****

**软件测试技术第一次实验报告**



**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 2016**

**姓 名 宋琦琪**

**2019年 3 月 12 日**

# 软件测试技术第一次实验报告

1. 需求分析（描述具体需求）
2. 用Eclipse安装JUnit(4.12)、Hamcrest(1.3)
3. 用Eclipse安装Eclemma
4. 编写三角形问题的java程序，并使用JUnit测试程序。

三角形问题的描述：

你口袋里有一张五十元，一张二十元，两张五元钞票和三枚一元硬币。写一个程序，看看你是否能拿出一个给定的数字(X)元。

1. 概要设计（简单描述设计思路，配合UML图）

安装:

1. Junit和Hamcrest的安装和使用

(1)[下载junit.jar和hamcrest-core.jar](https://github.com/junit-team/junit/wiki/Download-and-Install" \t "https://www.cnblogs.com/yongheng20/p/_blank)；

(2)建立Java项目，在Eclipse菜单栏选择项目（Project）-> 属性 -> Java构建路径 -> 库 -> 添加外部JAR，找到刚刚下载的junit.jar和hamcrest-core.jar，添加

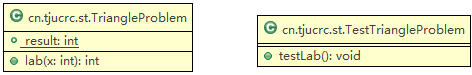
2. Eclemma的安装和使用

(1)打开Eclipse，选择菜单栏“帮助” -> Eclipse Marketplace，搜索“Eclemma”，

(2)点击安装即可。

三角问题：

1. 根据面值从大到小判断需要的张数
2. 从50开始，如果50元需要一张以上，则不能拿出给定的金额，如50元需要1张或不需要，则将金额x对50取余，并继续计算后面的情况，以此类推。
3. 测试用例中输入x值以及预期结果



1. 详细设计（详细描述具体如何实现，附代码及说明）

代码:

**package** cn.tjucrc.st;

**public** **class** TriangleProblem {

**public** **static** **int** *result*;

**public** **int** lab(**int** x){ //只考虑测试数据为整数 没有写存在小数情况

**if** (x/50 > 1)

*result* = 0;//需要多于1张50元 不符合条件

**else** **if** (x/50 <= 1){

x = x % 50;//将取余后的数赋给新的x 继续计算

**if** (x/20 >1)

*result* = 0;//需要多于1张20元 不符合条件

**else** **if** (x/20 <= 1){

x = x % 20;

**if** (x/5 >2)

*result* = 0;//需要多余2张5元 不符合条件

**else** **if** (x/5 <= 2){

x = x % 5;

**if** (x > 3)

*result* = 0;//需要多余3张1元 不符合条件

**else**

*result* = 1;

}

}

}

**return** *result*;

}

}

测试代码:

**package** cn.tjucrc.st;

**import** **static** org.junit.Assert.*assertEquals*;

**import** org.junit.Test;

**public** **class** TestTriangleProblem {

@Test

**public** **void** testLab(){

TriangleProblem Tri;

Tri = **new** TriangleProblem();

*assertEquals*(0,Tri.lab(149)); //测试 如果金额为149 期望结果应为0

}

}

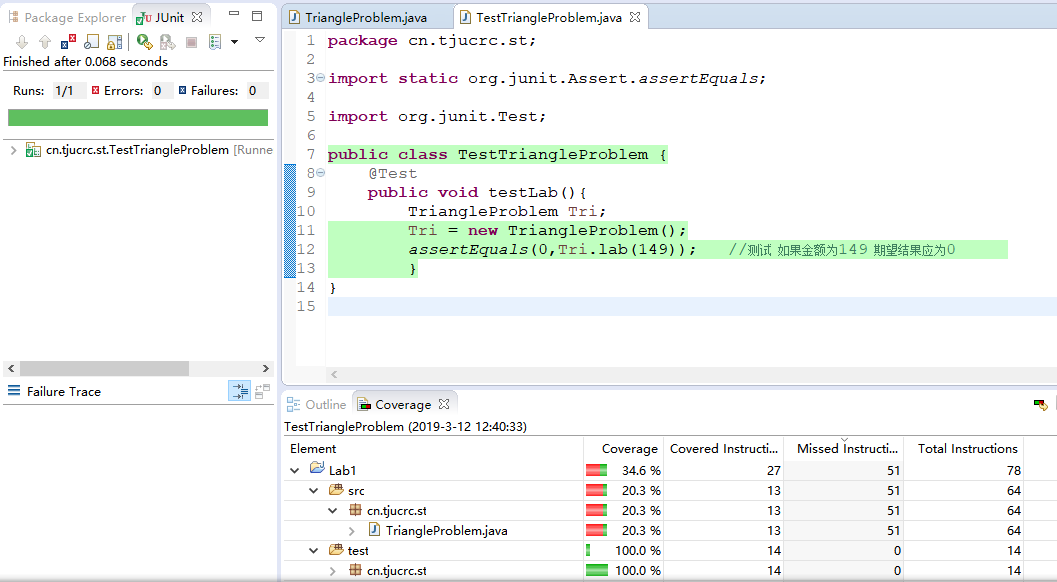
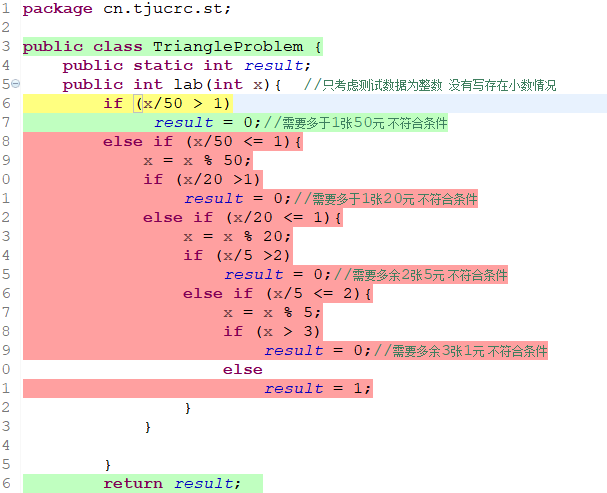
1. 调试分析（在实验过程中遇到的问题以及如何解决）
2. 最开始将lab函数设为了void，并且代码中判断能否拿出给定的数字使用的是bool值

问题：在测试用例中assertEquals（）函数不支持（boolean，void）格式

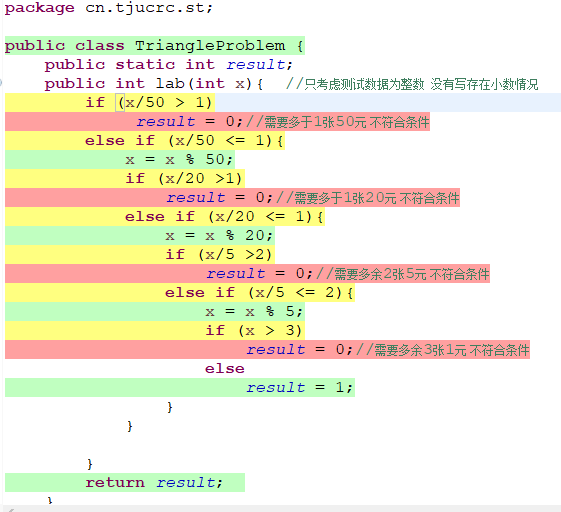
解决：之后对lab函数和判断方式进行了修改，都改成了int，问题得以解决。

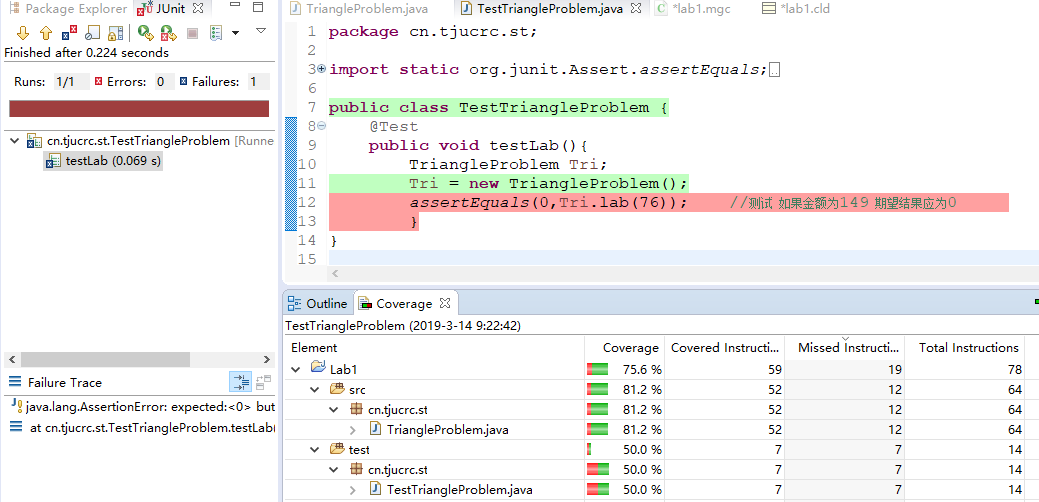
1. 测试结果（描述输入和输出）

测试1：*assertEquals*(0,Tri.lab(149));



测试2：*assertEquals*(0,Tri.lab(76));





1. 总结

第一次通过代码的方式测试程序，测试的过程就是找错的过程，要通过方法去找到软件中存在的问题，这才是软件测试的意义。