**Báo cáo bài tập kiểm thử**

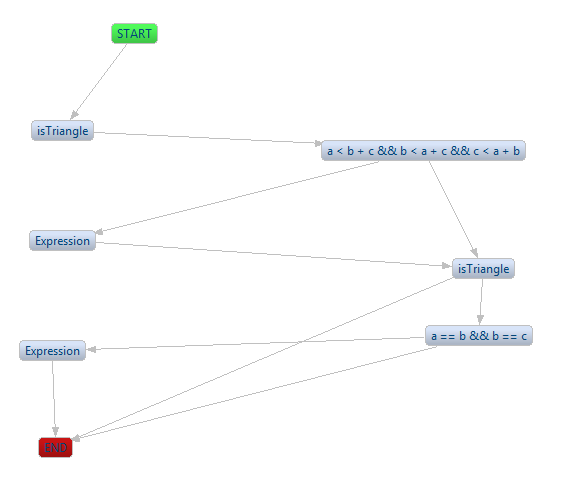
Họ và tên:Nguyễn Văn Trường

MSV:11020356

### Tool: [Eclipse Flow Chart Generator](http://eclipsefcg.sourceforge.net/installation.html)

**Code**: Kiểm tra tam giác

1.Flow Chart



2.Code chương trình

**package** truong.lee.uet;

**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;

}

**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

}

**public** String Print(){

String print="";

**if**(CheckTriangle() == -1){

print="NotATriangle";

}

**if**(CheckTriangle() == 0){

print="Scalene";

}

**if**(CheckTriangle() == 1){

print="Isosceles";

}

**if**(CheckTriangle() == 2){

print="Equilateral";

}

**return** print;

}

}

3.Code Test chương trình

package truong.lee.uet.test;

import org.junit.Assert;

import org.junit.Test;

import truong.lee.uet.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);

System.out.println(t.Print());

}

@Test

public void testcase2() throws Exception {

Triangle t = new Triangle(1, 4, 2);

int expected = -1;

int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

@Test

public void testcase3() throws Exception {

Triangle t = new Triangle(1, 2, 4);

int expected = -1;

int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

@Test

public void testcase4() throws Exception {

Triangle t = new Triangle(5, 5, 5);

int expected = 2;

int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

@Test

public void testcase5() throws Exception {

Triangle t = new Triangle(2, 2, 3);

int expected = 1;

int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

@Test

public void testcase6() throws Exception {

Triangle t = new Triangle(2, 3, 2);

int expected = 1;

int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

@Test

public void testcase7() throws Exception {

Triangle t = new Triangle(3, 2, 2);

int expected = 1;

int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

@Test

public void testcase8() throws Exception {

Triangle t = new Triangle(3, 4, 5);

int expected = 0;

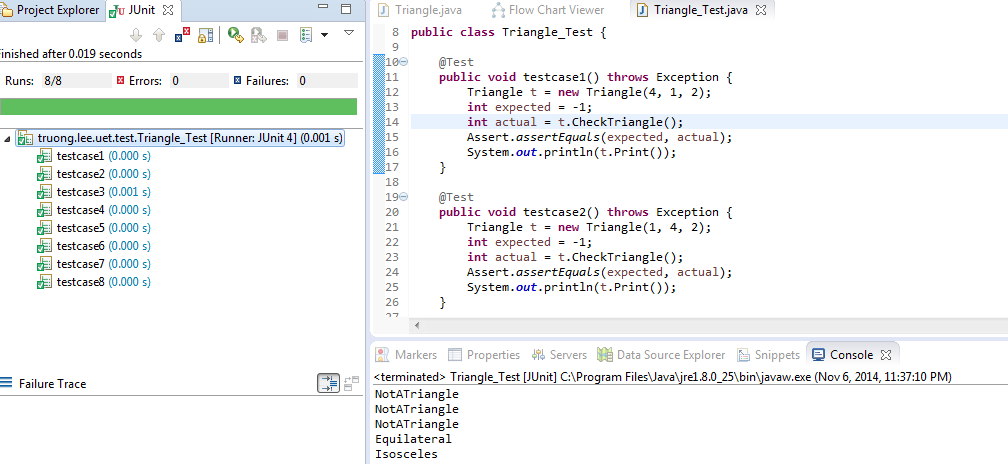
int actual = t.CheckTriangle();

Assert.assertEquals(expected, actual);

System.out.println(t.Print());

}

}



4.Sinh ca kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c |  |
| 1 | 2 | 4 | 1 11 12 |
| 2 | 2 | 5 | 1 2 3 5 6 |
| 5 | 5 | 5 | 1 2 3 7 8 |
| 3 | 5 | 6 | 1 2 8 10 |

F

7

9

6

8

10

5

12

4

3

11

Khong la tam giac

T

Tam giac can

Tam giac deu

Tam giac thuong

End

2

1

a=b||b=c||a=c

a!=b &&b!=c &&a!=c

a=b&&b=c

a<b+c && b<a+c && c<a+b

Nhap a,b,c

Start