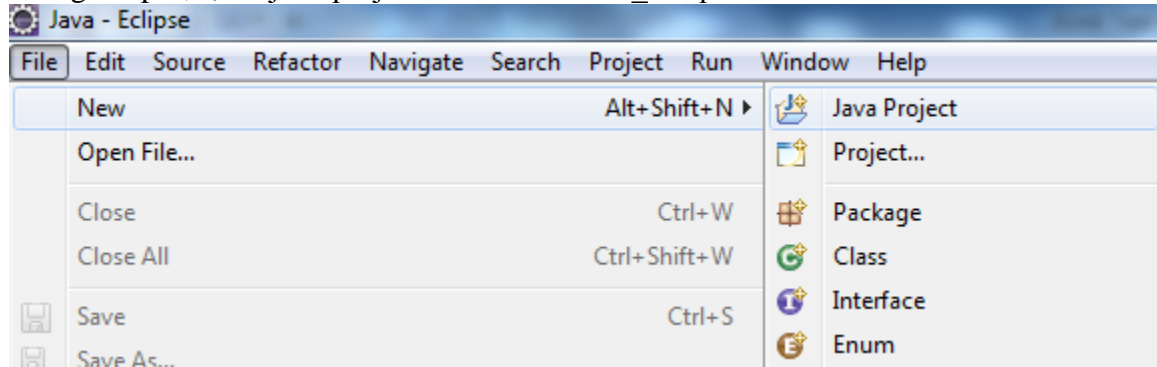



## JUNIT TEST VỚI ECLIPSE

Viết phương thức kiểm tra độ dài 3 cạnh có hợp thành 1 tam giác không ? nếu có đó là tam giác gì?  
Sinh các Testcases dựa vào phương pháp lập bảng quyết định.

Trước hết ta viết chương trình kiểm tra tam giác như sau:

Trong eclipse, tạo 1 java project có tên JunitTest\_Eclipse như hình:



 New Java Project

---

**Create a Java Project**

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:

**JRE**

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'jdk1.6.0\_16') [Configure JREs...](#)

**Project layout**


☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

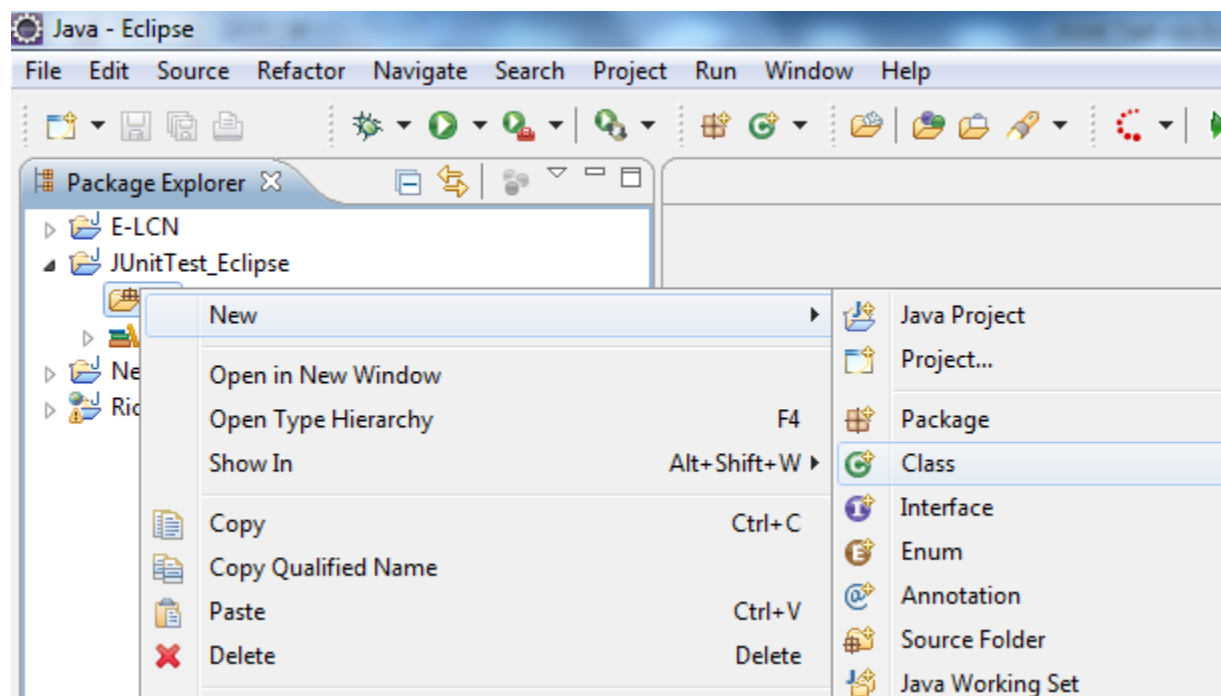
**Working sets**

☐ Add project to working sets

Working sets:



Nhấn Finish  
Thêm 1 lớp mới



Đặt tên là Triangle

**New Java Class**

Create a new Java class.

Source folder: JUnitTest\_Eclipse/src Browse...

Package: vovanhai.wordpress.com Browse...

☐ Enclosing type: Browse...

Name: Triangle

Modifiers: ☒ public ☐ default ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...  
Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

? Finish Cancel

Tiến hành code cho lớp Triangle này như sau:

```
package vovanhai.wordpress.com;

public class Triangle {
    private int a;
    private int b;
    private int c;

    public Triangle(int a, int b, int c) throws Exception{
        if(a<=0||b<=0||c<=0)
            throw new Exception("dữ liệu không hợp lệ");
        this.a = a;
        this.b = b;
        this.c = c;
    }
}
```

```

/**
 * Hàm kiểm tra 1 có phải là tam giác không. Nếu có là loại tam giác gì?
 * @param a
 * @param b
 * @param c
 * @return -1:không phải tam giác
 *         0:tam giác thường
 *         1:tam giác cân
 *         2:tam giác đều
 */
public int CheckTriangle(){
    boolean isTriangle=false;
    if(a<b+c&& b<a+c&&c<a+b)
        isTriangle=true;
    if(isTriangle){
        if(a==b&&b==c)
            return 2;//tam giác đều
        else if(a!=b&&a!=c&&b!=c)
            return 0;//tam giác thường
        else
            return 1;//tam giác cân
    }
    return -1;//không phải tam giác
}
}

```

Bây giờ ta tìm các testcases dựa vào phương pháp bảng quyết định  
 Lập bảng quyết định:

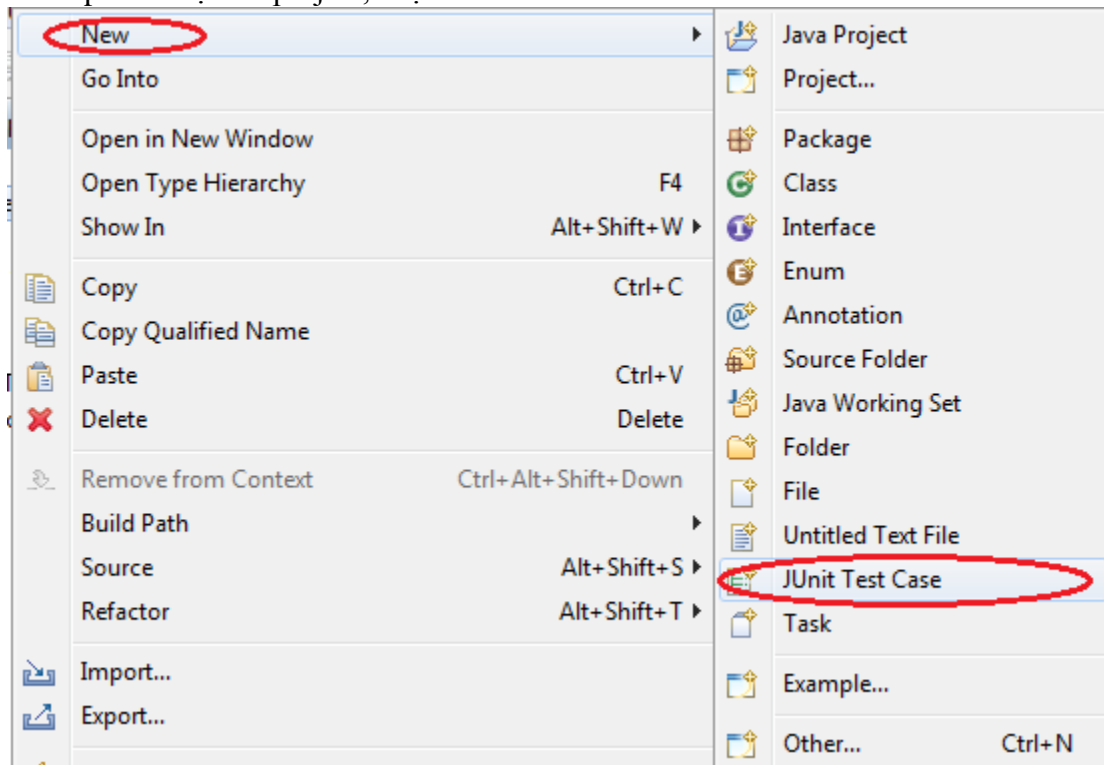
	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8	Rule 9	Rule 10	Rule 11
C1: $a < b + c$ ?	F	T	T	T	T	T	T	T	T	T	T
C2: $b < a + c$ ?	–	F	T	T	T	T	T	T	T	T	T
C3: $c < a + b$ ?	–	–	F	T	T	T	T	T	T	T	T
C4: $a = b$ ?	–	–	–	T	T	T	T	F	F	F	F
C5: $a = c$ ?	–	–	–	T	T	F	F	T	T	F	F
C6: $b = c$ ?	–	–	–	T	F	T	F	T	F	T	F
A1: Not a triangle	X	X	X								
A2: Scalene											X
A3: Isosceles							X		X	X	
A4: Equilateral				X							
A5: Impossible					X	X		X			

Từ bảng quyết định, ta có thể dẫn xuất các testcase như sau:

Test case	a	b	c	Expected output
1	4	1	2	Không phải tam giác
2	1	4	2	Không phải tam giác
3	1	2	4	Không phải tam giác
4	5	5	5	Tam giác đều
5	?	?	?	Không khả dĩ
6	?	?	?	Không khả dĩ
7	2	2	3	Tam giác cân
8	?	?	?	Không khả dĩ
9	2	3	2	Tam giác cân
10	3	2	2	Tam giác cân
11	3	4	5	Tam giác thường

Bây giờ ta tiến hành tạo các unit test.

Nhấn phải chuột lên project, chọn New->JUnit Test Case



Ta được

**New JUnit Test Case**

Select the name of the new JUnit test case. You have the options to specify the class under test and on the next page, to select methods to be tested.

☐ New JUnit 3 test ☒ **New JUnit 4 test**

Source folder: JUnitTest\_Eclipse/src Browse...

Package: **vovanhai.wordpress.com.test** Browse...

Name: **Triange\_Test**

Superclass: java.lang.Object Browse...

Which method stubs would you like to create?

☐ setUpBeforeClass() ☐ tearDownAfterClass()  
☐ setUp() ☐ tearDown()  
☐ constructor

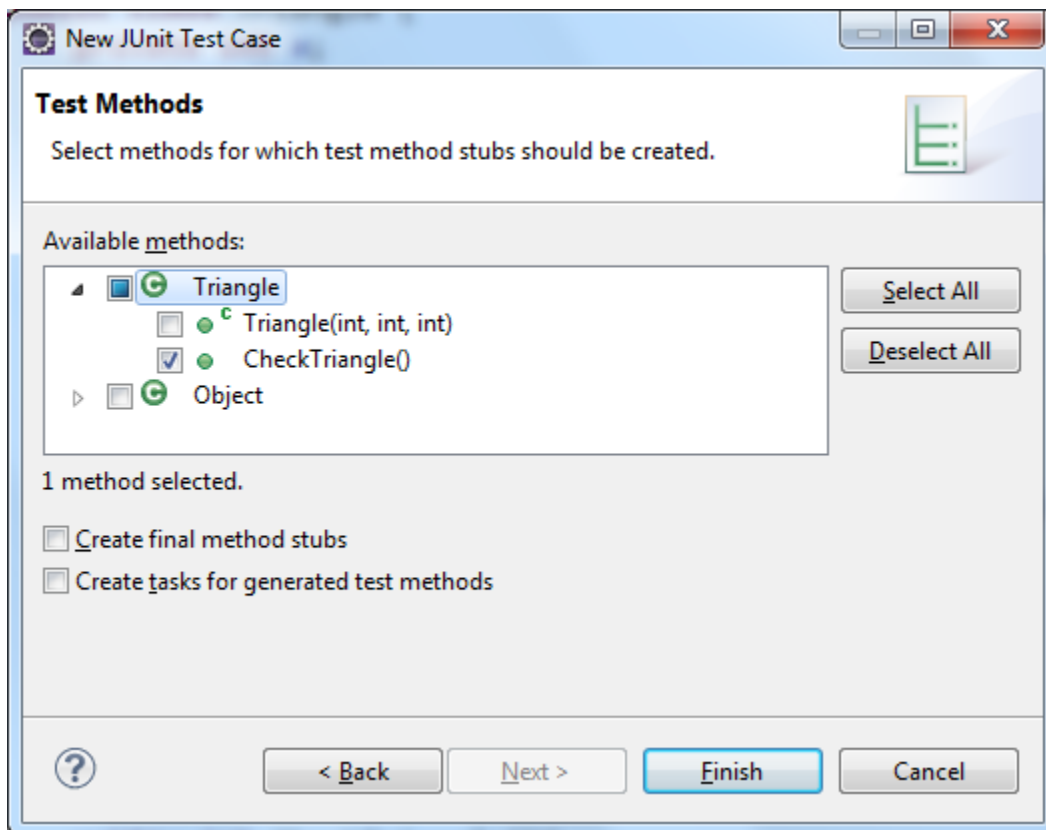
Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

Class under test: **vovanhai.wordpress.com.Triangle** Browse...

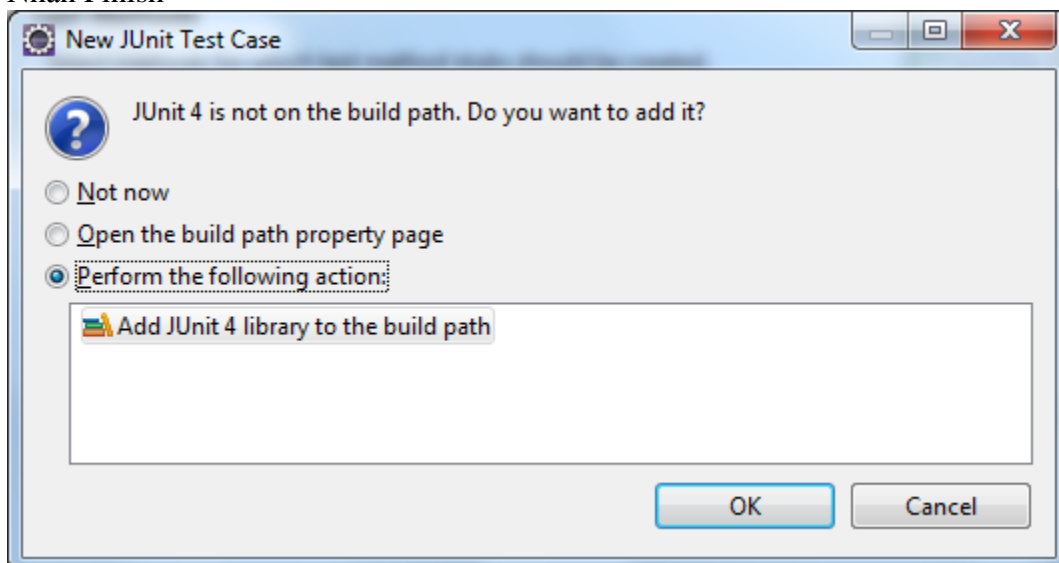
? < Back Next > Finish Cancel

Chú ý ở đây ta dùng JUnit version 4x (dùng anotation thay vì extends từ lớp TestCase như ở JUnit version 3x).

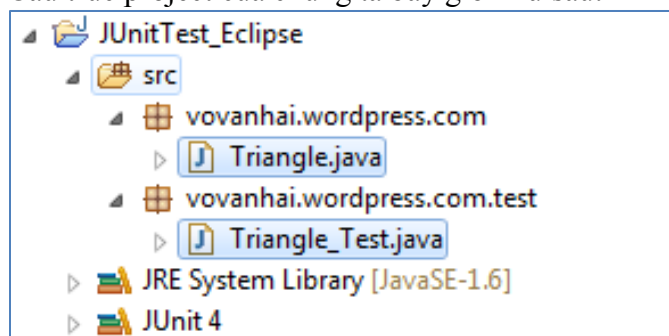
Nhấn Next



Nhấn Finish



Cấu trúc project của chúng ta bây giờ như sau:



Code của lớp Triangle\_Test như sau:



```
package vovanhai.wordpress.com.test;

import junit.framework.Assert;
import org.junit.Test;
import vovanhai.wordpress.com.Triangle;
public class Triangle_Test {
    @Test
    public void testcase1() throws Exception{
        Triangle t=new Triangle(4, 1, 2);
        int expected=-1;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }

    @Test
    public void testcase2() throws Exception{
        Triangle t=new Triangle(1, 4, 2);
        int expected=-1;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }

    @Test
    public void testcase3() throws Exception{
        Triangle t=new Triangle(1,2, 4);
        int expected=-1;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }

    @Test
    public void testcase4() throws Exception{
        Triangle t=new Triangle(5, 5, 5);
        int expected=2;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }

    @Test
    public void testcase5() throws Exception{
        Triangle t=new Triangle(2, 2, 3);
        int expected=1;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }

    @Test
    public void testcase6() throws Exception{
        Triangle t=new Triangle(2, 3, 2);
        int expected=1;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }

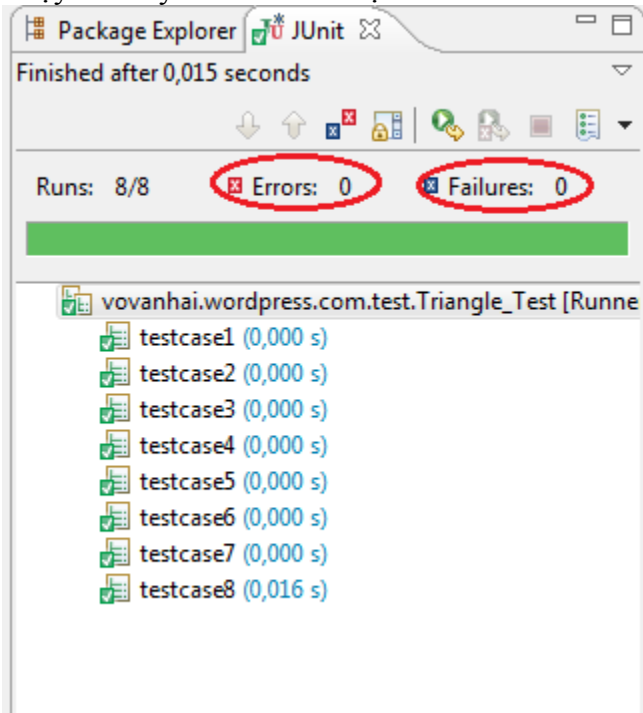
    @Test
    public void testcase7() throws Exception{
        Triangle t=new Triangle(3, 2, 2);
        int expected=1;
    }
}
```

```

        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }
    @Test
    public void testcase8() throws Exception{
        Triangle t=new Triangle(3, 4,5);
        int expected=0;
        int actual=t.CheckTriangle();
        Assert.assertEquals(expected, actual);
    }
}

```

Chạy file này với JUnit ta được:



Giả sử ở testcase 8, ta thay giá trị expected=1, thử chạy lại ta được

