**软件测试上机报告**

****

第一次上机作业

**学 院\_\_\_智能与计算学部\_\_\_\_\_\_**

**专 业\_\_\_\_ 软件工程\_\_\_\_\_\_\_\_\_\_**

**姓 名\_\_\_\_\_\_\_宋高超\_\_\_ \_\_\_\_**

**学 号\_\_\_\_\_3017214075\_\_\_\_\_\_\_\_**

**年 级\_\_\_\_\_\_\_ 2017\_\_\_\_ \_\_**

**班 级\_\_\_\_\_\_\_ 2\_\_ \_\_\_\_\_**

# 实验要求

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 is 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.

# 源代码

**Tool.java如下**

**package** experience.demo01;

**public** **class** Tool {

**public** **static** **boolean** findout(**int** n) {

**int**[] money= {50,20,10,5,5,1,1,1};

**int**[] ans=**new** **int**[8];

**for**(**int** i=0;i<8;i++) {

**if**(n>=money[i]) {

ans[i]=1;

n-=money[i];

}

}

**if**(n>0)**return** **false**;

**else** **return** **true**;

}

}

**TestTool.java如下**

**package** experiencetest.demo01test;

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

**import** **static** org.hamcrest.Matchers.\*;

**import** org.junit.Test;

**import** experience.demo01.Tool;

**public** **class** TestTool {

@Test

**public** **void** testFindout() {

// fail("Not yet implemented");

Tool tool=**new** Tool();

**boolean** ans1=Tool.*findout*(2);

**boolean** ans2=Tool.*findout*(4);

*assertThat*(ans1,*is*(**true**));

*assertThat*(ans2,*is*(**false**));

}

}

# 运行结果

* 1. The brief description that I install junit, hamcrest and eclemma.

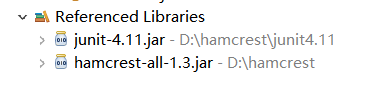
When I installed the junit, hamcrest, eclemma, firstly I put the eclemma files in the file of

eclipse-jee-oxygen-R-win32-x86\_64\eclipse\dropins\plugins

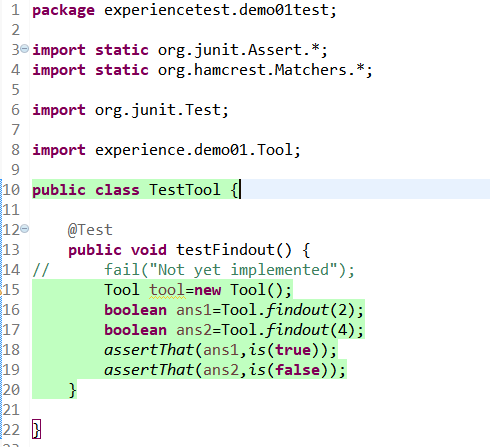
To make sure it can be used when I test the software.

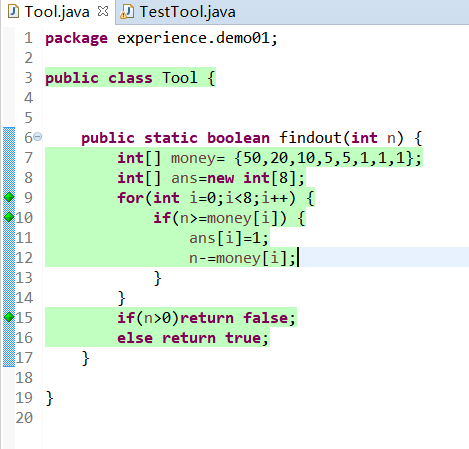
When I finished designing the test programme, I choose

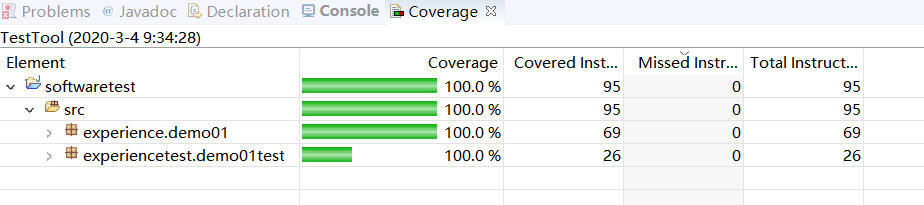
Add external archives to include junit and hamcrest in my project.



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







It is easily to see that my programme is 100% covered. Here is the result.

