软件测试上机报告

第一次上机作业 –Lab 1 Junit and Eclemma

学　　院　智能与计算学部

专　　业　软件工程

姓　　名　郎文翀

学　　号　**3019244247**

年　　级　2019 级

班　　级　软件工程 5 班

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.
   a) Description of the problem:

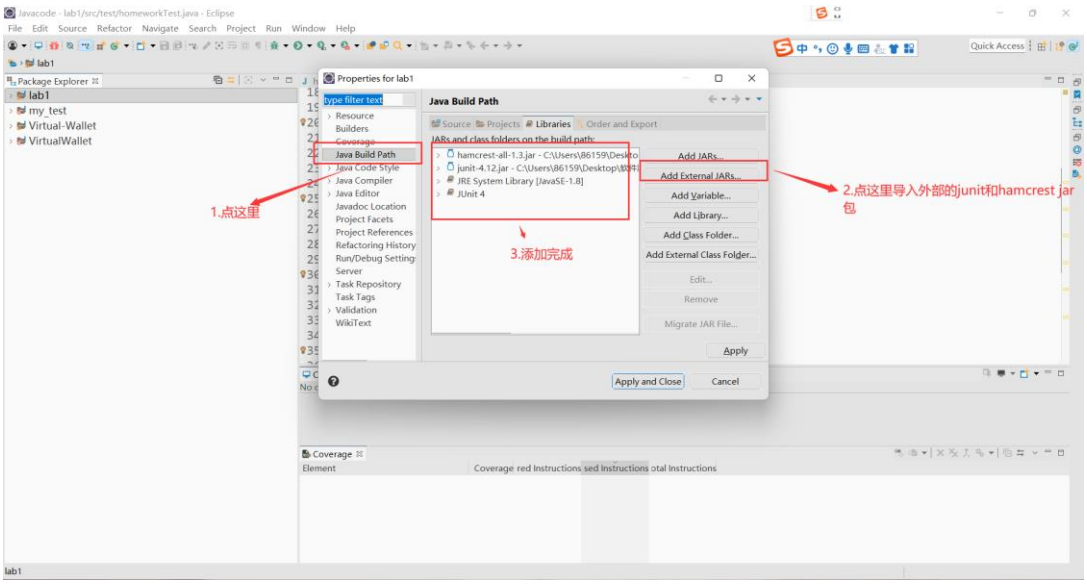
      There are one 50 yuan, one 20 yuan, one 10 yuan, two 5 yuan bills and three 1 yuan coins in your pocket. Write a program to find out whether you can take out a given number (x) yuan.

2. Configuration

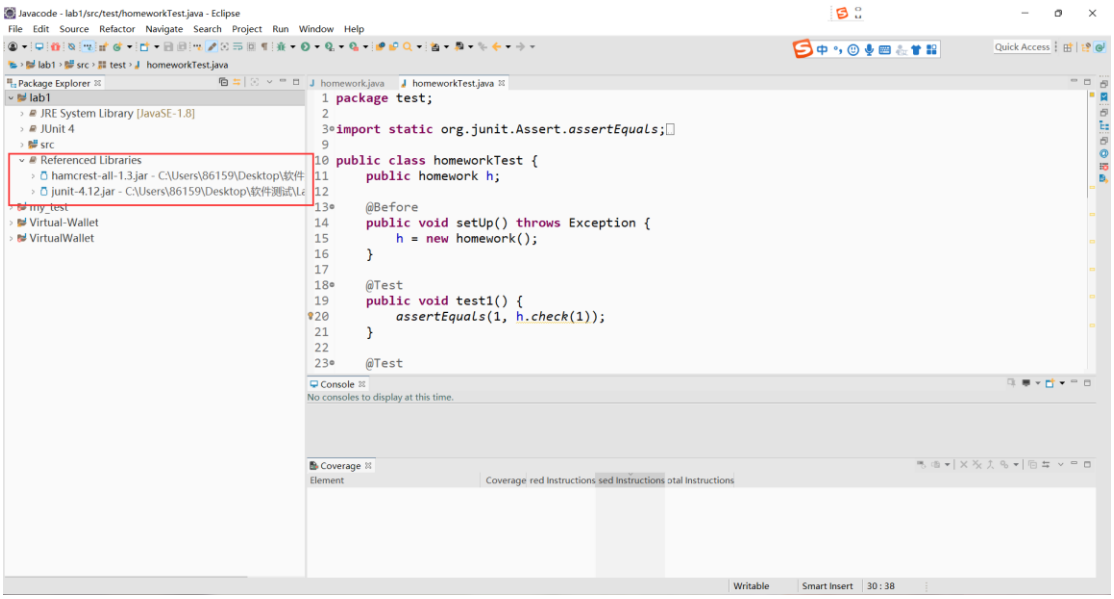
我在 idea 和 eclipse 中都进行了 junit 测试，首先演示 eclipse 然后再演示 idea 中 junit 测试。

2.1 Eclipse 导入 junit、Hamcrest

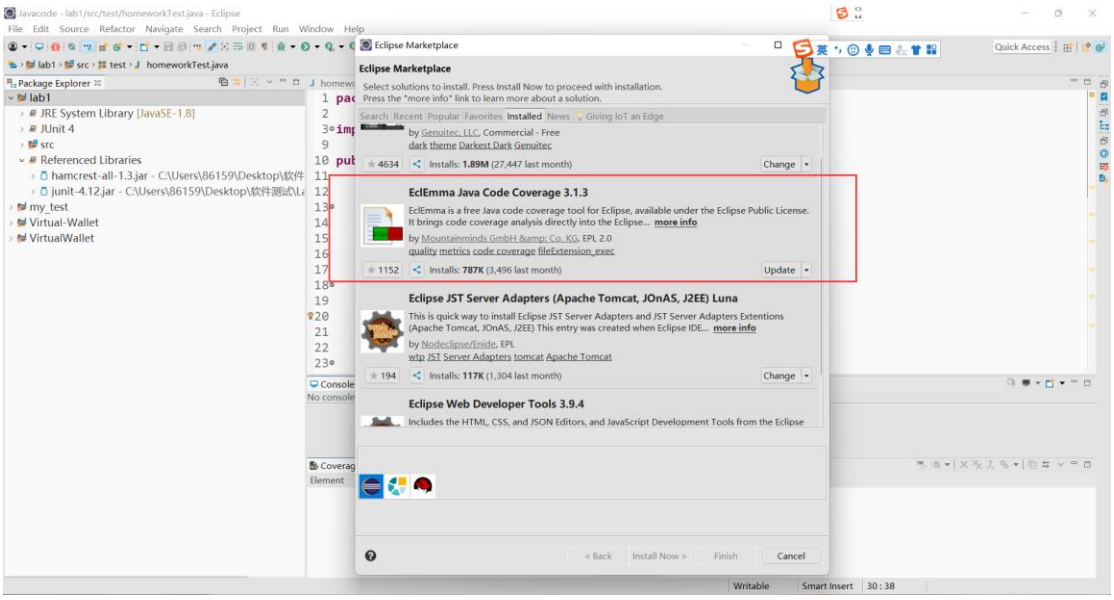
首先我们需要新建一个 java 项目，我这里命名为了 lab1,，然后右键新建的项目，选择属性，在弹出的窗口左侧中选择 Java Build Path，然后点击右侧的 Add external jars 导入外部的 juni jar 包如下图所示：



导入成功以后我们就可以在 referance lib 中看到导入的 jar 包了如下所示：



然后我们还需要进一步安装 Eclemma，这里老师给了压缩包供我们安装，但是实际上也可以直接使用 eclipse 商店提供的插件，只需要点击 Help->Eclipse Marketplace 既可以进入商店搜索 hamcrest 进行安装：

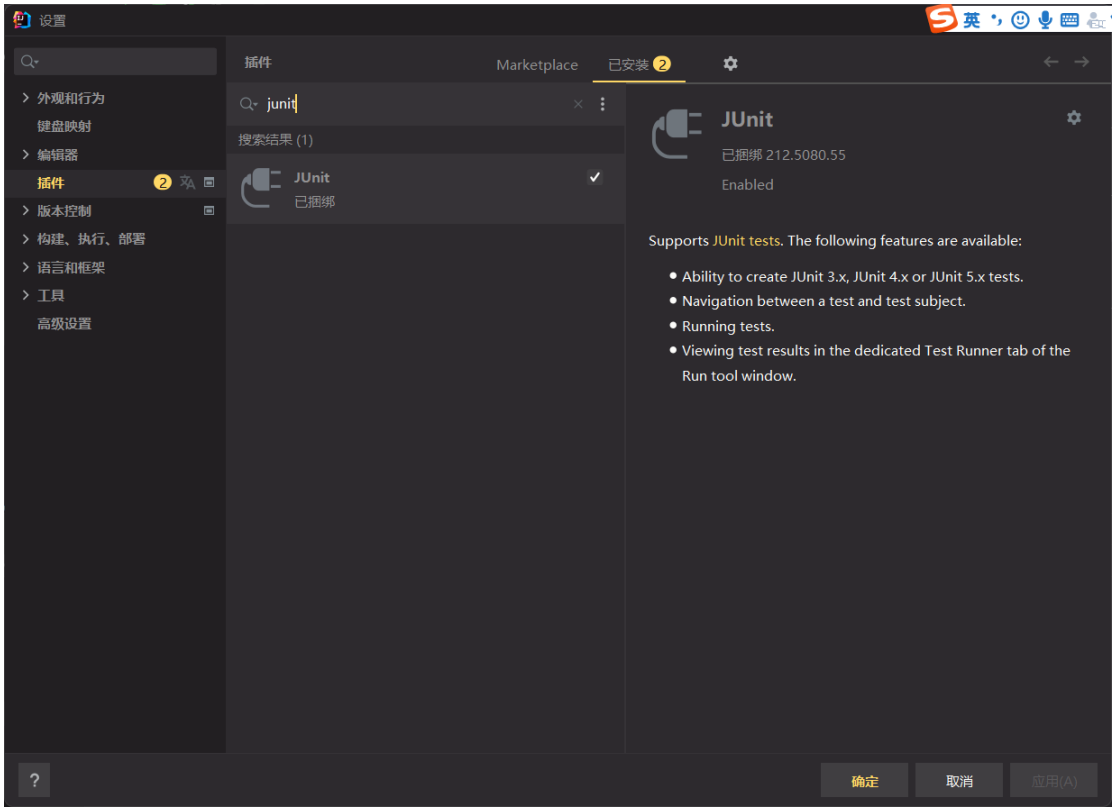


安装完成以后我们就可以看到在运行按钮旁边多出了 coverage 按钮如下图所示：

自此我们就成功完成了 eclipse 上 junit 测试所需要的环境配置。

2.2 iDea 导入 junit 测试

这个就比 eclipse 简单许多了，我们只需要打卡文件->设置->插件商店搜索 junit 进行安装即可了如下所示：



可以看到已经捆绑安装了此插件，那么我们就已经可以在 idea 上使用 junit 测试了。

3. Result analysis

3.1 测试之前的代码编写

首先我们再测试之前要编写需要测试的代码以及用来进行测试的代码，为了结构清晰，我分别将需要被测试的代码 homework.java 放到了 lab/src/main 下，将测试代码 homeworkTest.java 放到了 lab/src/test 中。

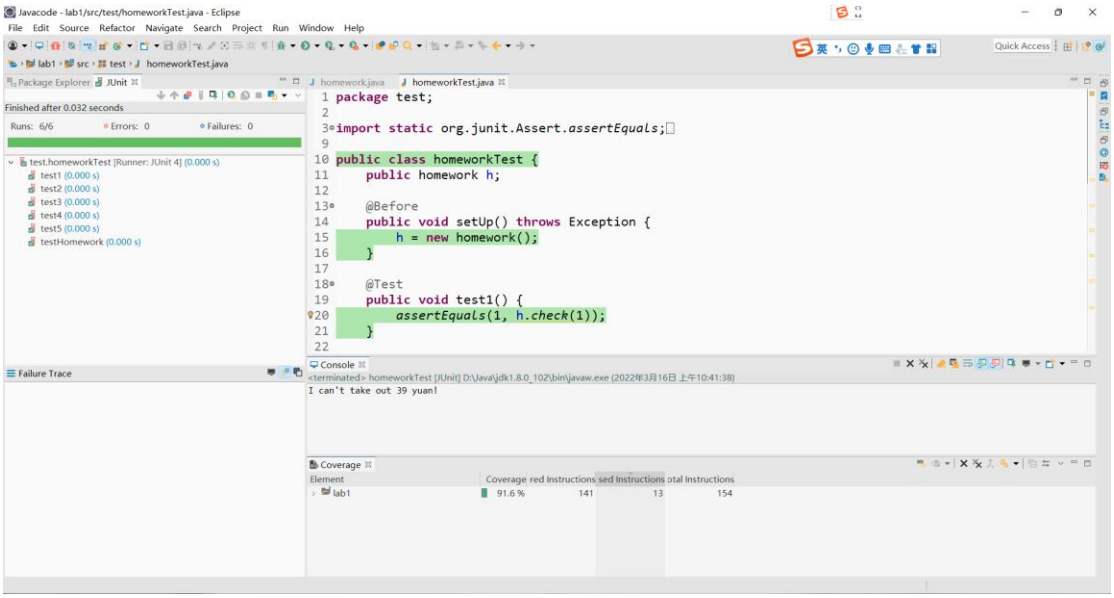
审题发现就是编写一个能够凑出给定数字的函数，我们用最简单的暴力枚举即可进行判断，具体的代码参考 4。编写完成被测试的代码以后，我们还需要进行测试代码的编写，这里我们需要引入@Test 注解和 assertEquals 函数用来进行断言判断，因此会引入 junit 库。

为了尽可能发现代码的 bug，我进行了 6 组测试，test1-test5 分别测试了 5 中不同的数字能够得到预期结果，同时还使用了 testHomework 函数来测试能否正常输出结果。

我们还要注意，无论是哪一个测试，首先我们需要有一个 homework 实例对象，才能进行测试，因此我们需要借用@Before 注解在每一个测试之前首先初始化一个 homework 类的实例对象，为了能够及时发现异常，使用 Exception 类接收异常信息。自此我们就完成了代码的编写，具体代码参考 4.。

3.2 eclipse 进行代码测试

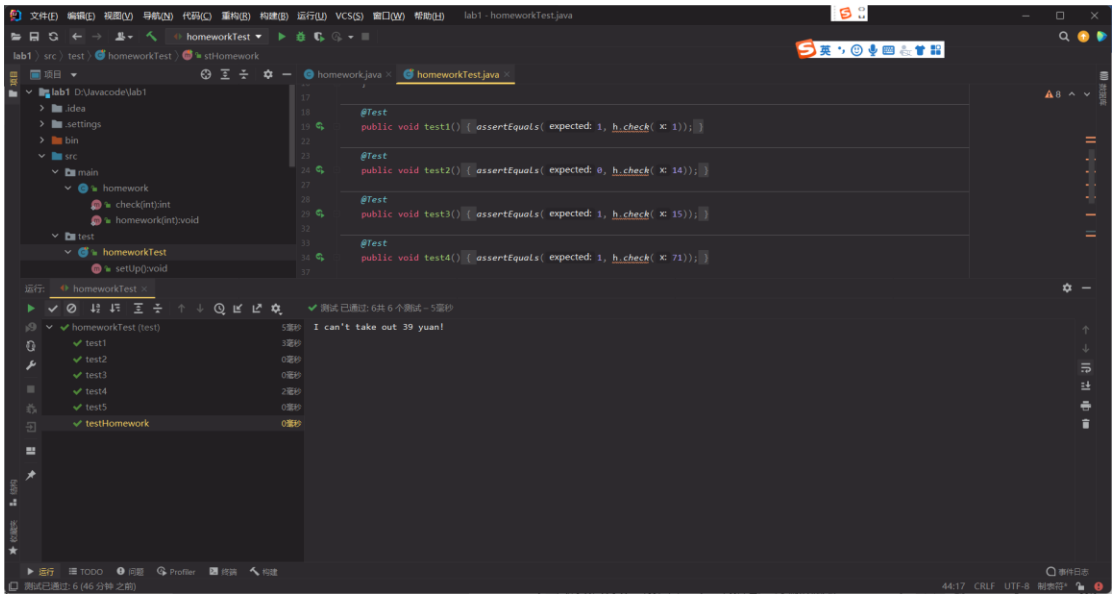
首先我们尝试在 eclipse 上测试，我们只需要右键 homeworkTest.java 选择 converage as 中的 junit4 即可触发测试，然后会得到如下的结果展示：



可以看到所有的测试代码都有被绿色标注说明全部执行了测试并且均得到预期的结果，说明测试并未发现代码 bug。同时对于 homeworkTest 我们也得到了预期的输出。自此我们就完成了 junit 测试，这里我们还要注意所有的测试方法都要是 public 方法（小坑）。

3.4 idea 进行代码测试

同样的我们再在 idea 中尝试进行 junit 测试，操作略有区别，在 idea 中我们需要在 homework.java 中点击 shift+insert 打卡快捷工具栏，然后选择测试->junit4 即可，然后我们就可以看到类似的结果了：



可以看到所有的测试都有绿色对勾说明全部执行了测试并且所有的测试都得到了预期的结果，没有发现代码在执行过程中出现失败。

4. Source code

这里我再给出实验中使用到的代码，首先是被测试的代码 lab1/src/main/homework.java

```
package main;

public class homework {
                    public static int check(int x) {
                        int flag = 0;
                        for (int i = 0; i <= 1; i++) {
                            for (int j = 0; j <= 1; j++) {
                                for (int k = 0; k <= 1; k++) {
                                    for (int l = 0; l <= 2; l++) {
                                        for (int m = 0; m <= 3; m++) {
                                            if (50 * i + 20 * j + 10 * k + 5 * l + m
== x) {
                                                flag = 1;
                                                break;
                                            }
                                        }
                                    }
                                }
                            }
                        }
                        return flag;
                    }

                    public static void homework(int x) {
```

```
                if (check(x) == 1) {
                    System.out.println("I can take out " + x + " yuan!");
                } else {
                    System.out.println("I can't take out " + x + " yuan!");
                }
            }
}
```

然后是 lab1/src/test/homeworkTest.java

```
package test;

import static org.junit.Assert.assertEquals;

import org.junit.Before;
import org.junit.Test;

import main.homework;

public class homeworkTest {
    public homework h;

    @Before
    public void setUp() throws Exception {
        h = new homework();
    }

    @Test
    public void test1() {
        assertEquals(1, h.check(1));
    }

    @Test
    public void test2() {
        assertEquals(0, h.check(14));
    }

    @Test
    public void test3() {
        assertEquals(1, h.check(15));
    }

    @Test
    public void test4() {
        assertEquals(1, h.check(71));
    }

    @Test
    public void test5() {
        assertEquals(0, h.check(100));
```

```
        }

        @Test
        public void testHomework() {
            h.homework(39);
        }

}
```

5. 实验心得总结

经过本次实验我第一次尝试在 eclipse 和 idea 中进行了 junit 单元测试，尤其是 idea，在开发过程中，我总是为难以对编写的代码进行及时的测试感到苦恼，经过本次实验我进一步学习在 idea 中 junit 中的各种测试方法，方便我对编写的代码即使进行测试发现 bug 并进行解决。