

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

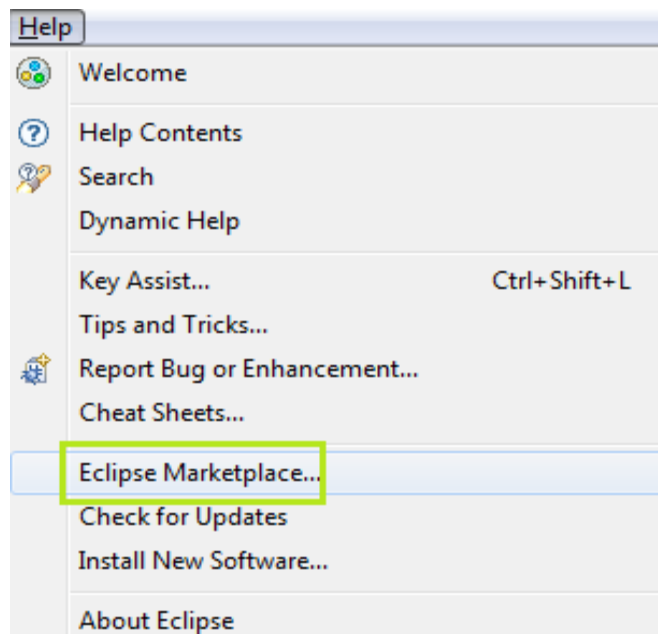
- ✓ Hiểu được các khái niệm về TestNG
- ✓ Tích hợp TestNG và Eclipse
- ✓ Tạo lớp kiểm thử với TestNG

PHẦN I

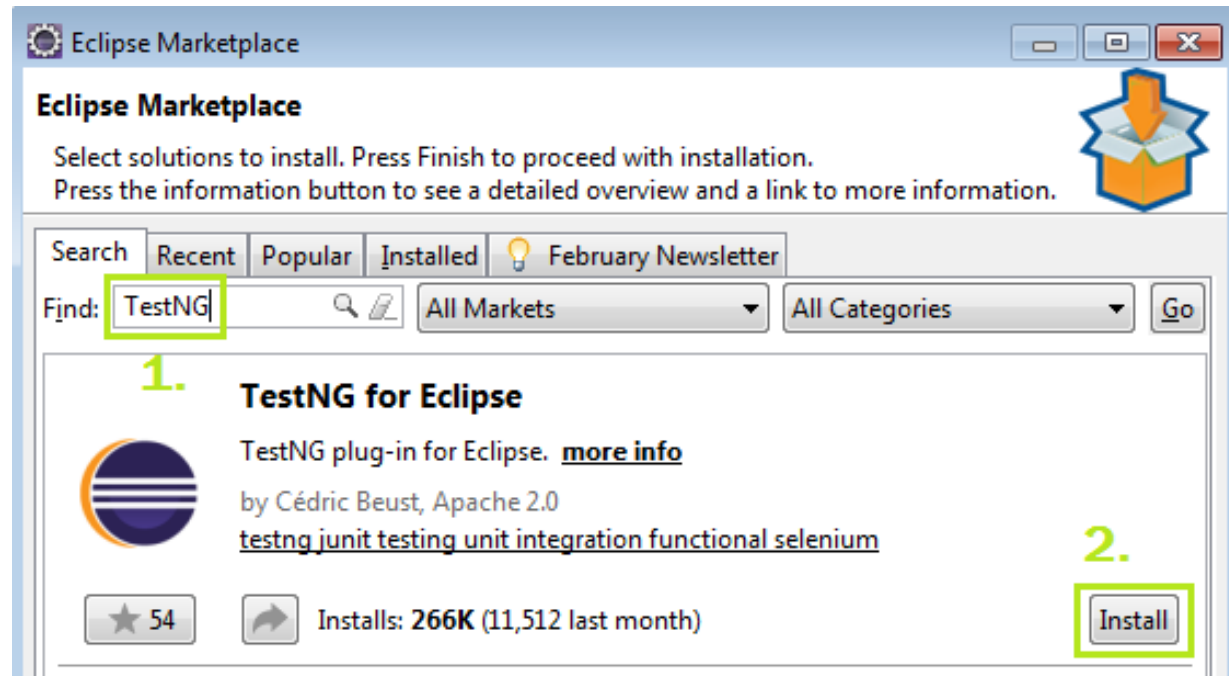
Bài 1 (2 điểm)

Cài đặt TestNG với Eclipse

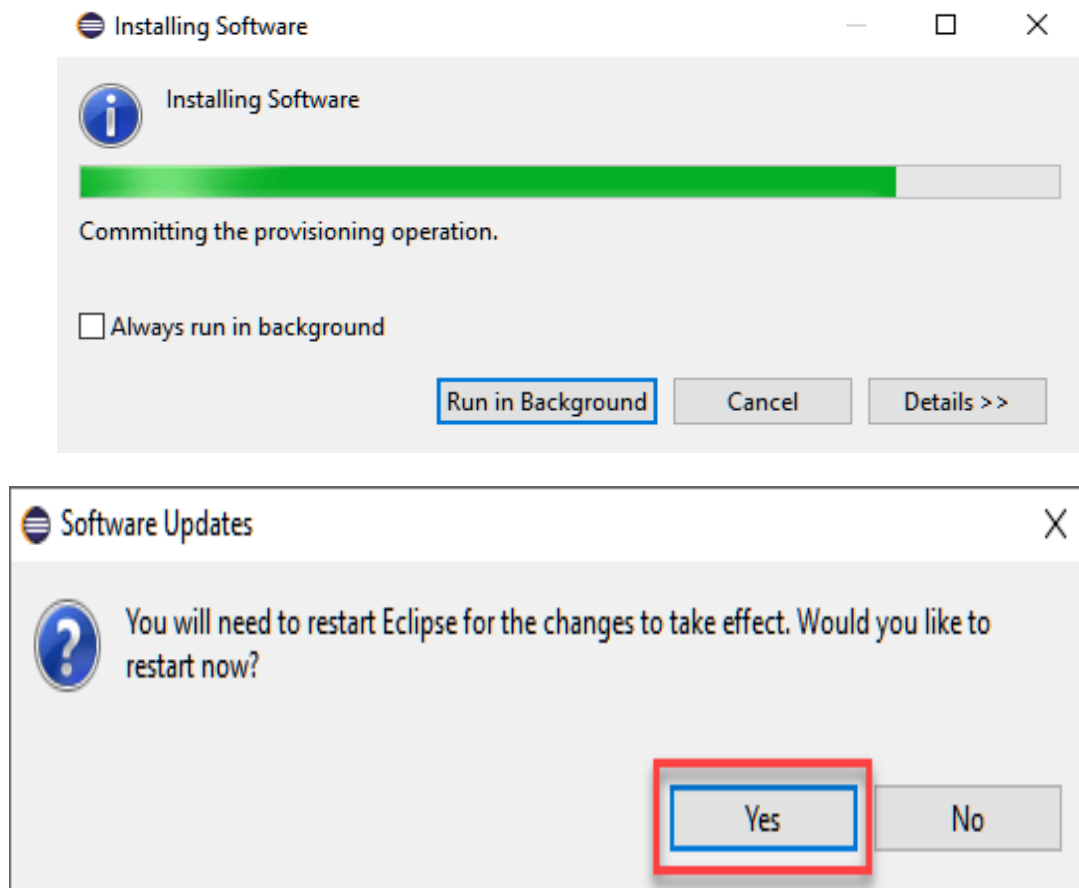
- Sử dụng Eclipse Marketplace
 - Mở Eclipse, từ menu help chọn “Eclipse Marketplace...”



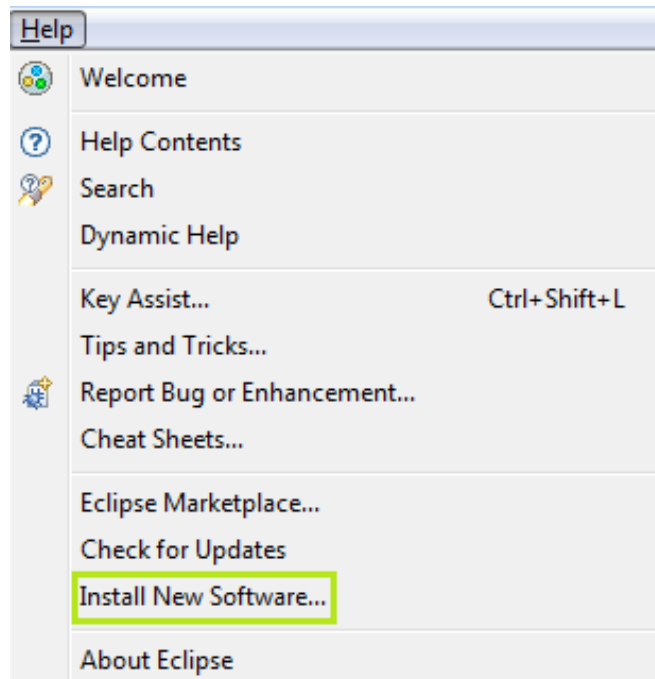
- Tại cửa sổ “Eclipse Marketplace”, gõ “TestNG” vào ô tìm kiếm, click “Go” để tìm plugin
- Chọn mục “testing for eclipse” và nhấn Install để tiến hành cài đặt Testng



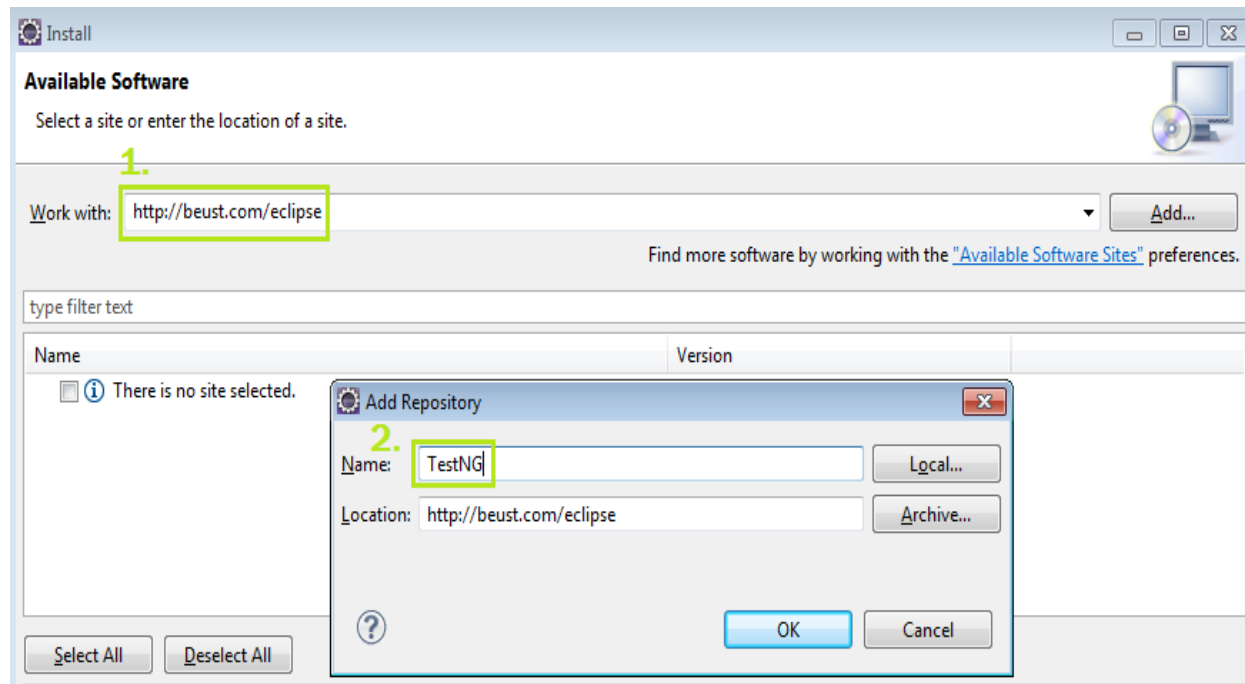
- Cài đặt bình thường như các plugin khác, khi thành công eclipse yêu cầu khởi động lại phần mềm



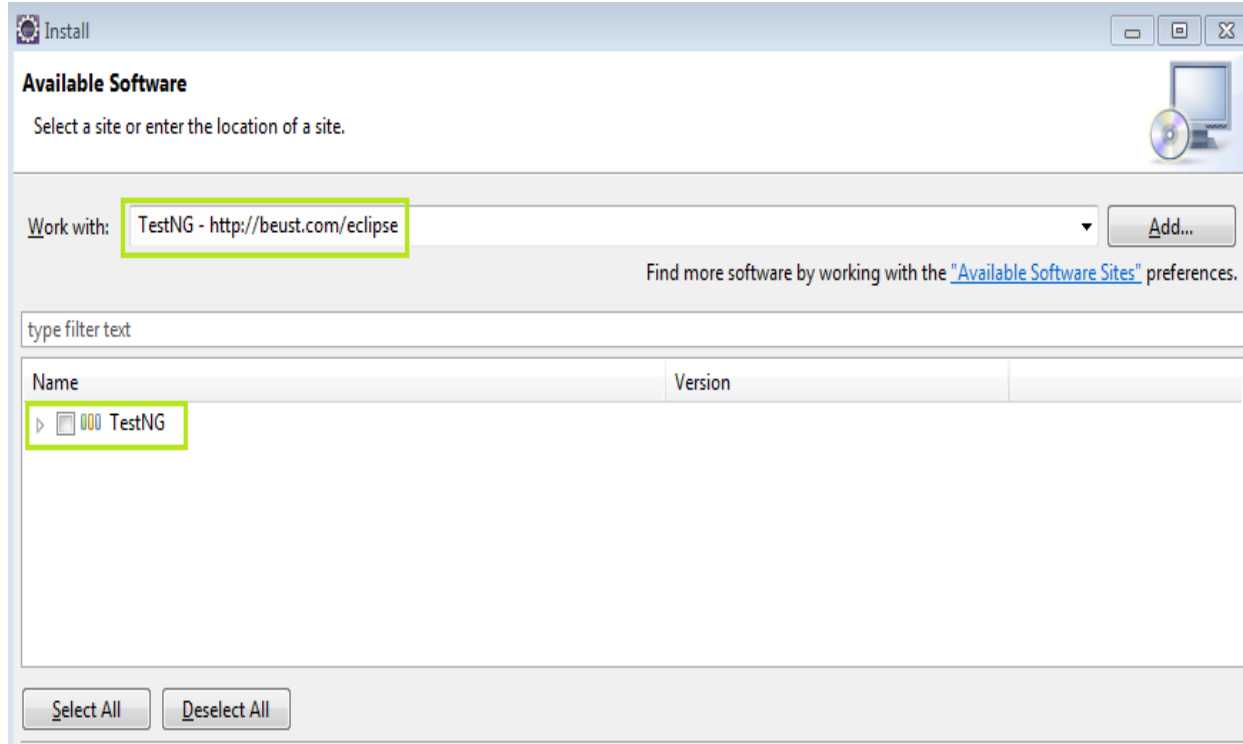
- Sử dụng “Install New Software”
 - Từ menu help chọn “Install New Software”



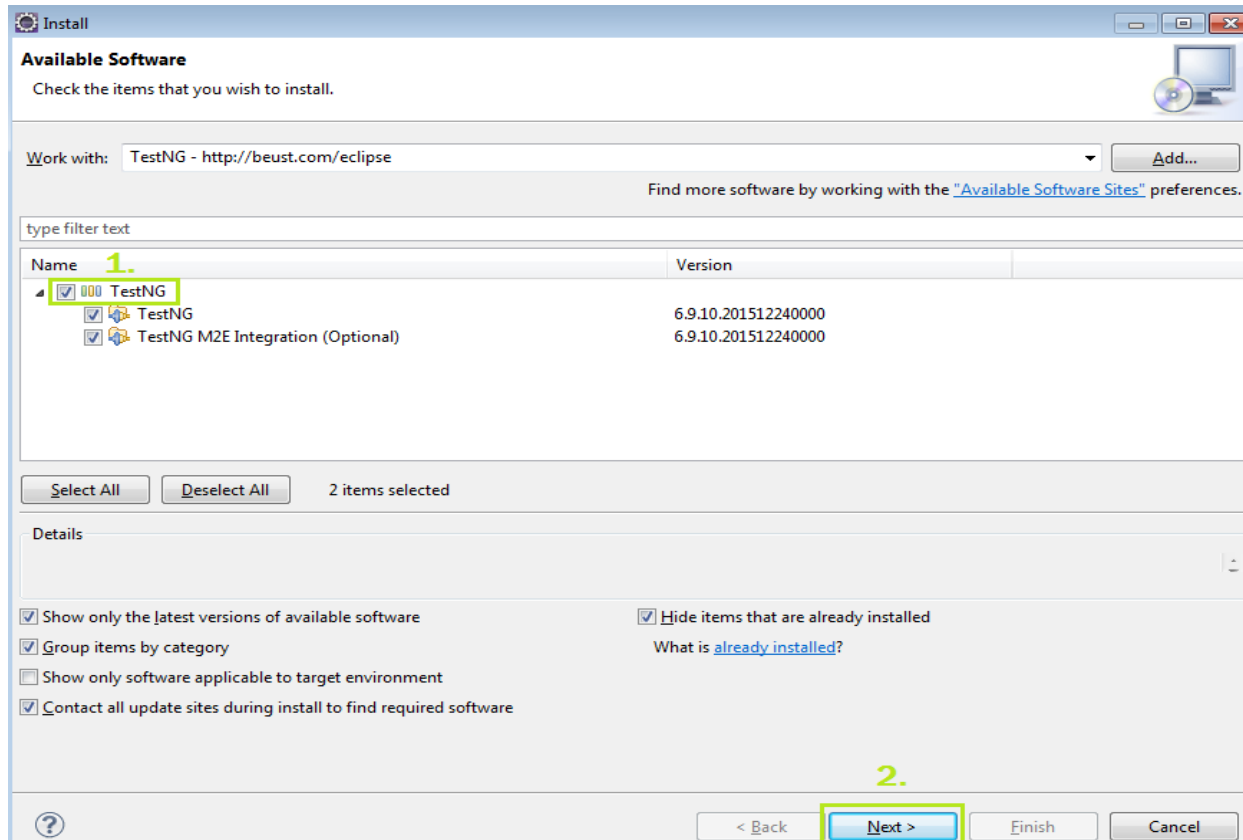
- Trên cửa sổ “install”, nhập link “http://beust.com/eclipse” vào ô “Work with”, nhấn “Add”
- Trên cửa sổ “Add repository” nhập “TestNG” và chọn OK



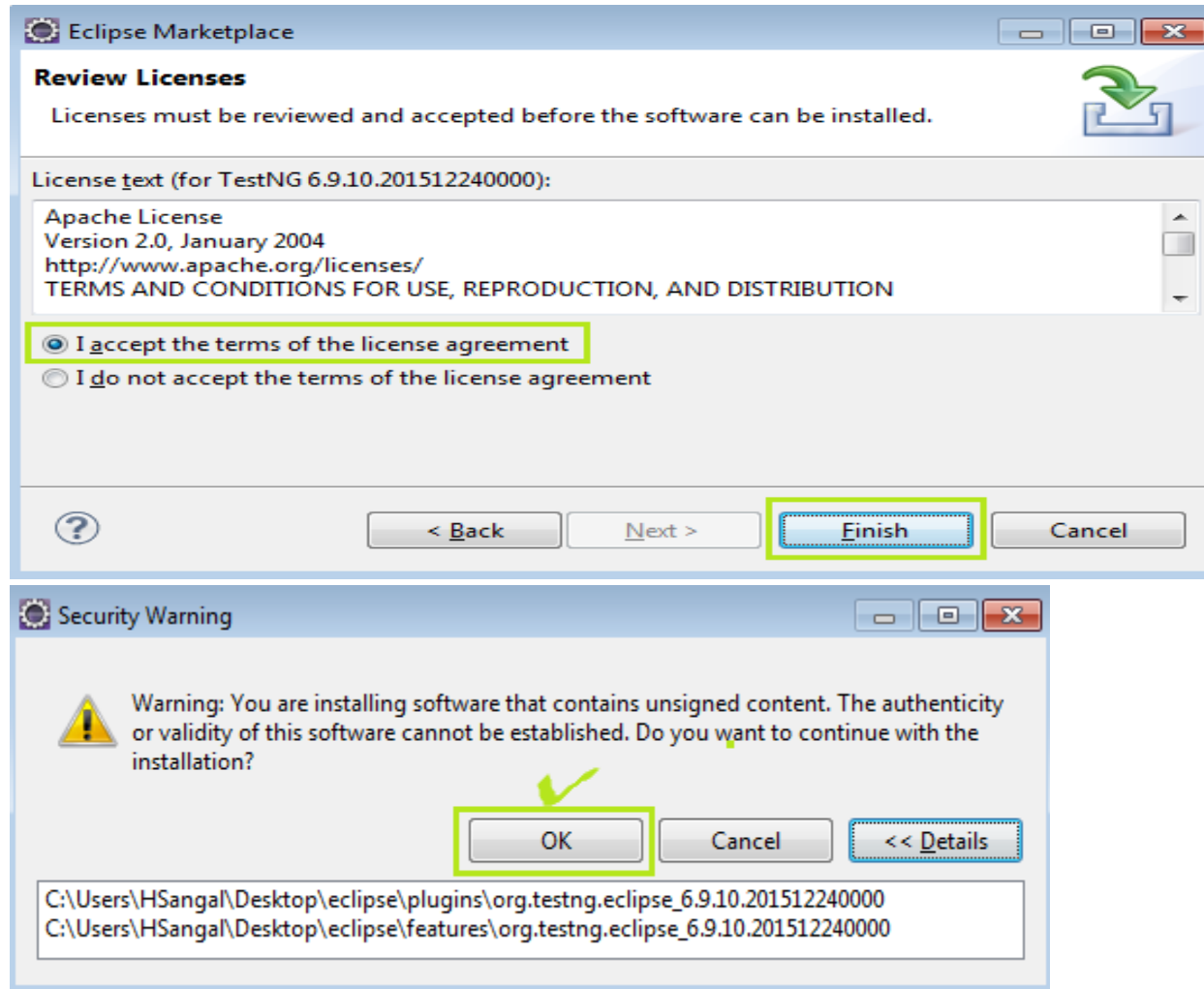
- Dựa vào đường link cung cấp, eclipse tìm plugin TestNG



- Chọn các mục TestNG và nhấn next



- Tiến hành cài đặt bình thường

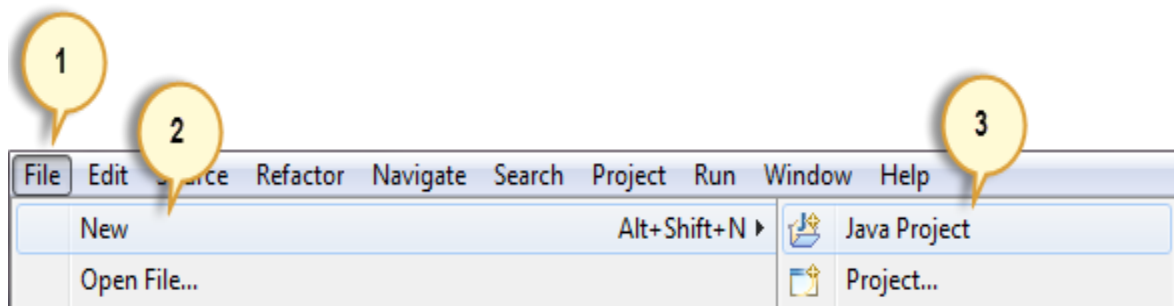


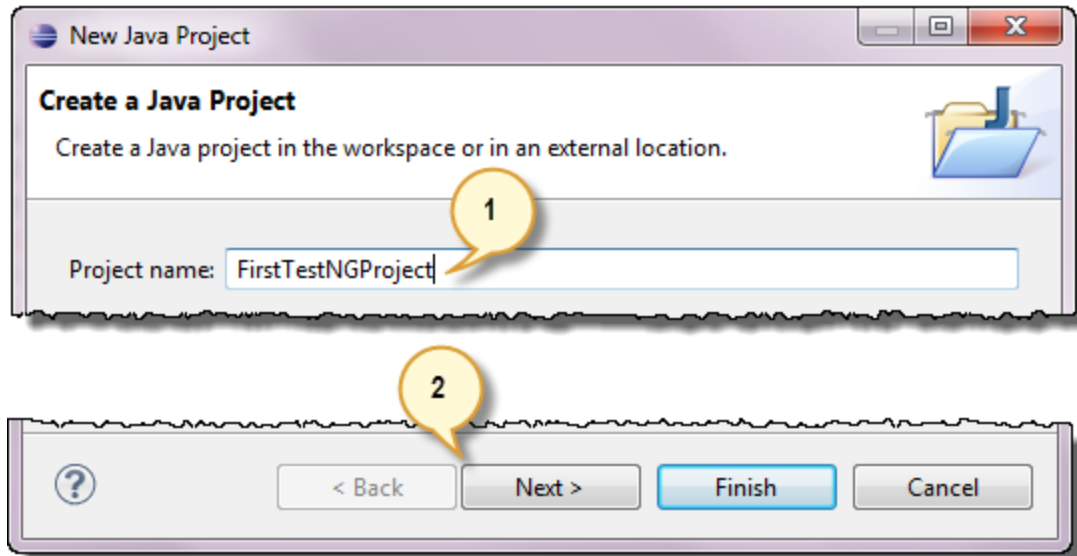
- Sử dụng file jar : Sinh Viên tự tìm hiểu cài đặt, chụp lại hình ảnh trong quá trình cài và nộp vào lab

Bài 2 (3 điểm)

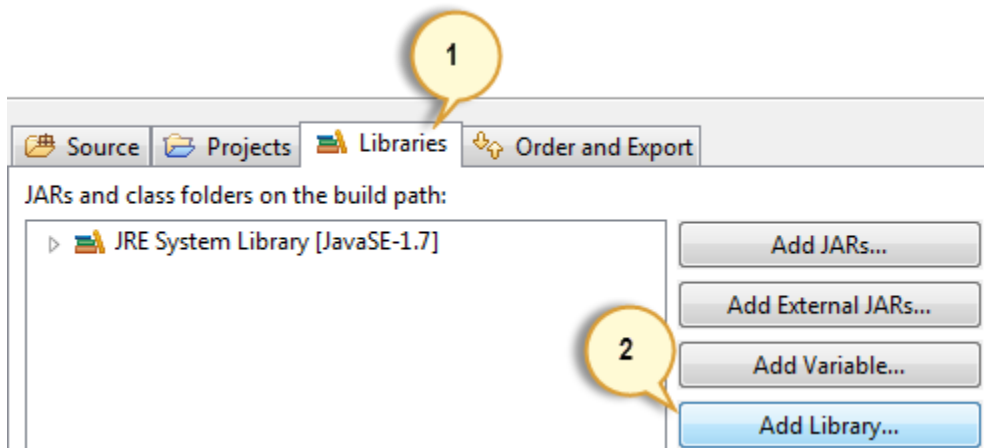
Xây dựng lớp TestNG với Eclipse và Selenium API.

- Tạo mới project có tên FirstTestNGFile

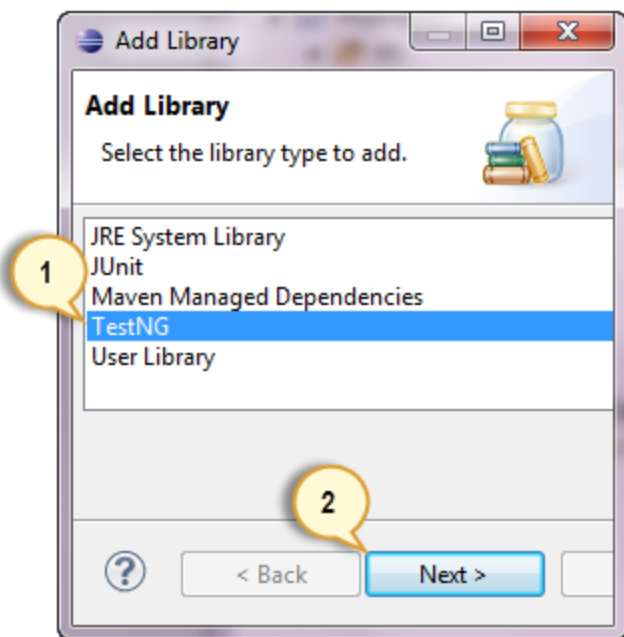




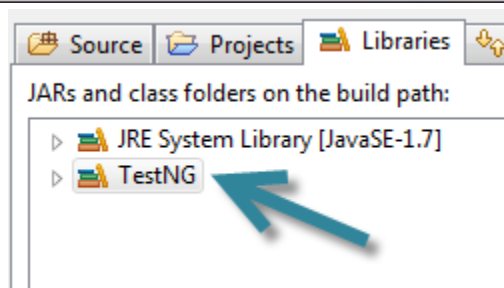
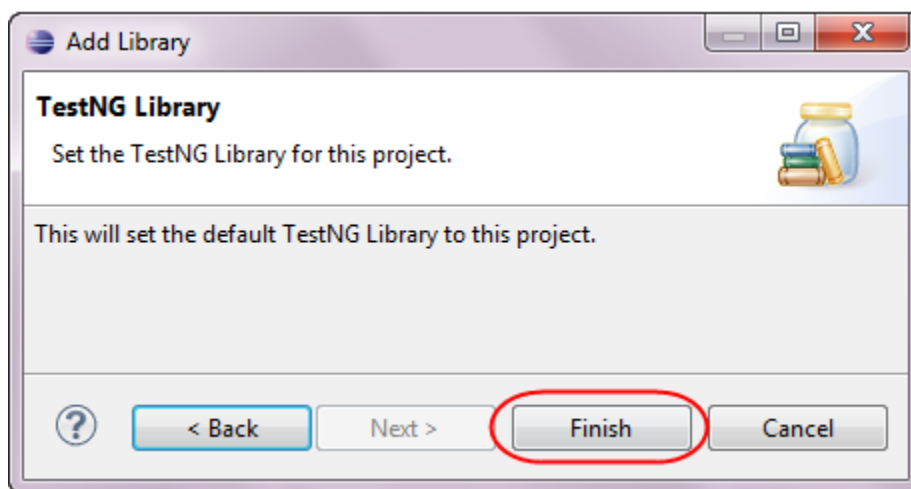
- Tiếp theo để Eclipse dùng được thư viện TestNG thì phải import vào. Chọn tab "Libraries" và chọn "Add Library..."



- Tiếp tục chọn "TestNG"



- Hoàn thành việc import thư viện TestNG, kiểm tra lại kết quả



- Tải các tập tin Selenium JAR từ link : <http://docs.seleniumhq.org/download/>

Language

Java

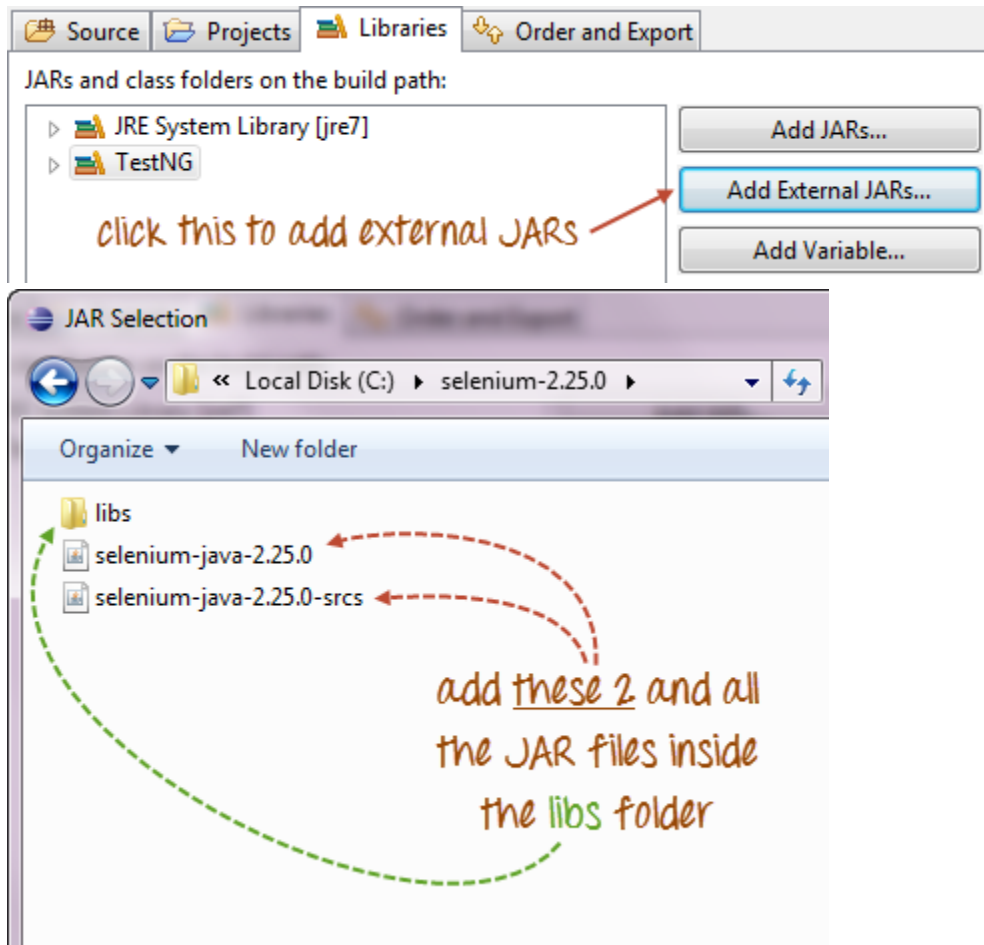
Client Version Release Date

3.12.0

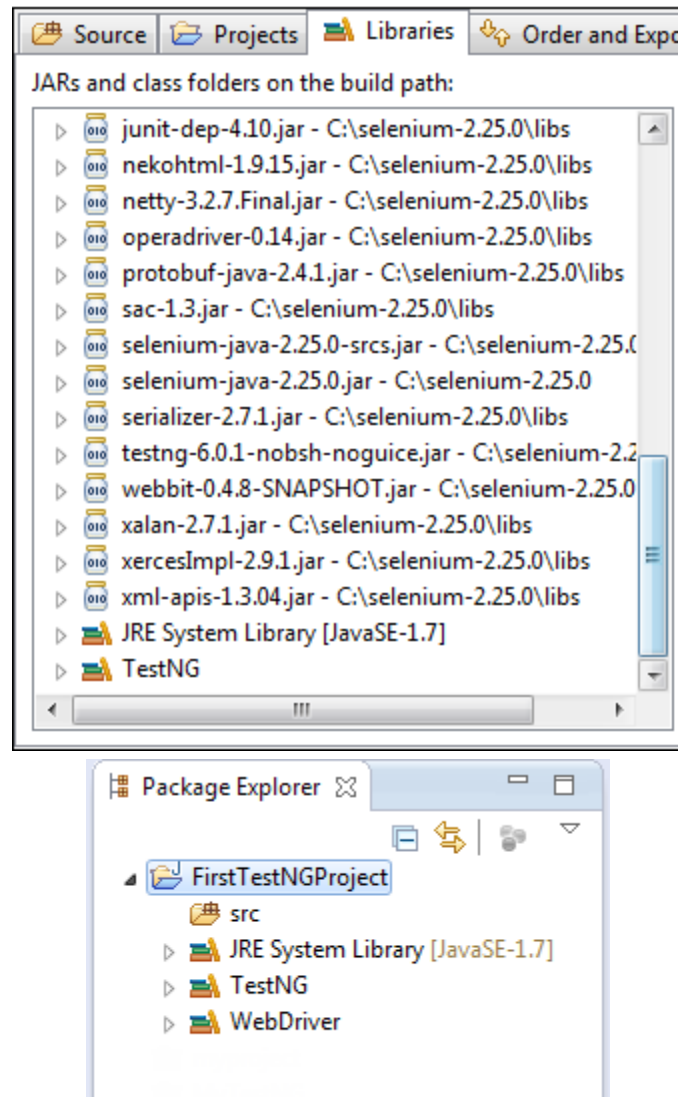
2018-05-08

[Download](#)

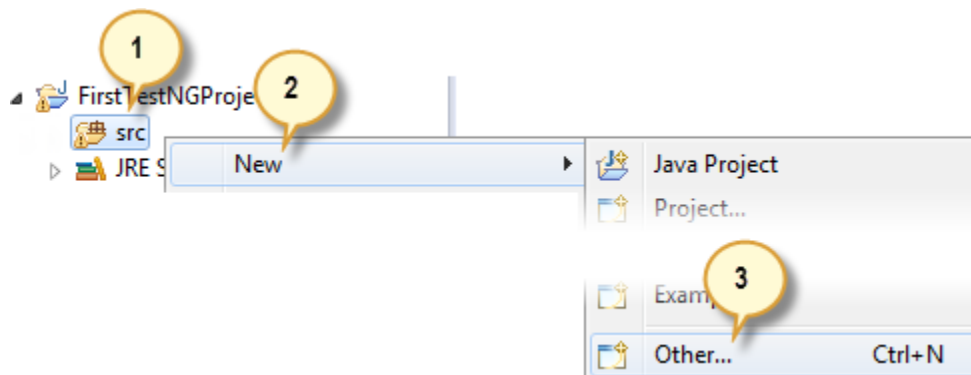
- Thêm các tập tin Selenium JAR vào project

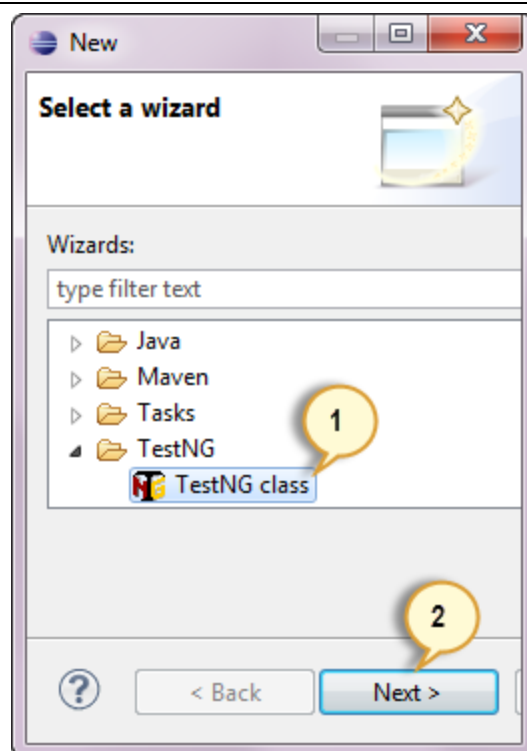


- Hoàn thành các bước cài đặt và kiểm tra kết quả

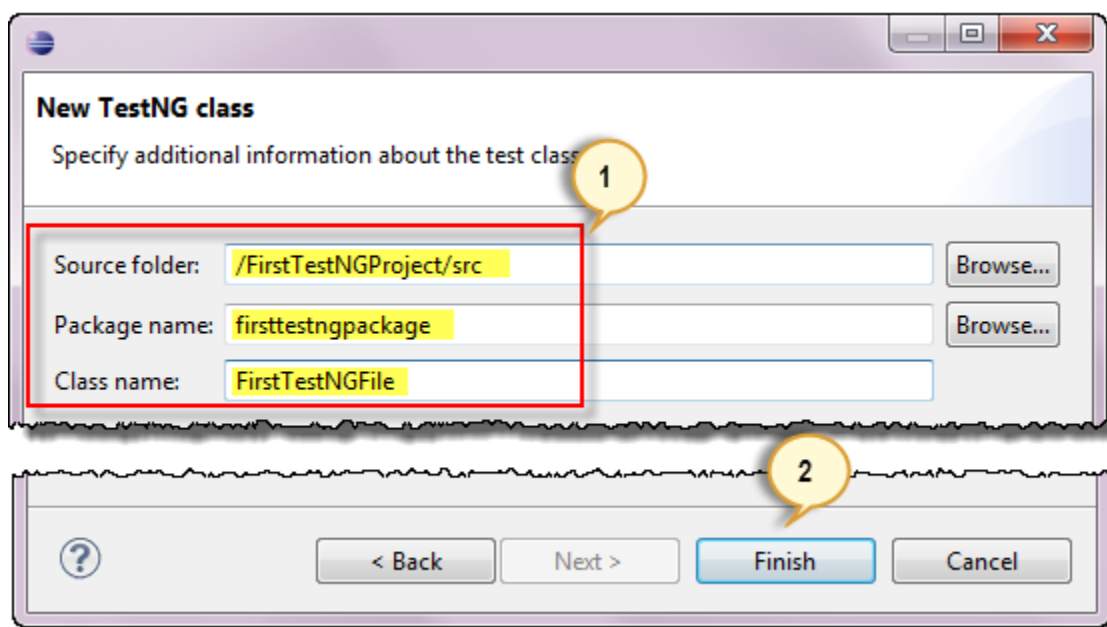


➤ Tạo lớp kiểm thử TestNG





➤ Đặt tên lớp



➤ Kết quả sau khi tạo thành công lớp kiểm thử

```

FirstTestNGFile.java
└─ FirstTestNGProject └─ src └─ firsttestngpackage
1 package firsttestngpackage;
2
3 import org.testng.annotations.Test;
4
5 public class FirstTestNGFile {
6     @Test
7     public void f() {
8     }
9 }
10

```

- Thêm code vào phương thức testcase và chạy thử

```

1 package firsttestngpacket;
2
3 import org.testng.annotations.AfterClass;
4
5 public class FirstTestNGFile {
6     // Test Case 1
7     @Test
8     public void testCase1() {
9         System.out.println("in Test Case 1");
10    }
11 }
12

```

Console window

```

Console
<terminated> FirstTestNGFile [TestNG] C:\Program Files\Java\jre7\bin\javaw.exe (Fel
[TestNG] Running:
  C:\Users\Binatang Mustasa\AppData\Local\Temp\testng-eclipse
ustomsuite.xml

PASSED: verifyHomepageTitle

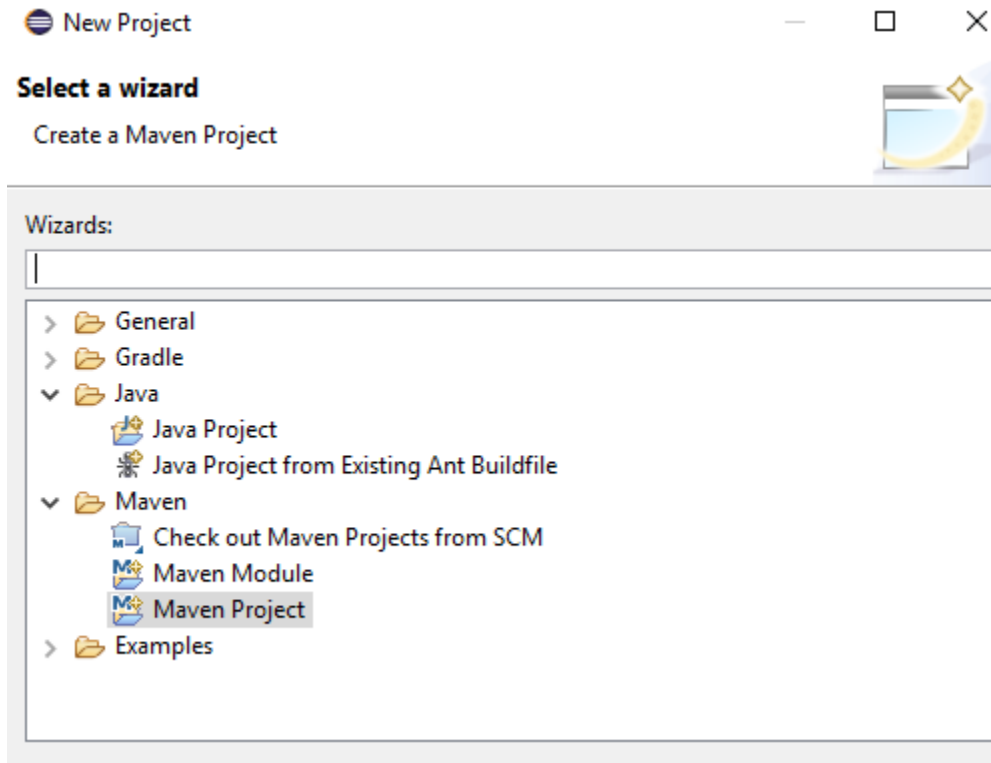
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

```

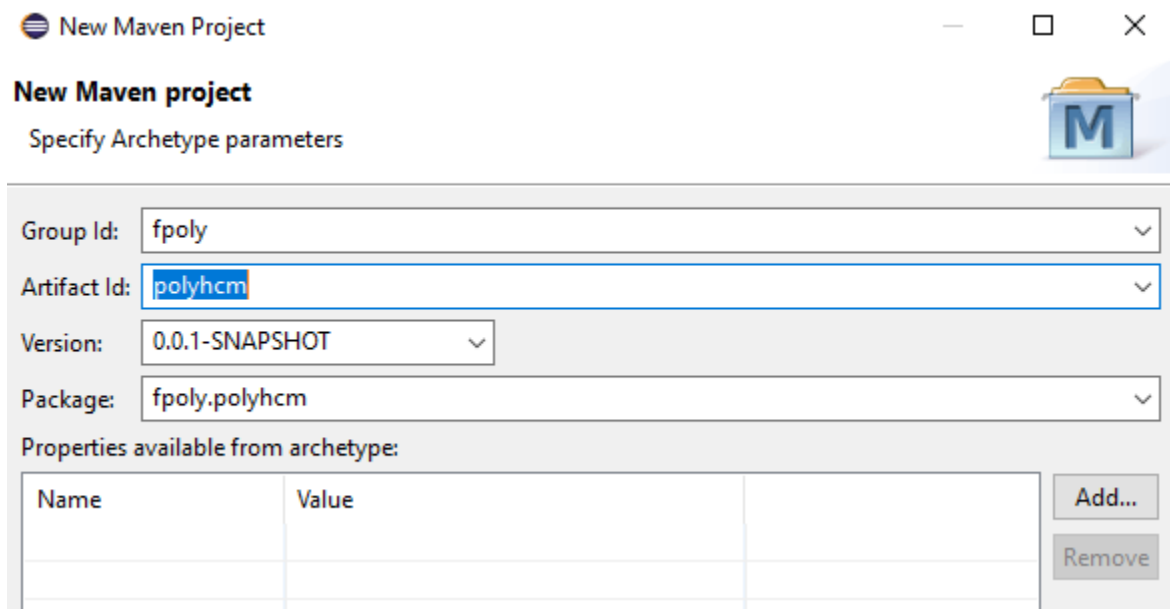
Bài 3 (3 điểm)

Tạo lớp kiểm thử TestNG dùng Maven project

➤ Tạo maven project



➤ Điền các thông tin của id



- Cấu hình tập tin pom.xml giúp nhận diện các thư viện cần dùng trong TestNG

```

17- <dependencies>
18-   <dependency>
19-     <groupId>org.testng</groupId>
20-     <artifactId>testng</artifactId>
21-     <version>6.11</version>
22-     <scope>test</scope>
23-   </dependency>
24- </dependencies>

```

- Tạo lớp có tên VatCalculator, bên trong chứa phương thức tính thuế giá trị gia tăng 10%

```

public class VatCalculator {
    public double getVatOnAmount(double amount){
        return amount * 0.1;
    }
}

```

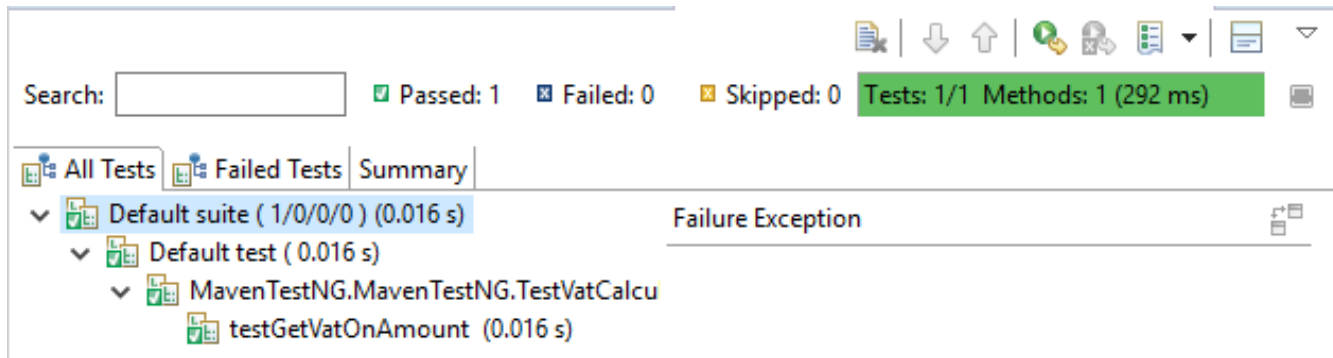
- Tạo lớp kiểm thử tên TestVatCalculator, kiểm thử phương thức tính giá trị gia tăng bên trên chạy có đúng không.

```

2- import org.testng.annotations.Test;
3- import org.testng.Assert;
4- public class TestVatCalculator {
5-     @Test
6-     public void testGetVatOnAmount(){
7-         VatCalculator calc = new VatCalculator();
8-         double expected = 10;
9-         Assert.assertEquals(calc.getVatOnAmount(100), expected);
10-        Assert.assertNotEquals(calc.getVatOnAmount(120), expected);
11-    }
12- }
13

```

- Khi gọi phương thức getVatOnAmount với đối số 100 thì đúng bằng giá trị mong đợi expected (100 x 10 = 10), tương tự cho giá trị 120. Kết quả test pass



The screenshot shows a test runner interface with the following elements:

- Search:** A text input field.
- Status:** Passed: 1, Failed: 0, Skipped: 0.
- Summary:** Tests: 1/1, Methods: 1 (292 ms).
- Test Tree:**
 - Default suite (1/0/0/0) (0.016 s)
 - Default test (0.016 s)
 - MavenTestNG.MavenTestNG.TestVatCalcu
 - testGetVatOnAmount (0.016 s)

Bài 4 (2 điểm)

Giảng viên cho thêm