****

**软件测试技术第一次实验报告**



**学 院 软件学院**

**专 业 软件工程**

**年 级 2016级**

**姓 名 夏子霞**

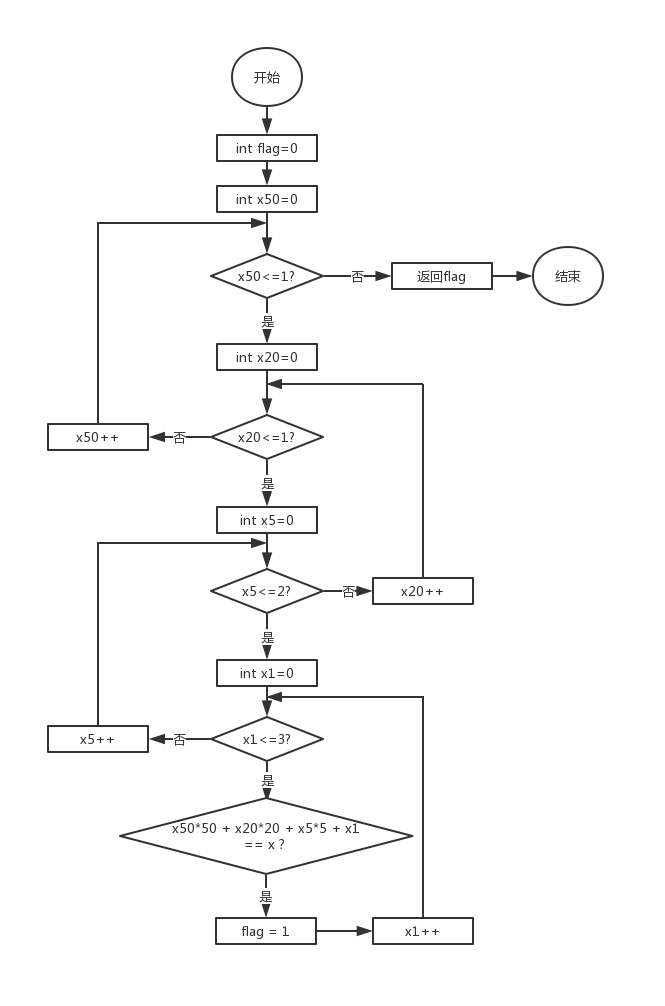
**2019年 3 月 12 日**

# 软件测试技术第一次实验报告

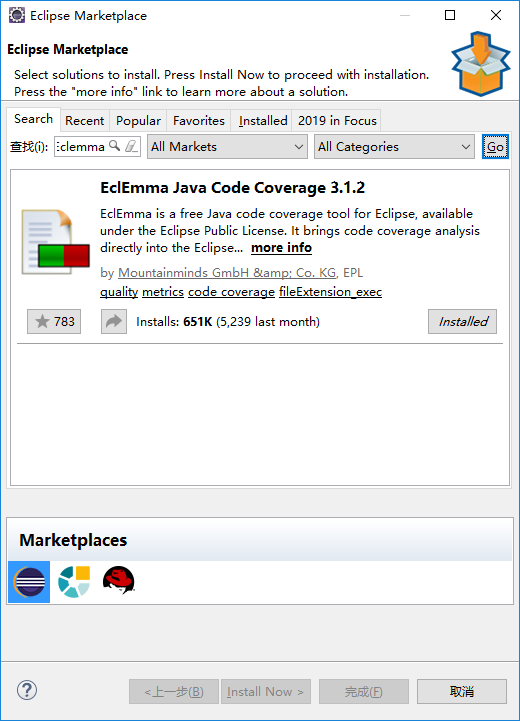
1. 需求分析（描述具体需求）
2. 在Eclipse上安装Junit（4.12），Hamcrest(1.3)
3. 在Eclipse上安装Eclemma
4. 编写程序解决下列问题并用Junit测试该程序

有1张50元、1张20元、2张5元纸币以及3个1元硬币，编写程序证明这些钱是否能组成给定金额x元。

1. 概要设计（简单描述设计思路，配合UML图）
2. 本地添加Junit（4.12），Hamcrest(1.3)的jar包
3. 在Eclipse端查找安装Eclemma
4. 采用穷举法，流程图如下：



1. 详细设计（详细描述具体如何实现，附代码及说明）
2. 网上下载Junit和Hamcrest包文件，保存在本地。新建Java项目，在Eclipse菜单栏选择项目（Project）-> Properties -> JavaBuildPath -> Libraries -> Add External JARs，找到刚刚下载的junit.jar和hamcrest-core.jar，添加。
3. 在Eclipse顶部菜单栏中，选择菜单栏“帮助” -> Eclipse Marketplace，搜索“Eclemma”，点击install。



1. 新建cal.java

package lab1;

public class cal {

public int combination(int x) {

int flag = 0;

for(int x50 = 0; x50 <= 1 ; x50++) {

for(int x20 = 0; x20 <= 1 ; x20++) {

for(int x5 = 0; x5 <= 2 ; x5++) {

for(int x1 = 0; x1 <= 3 ; x1++) {

if(x50\*50 + x20\*20 + x5\*5 + x1 == x) {

flag = 1;

}

}

}

}

}

return flag;

}

}

新建calTest.java

package lab1;

import org.junit.Assert;

import org.junit.Test;

public class calTest {

@Test

public void testCombination() {

cal c = new cal();

int result = c.combination(76);

Assert.assertEquals(result, 1);

}

}

右击项目，选择Coverage As->2 JUnit Test,进行测试。

1. 调试分析（在实验过程中遇到的问题以及如何解决）
2. 如果cal.java程序如下：

package lab1;

public class cal {

public int combination(int x) {

for(int x50 = 0; x50 <= 1 ; x50++) {

for(int x20 = 0; x20 <= 1 ; x20++) {

for(int x5 = 0; x5 <= 2 ; x5++) {

for(int x1 = 0; x1 <= 3 ; x1++) {

if(x50\*50 + x20\*20 + x5\*5 + x1 == x) {

return 1;

}

}

}

}

}

return 0;

}

}

则无法找到覆盖率为100%的测试用例，用flag指示返回值，可以达到测试目的。

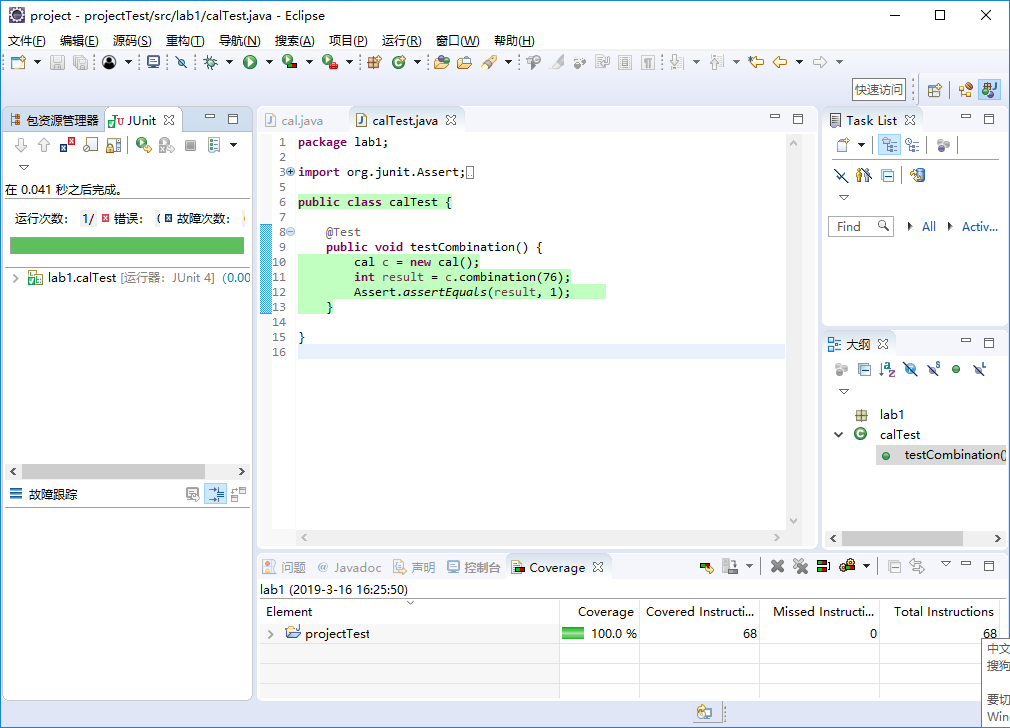
1. 注意每一个for循环括号中x的下标应该是一致的，并且取值范围应该是0到最大值（上下边界均能取到）。
2. 测试结果（描述输入和输出）

(1)

输入：x = 76，expected = 1

输出：flag = 1

测试结果：

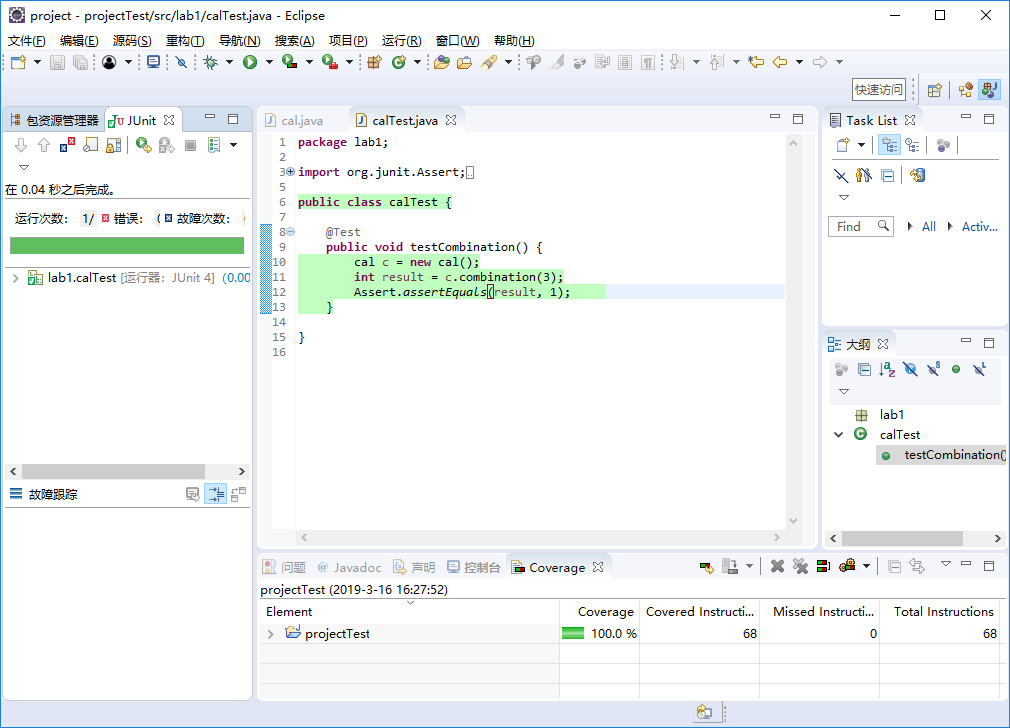


(2)

输入：x = 3，expected = 1

输出：flag = 1

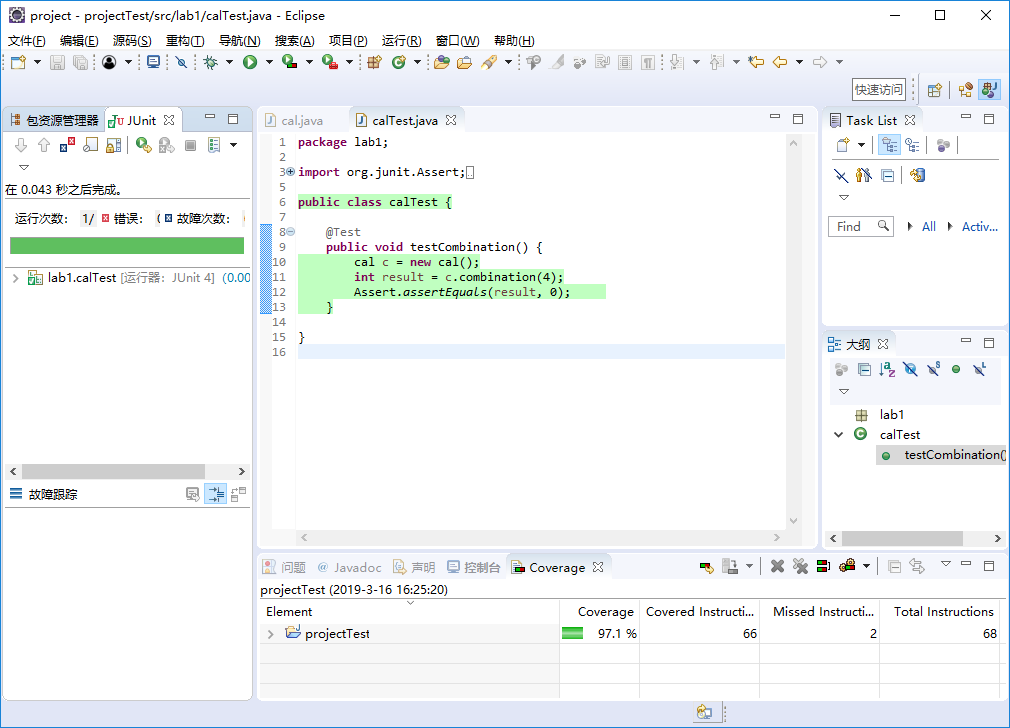
测试结果：

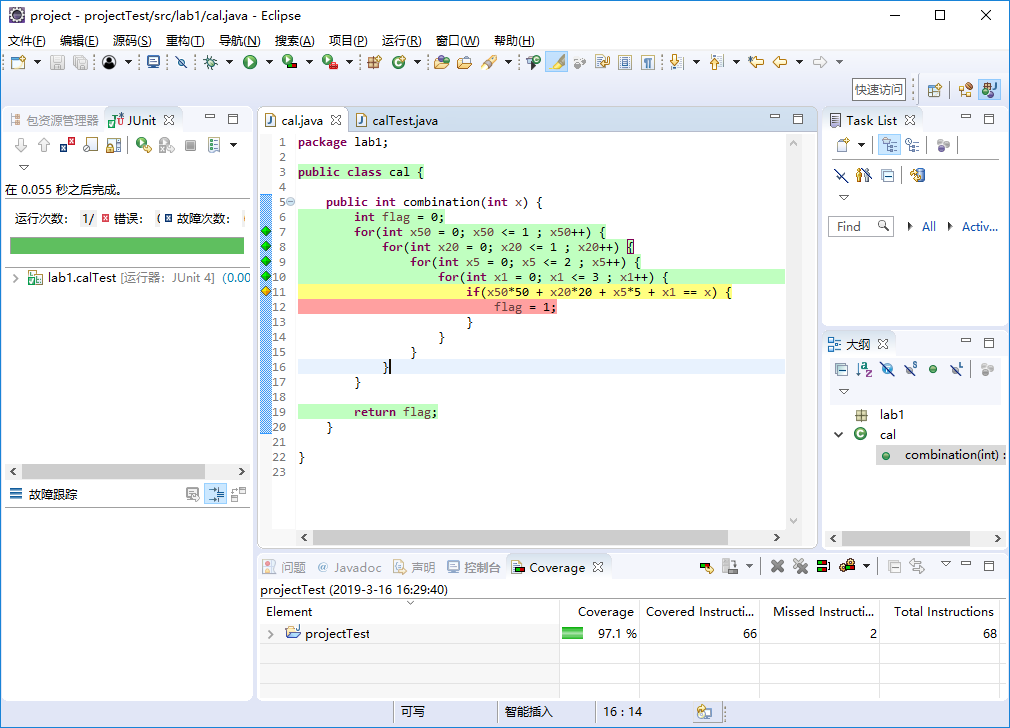


(3)  
输入：x = 4，expected = 0

输出：flag = 0

测试结果：





1. 总结

这次实验我对Junit测试有了初步的了解，并且体会到了Junit语言给软件测试带来的便捷性。除此之外，我还认识到了编程一定要仔细，例如for循环中x的下标问题就是一个深刻的教训。