long expected, long actual)
assertEquals(String message, Object expected, Object actual)
assertNotNull(String message, Object object)
assertNull(String message, Object object)
assertArrayEquals(String message, byte[] expecteds, byte[] actuals)
assertArrayEquals(String message, char[] expecteds, char[] actuals)
assertArrayEquals(String message, int[] expecteds, int[] actuals)
assertArrayEquals(String message, long[] expecteds, long[] actuals)
assertArrayEquals(String message, Object[] expecteds, Object[] actuals)
-
```

- ❑ Cung cấp các runner cho việc chạy thử
- ❑ JUnit cho thấy kết quả pass là màu xanh và fail là màu đỏ
- ❑ Test JUnit có thể được tổ chức thành các case và sự kiện test khác

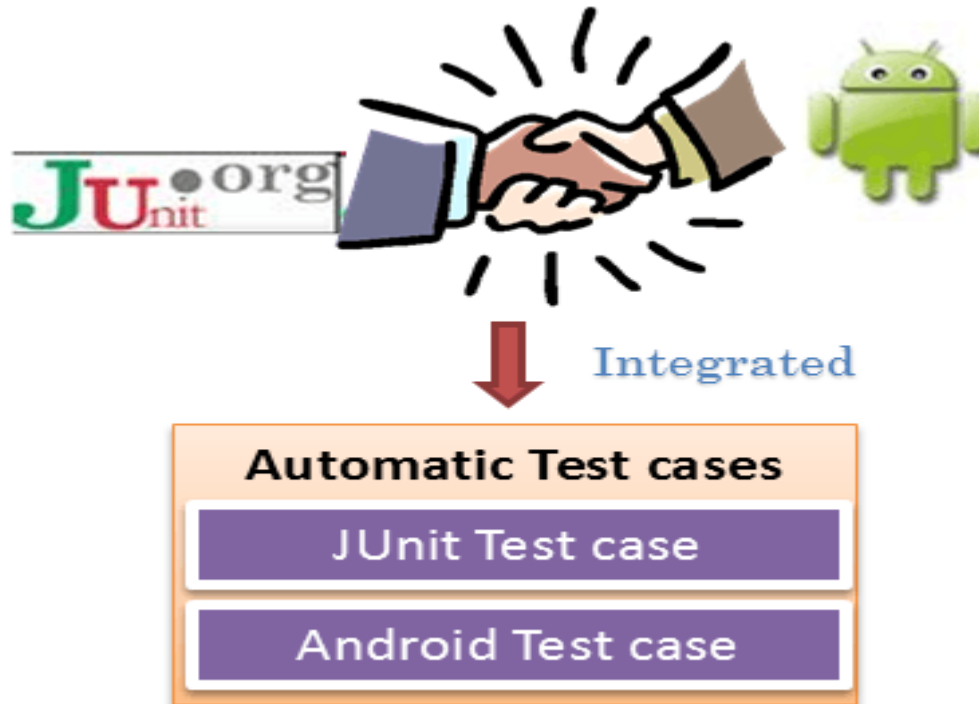


Test Runner

- ❑ JUnit có thể tự động chạy, kiểm tra chính nó , cho kết quả theo thời gian thực.
- ❑ JUnit hỗ trợ viết mã nhanh hơn và chính xác trên các framework dùng ngôn ngữ lập trình java



- ❑ JUnit có thể tự động chạy, kiểm tra chính nó , cho kết quả theo thời gian thực.
- ❑ JUnit hỗ trợ viết mã nhanh hơn và chính xác trên các framework dùng ngôn ngữ lập trình java





DEMO

- Chuẩn bị môi trường chạy Junit
- Cài đặt Java SDK





KIỂM THỬ NÂNG CAO

BÀI 3: KIỂM THỬ ĐƠN VỊ (P2)

❑ Download và Install (<http://junit.org/junit4/>)

JUnit

JUnit 4 / About

Version: 4.12 | Last Published: 2018-02-07

Let's take a tour »

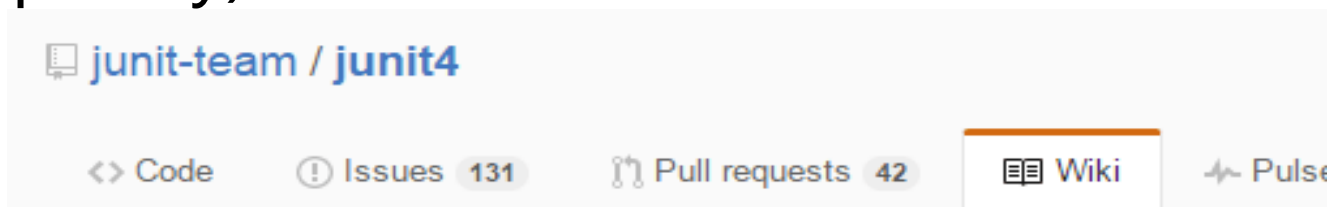
Welcome

- [Download and install](#)
- [Getting started](#)
- [Release Notes](#)
 - [4.12](#)

Usage and Idioms

- [Assertions](#)
- [Test Runners](#)
- [Aggregating tests in Suites](#)
- [Test Execution Order](#)

- ❑ Chọn file junit.jar (tạo thư mục riêng lưu trữ để dễ quản lý)



Download and Install

Shaun Reich edited this page on Nov 29, 2015 · 20 revisions

To download and install JUnit you currently have the following options

Plain-old JAR

Download the following JARs and put them on your test classpath:

- `junit.jar`
- `hamcrest-core.jar`

- Trong mục central repository lựa chọn version 4.12 (chạy ổn định)

The Central Repository

SEARCH | ADVANCED SEARCH | BROWSE | QUICK STATS

g:"junit" AND a:"junit"

SEARCH

[New: About Central](#)

[Advanced Search](#) | [API Guide](#) | [Help](#)

Search Results

< 1 2 > displaying 1 to 20 of 24

GroupId	ArtifactId	Version	Updated	Download
junit	junit	4.12	04-Dec-2014	pom jar javadoc.jar sources.jar
junit	junit	4.12-beta-3	09-Nov-2014	pom jar javadoc.jar sources.jar
junit	junit	4.12-beta-2	25-Sep-2014	pom jar javadoc.jar sources.jar
junit	junit	4.12-beta-1	28-Jul-2014	pom jar javadoc.jar sources.jar
junit	junit	4.11	15-Nov-2012	pom jar javadoc.jar sources.jar
junit	junit	4.11-beta-1	16-Oct-2012	pom jar javadoc.jar sources.jar
junit	junit	4.10	30-Sep-2011	pom jar javadoc.jar sources.jar
junit	junit	4.9	22-Aug-2011	pom jar javadoc.jar sources.jar

- ❑ Tải tập tin “hamcrest-core.jar” từ link <https://github.com/junitteam/junit4/wiki/Download-and-Install>

Plain-old JAR

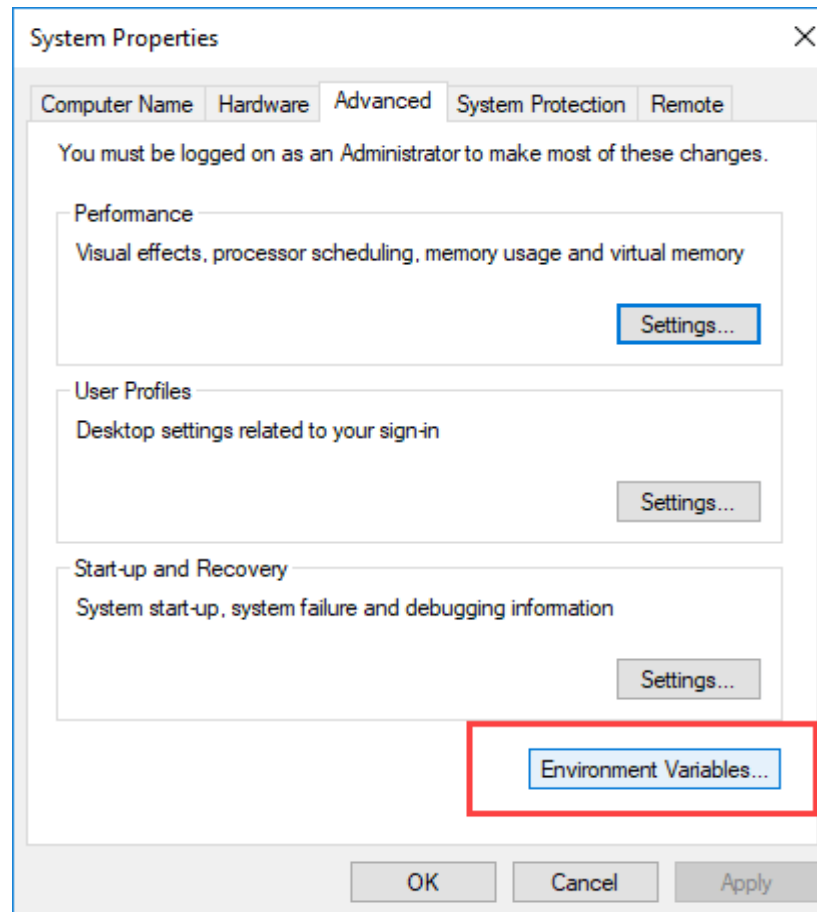
Download the following JARs and put them on your test classpath:

- `junit.jar`
- `hamcrest-core.jar`

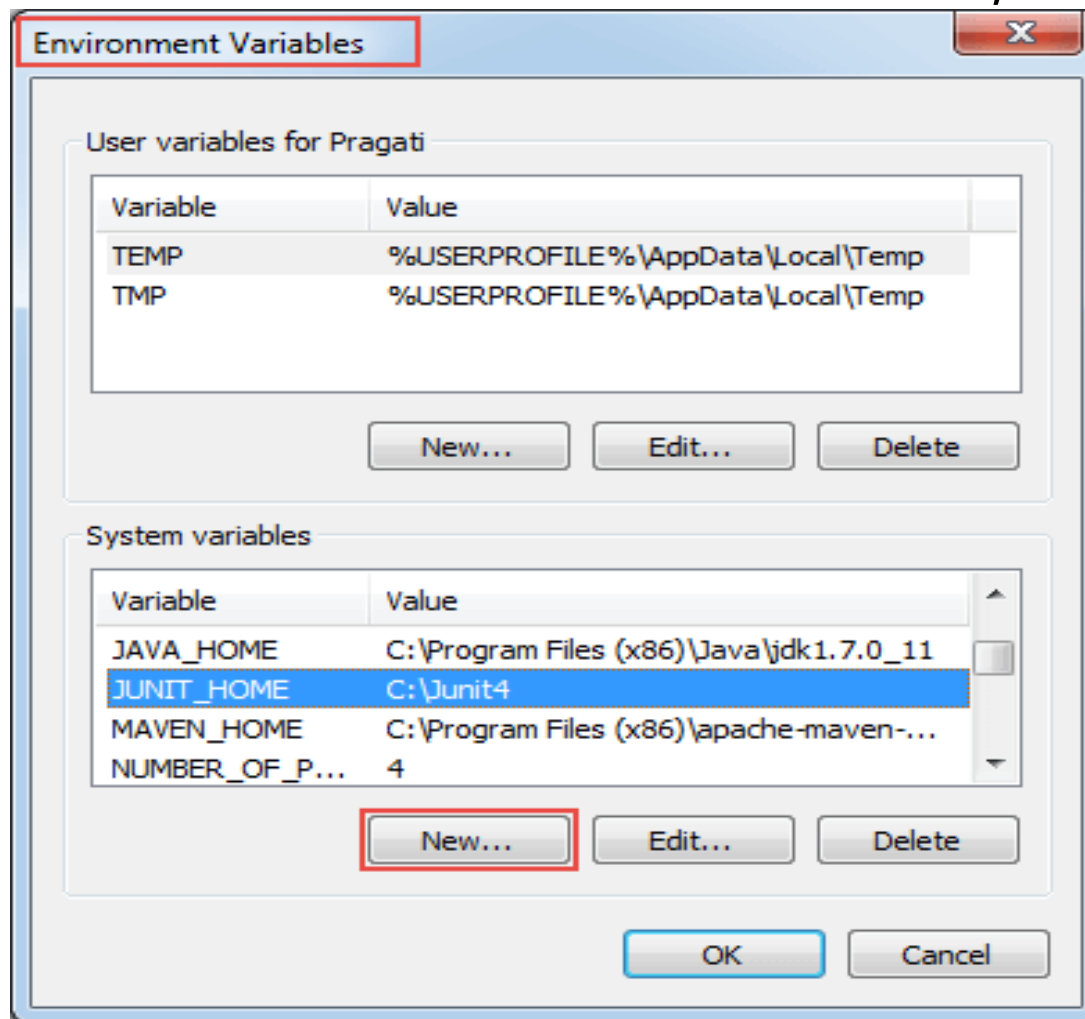
Search Results

GroupId	ArtifactId	Version	Updated	Download
org.hamcrest	hamcrest-core	1.3	10-Jul-2012	pom jar javadoc.jar sources.jar
org.hamcrest	hamcrest-core	1.3.RC2	22-Dec-2010	pom jar javadoc.jar sources.jar
org.hamcrest	hamcrest-core	1.3.1	22-Dec-2010	pom jar javadoc.jar sources.jar

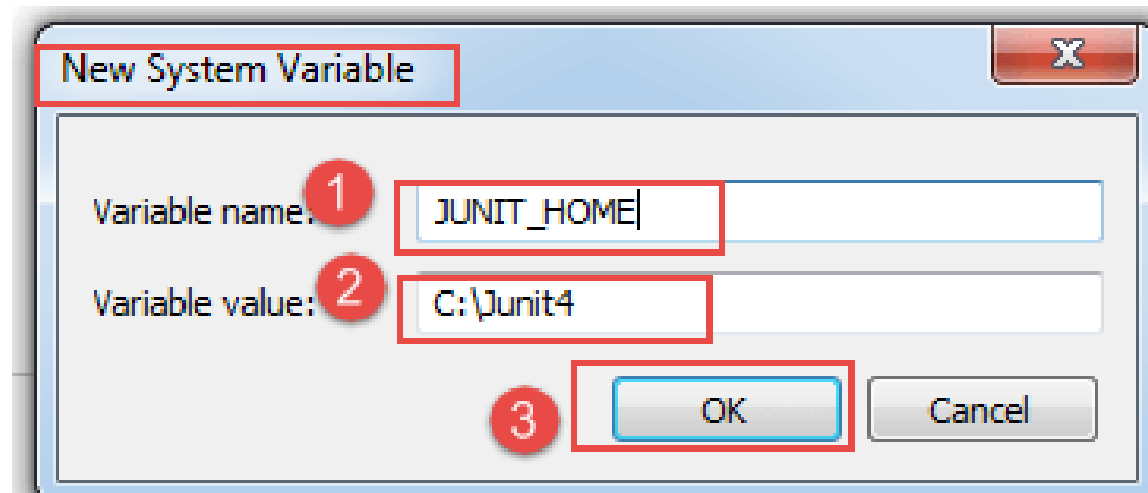
- ❑ Thiết lập biến môi trường JUNIT_HOME
- ❑ Chuột phải my computer -> advanced system settings -> Environment Variables



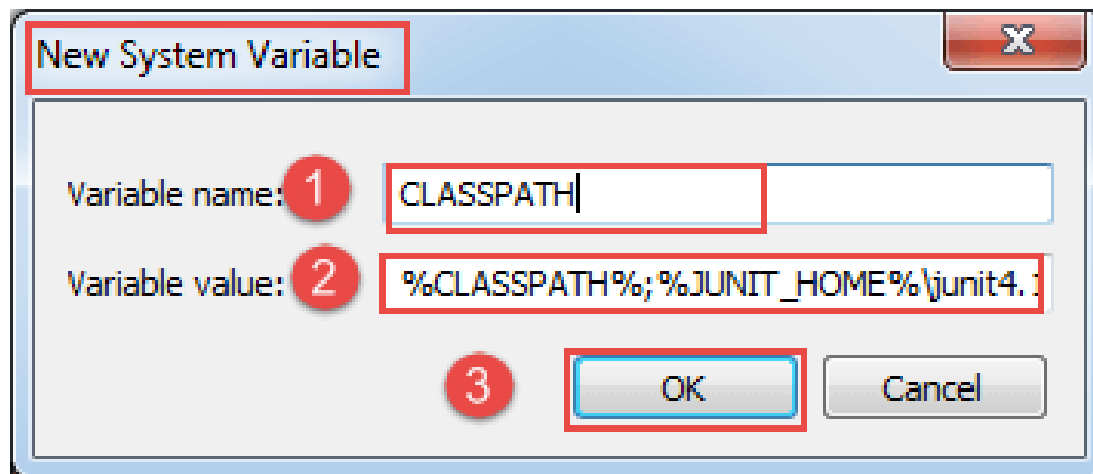
- ❑ Từ màn hình Environment Variables, chọn "new"



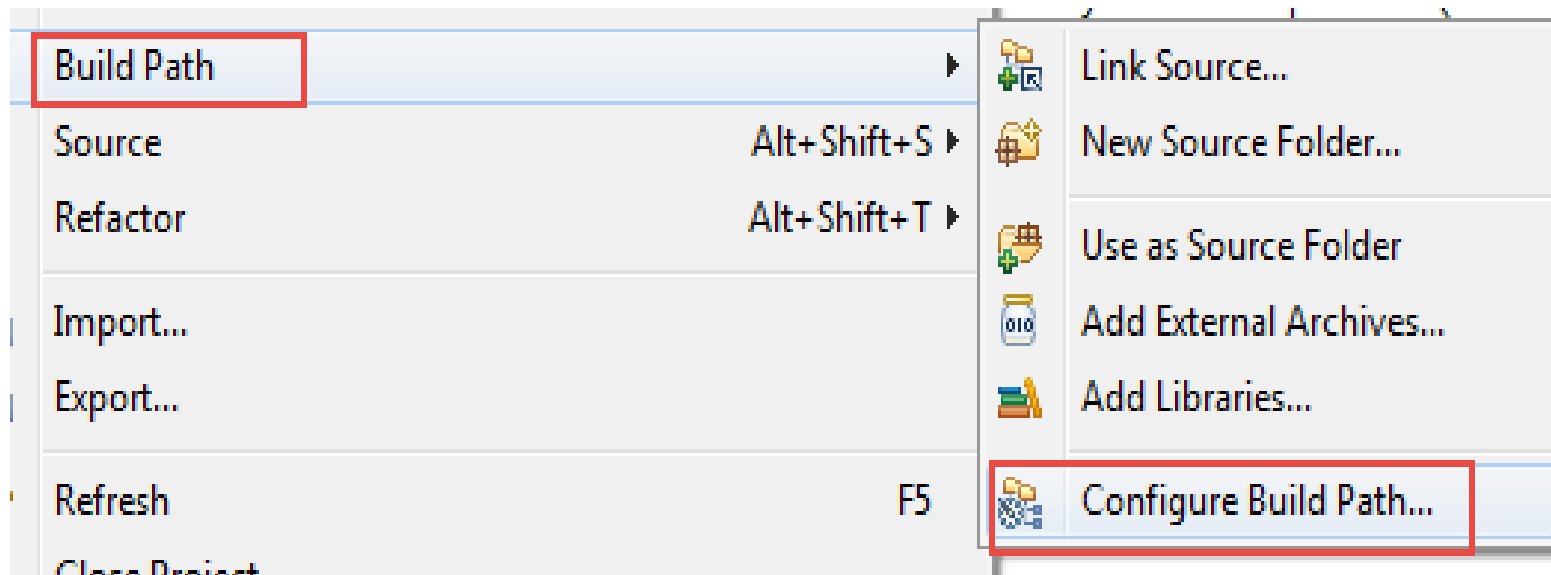
- ❑ Bên trong "New System Variable" ta cung cấp các thông tin để tạo mới biến môi trường trong hệ thống (thư mục Junit4 được tạo ra từ trước lưu Unit.jar và hamcrest-core.jar)



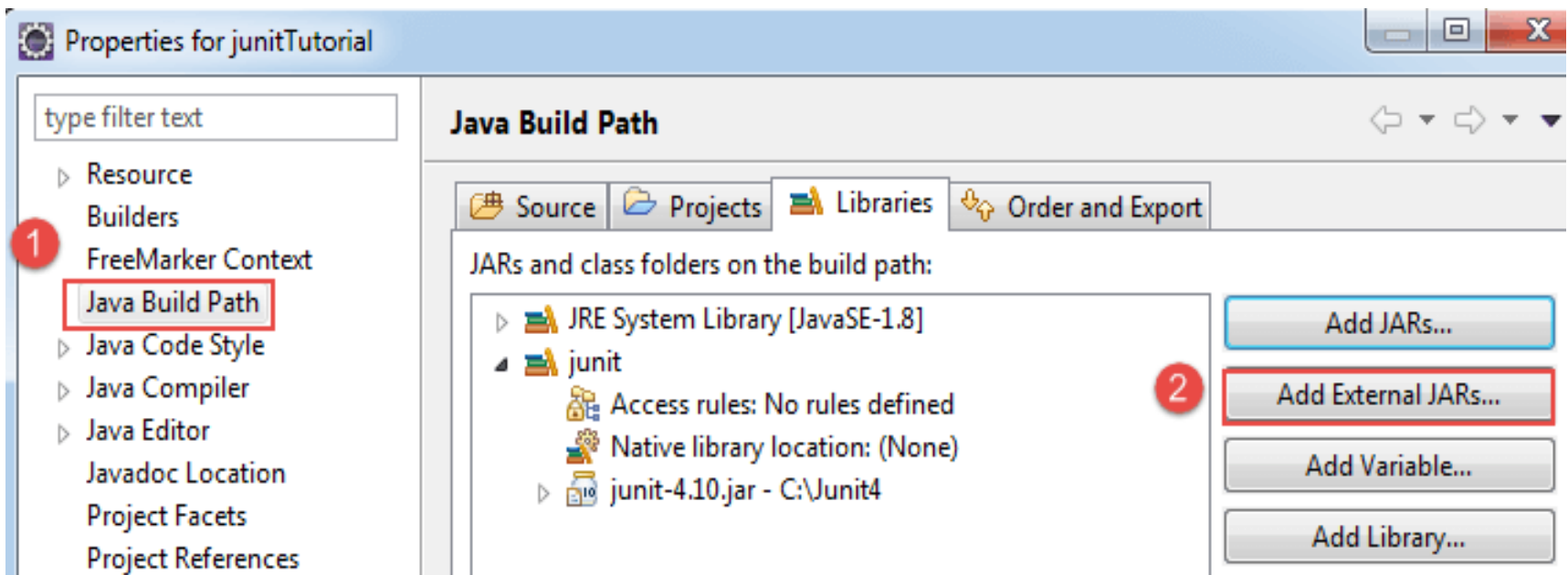
- ❑ Tương tự tạo biến tên CLASSPATH, lưu ý trong phần “Variable Value” cần liên kết tới thư mục chứa Junit.jar
- ❑ Variable Value:
`%CLASSPATH%;%JUNIT_HOME%\JUnit4.10.jar;.`



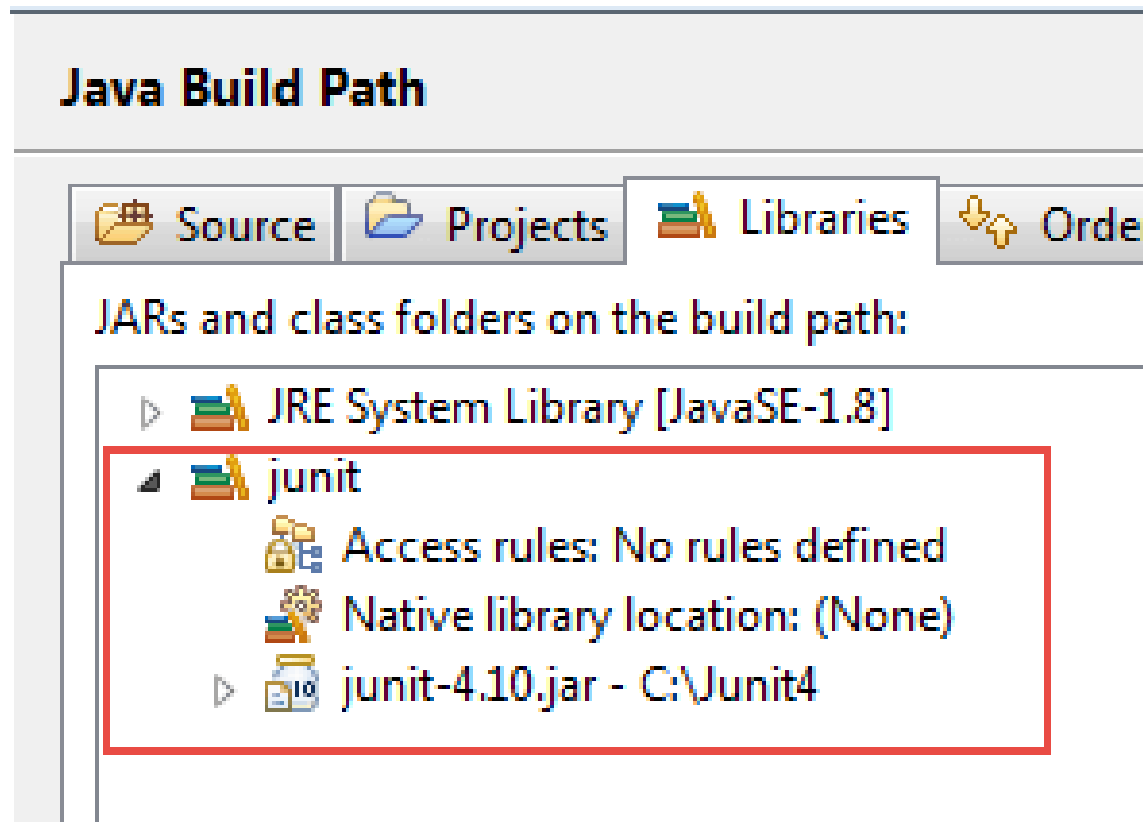
- ❑ Chọn phải chuột vào project, chọn "build path" và chọn "Configure build path"



- ❑ Tìm vào mục "java build path"
- ❑ Chọn "Add External JARs" để tải Junit.jar cho eclipse, nhấn ok để thực hiện



- ❑ Kiểm tra việc cấu hình thành công junit.jar trên eclipse bằng cách vào "Configure build path", chọn "libraries"



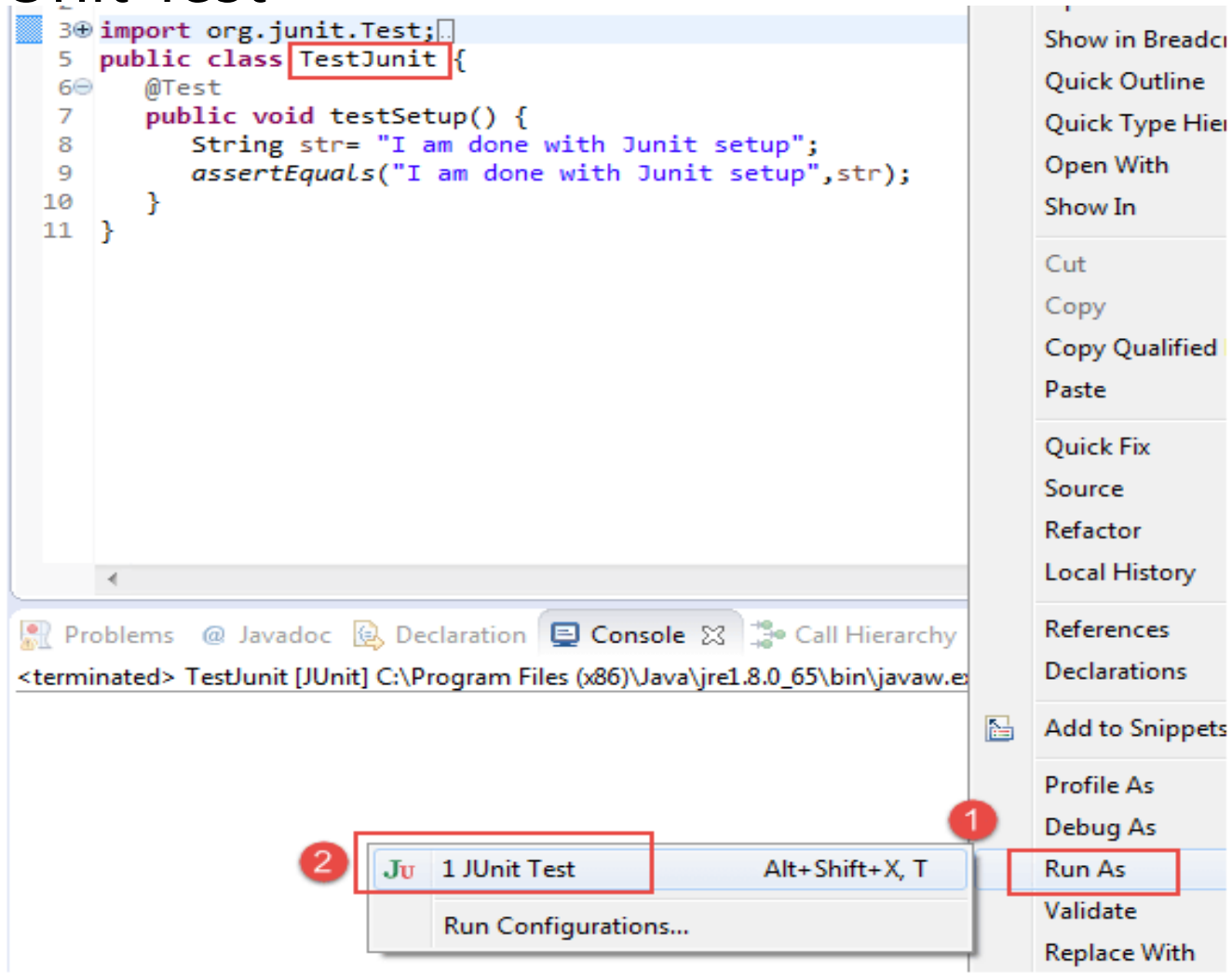
❑ Tạo class có name "TestJUnit.java"

```
1 package test1;
2 import static org.junit.Assert.*;
3 import org.junit.Test;
4 import static org.junit.Assert.assertEquals;
5 public class TestJUnit {
6     @Test
7     public void testSetup() {
8         String str= "Fpoly demo Junit";
9         assertEquals("Fpoly demo Junit",str);//so sanh str co giiong voi "Fpoly demo Junit"
10    }
11 }
```

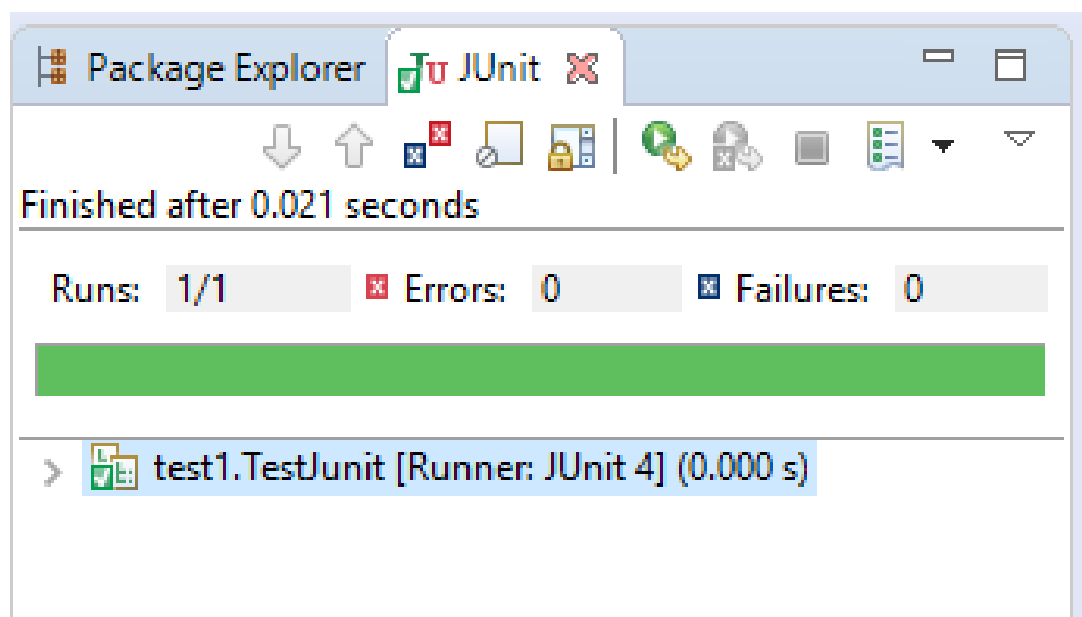
- ❑ Tạo class có name "Test Runner" để thực thi lớp "TestJUnit"

```
3+ import org.junit.runner.JUnitCore;
6
7 public class TestRunner {
8-     public static void main(String[] args) {
9         Result result = JUnitCore.runClasses(TestJUnit.class);
10        for (Failure failure : result.getFailures()) {
11            System.out.println(failure.toString());
12        }
13        System.out.println("Result==" + result.wasSuccessful());
14    }
15 }
```

- ❑ Thực thi test: phải chuột "JUnit" -> run as -> "1 JUnit Test"




❑ Kết quả thực thi test thành công



Tổng kết bài học

Phần I: Tầm quan trọng kiểm thử đơn vị

 Khái niệm Unit Testing

 Kỹ thuật viết test case cho Unit Testing

 Công cụ Junit

Phần II: Ứng dụng Junit

 Cấu hình Junit

 Junit và Eclipse





KẾT THÚC