软件测试上机报告



第1次上机作业 – Junit & Eclemma

**学 院 智能与计算学部**

**专    业 软件工程**

**姓 名 白文杰**

**学 号 3018216031**

**年 级 2018**

**班    级 1**

#### Experimental requirements

1. Install Junit(4.12), Hamcrest(1.3) with Eclipse/IDEA
2. Install Eclemma with Eclipse
3. Write a java program for the given problem and test the program with Junit.
   1. Description of the problem:

There are one 50 yuan, one 20 yuan, one 10 yuan, two 5 yuan bills and three 1 yuan coins in your pocket. Write a program to find out whether you can take out a given number (x) yuan.

#### Configuration

1. java code and junit test program

代码的实现（java源程序）类是CanTakeOut.java

测试类是TestLab1.java

详细代码见Part 4

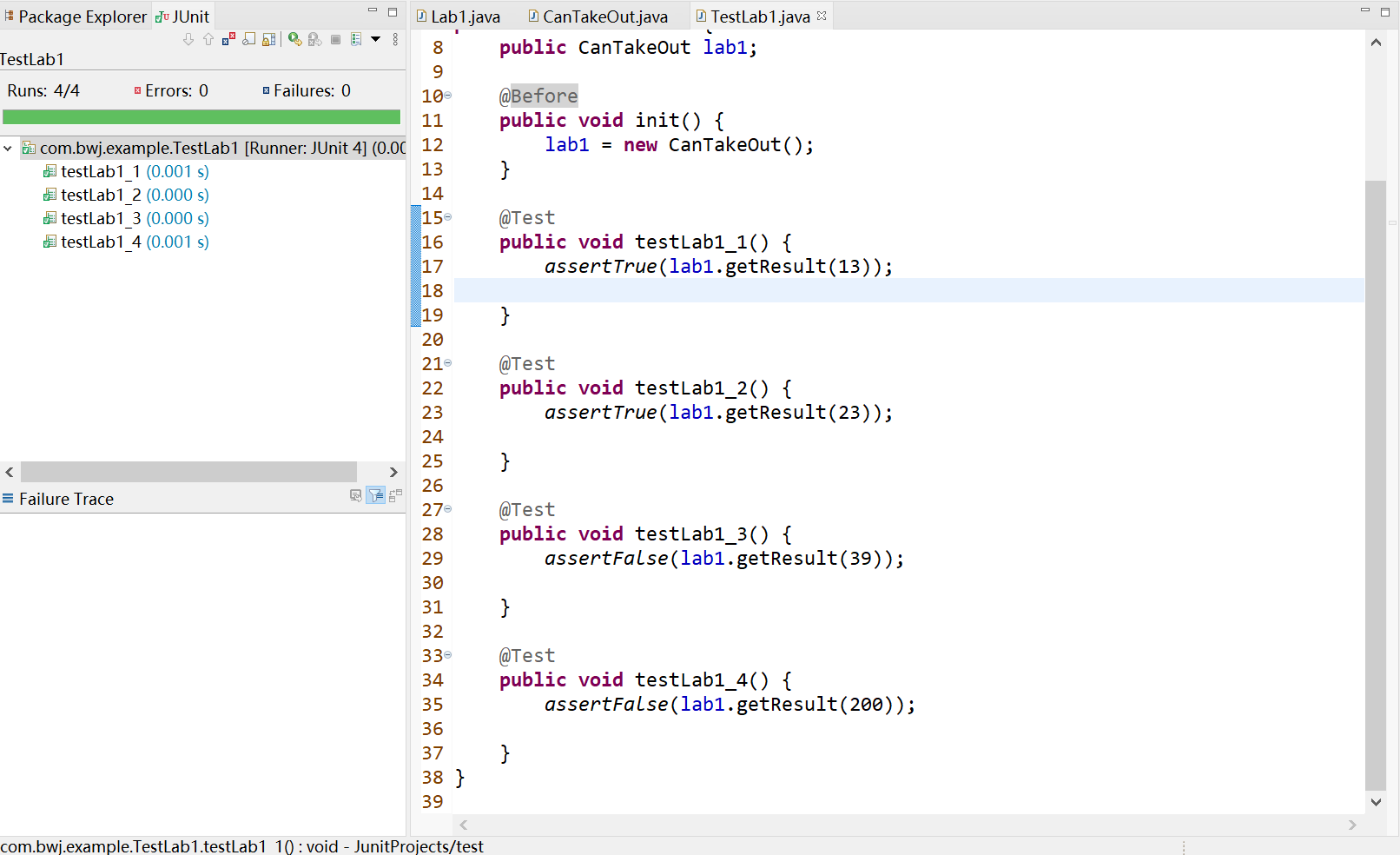
1. The brief description of installing junit, hamcrest and eclemma.  
   安装这三者的过程是简单的，因为实验材料中已经包含了所有的依赖。

对于Junit和Hamcrest，我的做法是：将实验材料中提供的jar包放在项目的lib目录下，打开eclipse，右键jar包，点击Build Path，将依赖加入项目中。

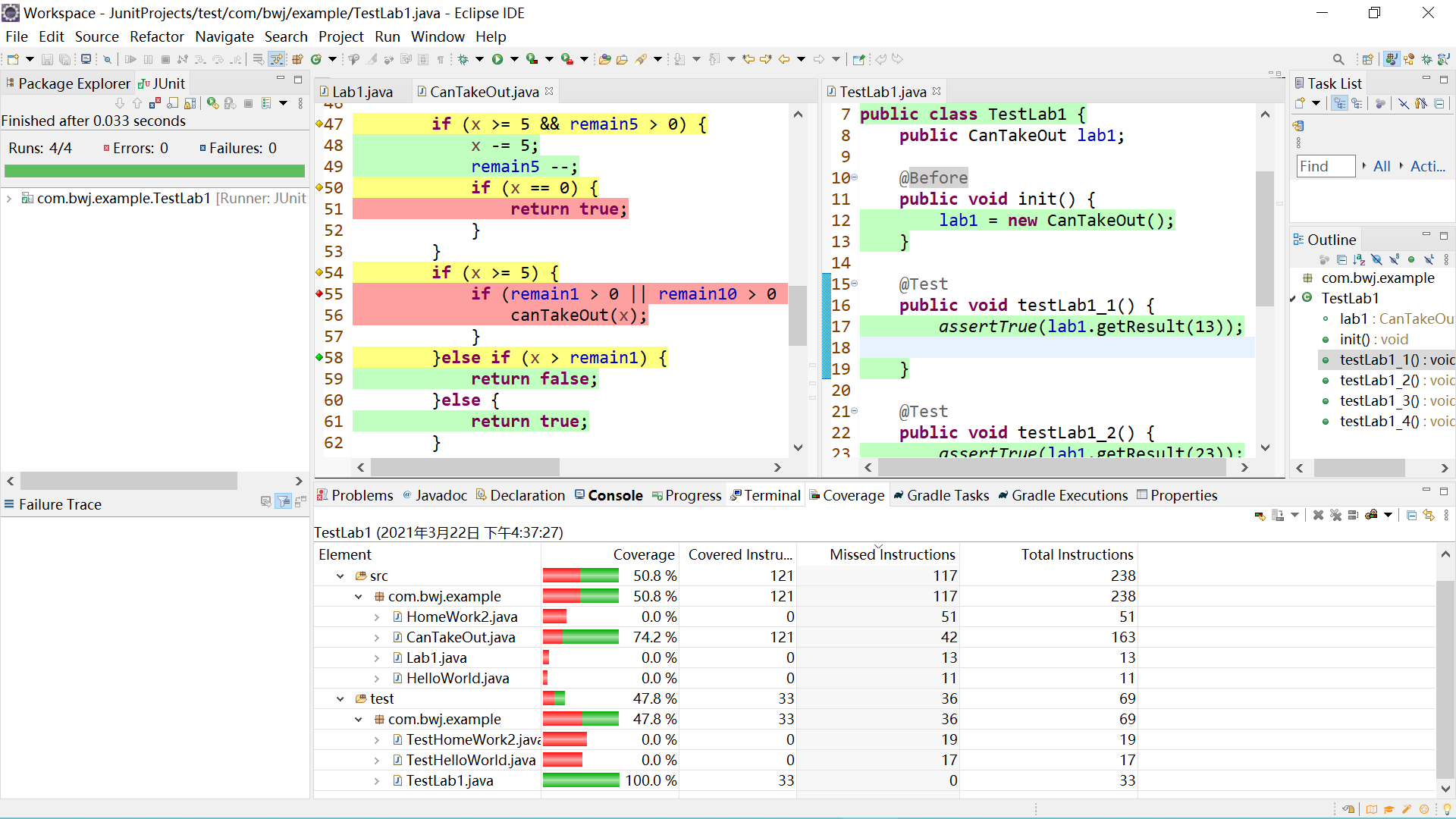
对于eclemma，我查阅了网上一个访问量较多的帖子，将实验材料中压缩包解压在：eclipse安装目录\dropins下，然后重启eclipse。  
（但是还是不好用，我又点击了：Help->Install New Software ->add->local->选择解压的文件夹安装）

1. The test result and coverage report (print screen) of your tests on the problem.

Test Result:



Coverage Report:



#### Result analysis

从测试结果中可以看到，给出的四个测试用例都通过了测试。

但是覆盖率却不高，约为74.2%。出现这一现象的主要原因是测试用例过少，没有充分的测试到。

#### Source code

CanTakeOut.java

|  |
| --- |
| package com**.**bwj**.**example**;**  public class CanTakeOut **{**  final int numOf50 **=** 1**;**  final int numOf20 **=** 1**;**  final int numOf10 **=** 1**;**  final int numOf5 **=** 2**;**  final int numOf1 **=** 3**;**    int remain50**;**  int remain20**;**  int remain10**;**  int remain5**;**  int remain1**;**    int x**;**    public CanTakeOut**()** **{**  reset**();**  **}**  private boolean canTakeOut**(**int x**)** **{**  **if** **(**x **>=** 50 **&&** remain50 **>** 0**)** **{**  x **-=** 50**;**  remain50 **--;**  **if** **(**x **==** 0**)** **{**  **return** **true;**  **}**  **}**    **if** **(**x **>=** 20 **&&** remain20 **>** 0**)** **{**  x **-=** 20**;**  remain20 **--;**  **if** **(**x **==** 0**)** **{**  **return** **true;**  **}**  **}**    **if** **(**x **>=** 10 **&&** remain10 **>** 0**)** **{**  x **-=** 10**;**  remain10 **--;**  **if** **(**x **==** 0**)** **{**  **return** **true;**  **}**  **}**    **if** **(**x **>=** 5 **&&** remain5 **>** 0**)** **{**  x **-=** 5**;**  remain5 **--;**  **if** **(**x **==** 0**)** **{**  **return** **true;**  **}**  **}**  **if** **(**x **>=** 5**)** **{**  **if** **(**remain1 **>** 0 **||** remain10 **>** 0 **||** remain20 **>** 0 **||** remain5 **>** 0 **||** remain50 **>** 0**)** **{**  canTakeOut**(**x**);**  **}**  **}else** **if** **(**x **>** remain1**)** **{**  **return** **false;**  **}else** **{**  **return** **true;**  **}**      **return** **false;**  **}**    public boolean getResult**(**int a**)** **{**  **this.**x **=** a**;**  int maxValue **=** 50 **\*** numOf50 **+** 20 **\*** numOf20 **+** 10 **\*** numOf10 **+** 5 **\*** numOf5 **+** numOf1**;**  int minValue **=** 0**;**  **if** **(**x **>** maxValue **||** x **<** minValue**)** **{**  **return** **false;**  **}**  boolean tmp **=** canTakeOut**(this.**x**);**  reset**();**  **return** tmp**;**  **}**    public void reset**()** **{**  **this.**remain50 **=** **this.**numOf50**;**  **this.**remain20 **=** **this.**numOf20**;**  **this.**remain10 **=** **this.**numOf10**;**  **this.**remain5 **=** **this.**numOf5**;**  **this.**remain1 **=** **this.**numOf1**;**  **}**  **}** |

TestLab1.java

|  |
| --- |
| package com**.**bwj**.**example**;**  **import** static org**.**junit**.**Assert**.\*;**  **import** org**.**junit**.\*;**  public class TestLab1 **{**  public CanTakeOut lab1**;**    @Before  public void init**()** **{**  lab1 **=** **new** CanTakeOut**();**  **}**    @Test  public void testLab1\_1**()** **{**  assertTrue**(**lab1**.**getResult**(**13**));**    **}**    @Test  public void testLab1\_2**()** **{**  assertTrue**(**lab1**.**getResult**(**23**));**    **}**    @Test  public void testLab1\_3**()** **{**  assertFalse**(**lab1**.**getResult**(**39**));**    **}**    @Test  public void testLab1\_4**()** **{**  assertFalse**(**lab1**.**getResult**(**200**));**    **}**  **}** |