****

**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级**

**姓 名 王晨**

**2019年 3 月 16 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）

Tasks:

1.Install Junit(4.12), Hamcrest(1.3) with Eclipse

2.Install Eclemma with Eclipse

3.Write a java program for the triangle problem and test the program with Junit.

a)Description of triangle problem:

There is one 50 yuan, one 20 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.

1. 概要设计（简单描述设计思路，配合UML图）

本次实验分为两个类设计，一个是待测试的类，一个是测试类。

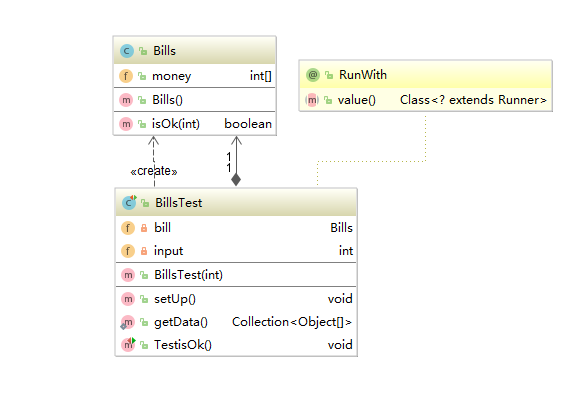
1.待测试类Bills

该类用于存放货币，判断能否成功。

2.测试类

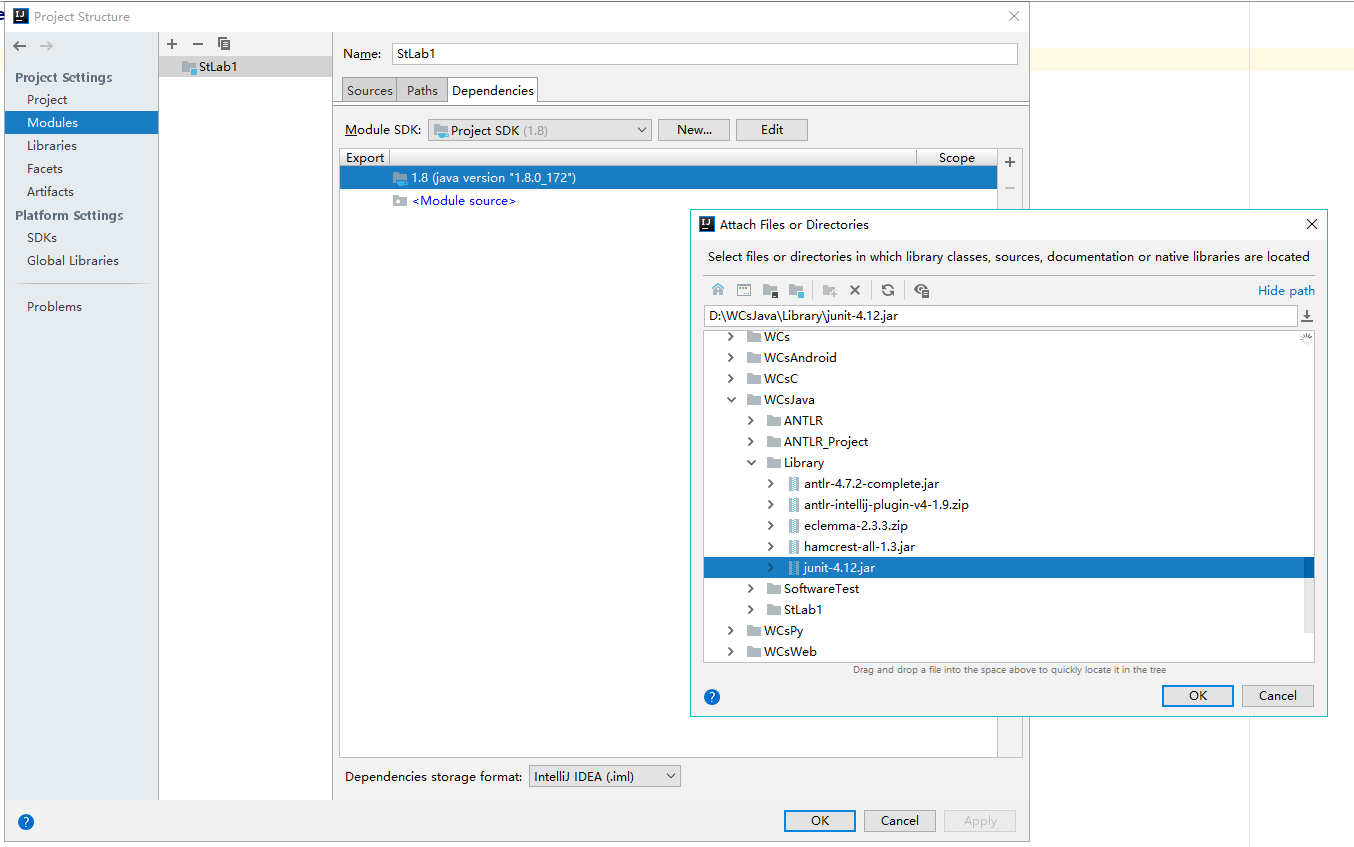
该类用于设计测试用例，并进行测试

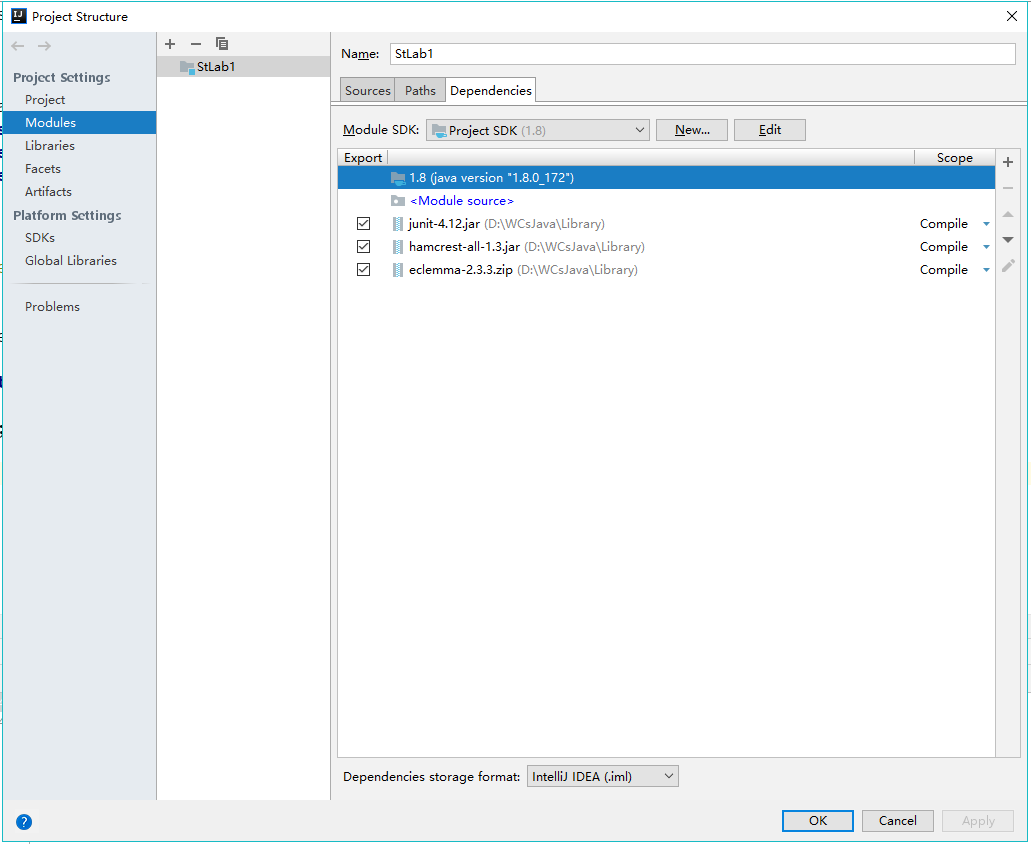
3.UML类图



1. 详细设计（详细描述具体如何实现，附代码及说明）

1.安装junit,hamcrest 和 eclemma的jar包。





2.待测试类的设计

package src;

public class Bills {

public int[] money = {50, 20, 5, 5, 1, 1, 1}; // 初始化货币

public Bills() {

}

public boolean isOk(int x) { // 判断能否成功的函数

for (int i=0; i< money.length; i++) { // 将货币从大到小遍历

if (money[i]<x) { // 如果当前金额比目标小

x = x - money[i]; // 减小目标金额

}

else if (money[i]==x) { // 如果当前金额与目标相同

x = x - money[i];

return true; // 成功

}

else {

continue;

}

}

return false; // 失败

}

}

3.测试类的设计

package test;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.junit.runners.Parameterized;

import src.Bills;

import java.util.Arrays;

import java.util.Collection;

import static junit.framework.TestCase.assertTrue;

@RunWith(Parameterized.class)

public class BillsTest {

private Bills bill;

private int input;

public BillsTest(int input) {

this.input = input;

}

@Before

public void setUp() {

bill = new Bills();

}

@Parameterized.Parameters

public static Collection<Object[]> getData() {

return Arrays.asList(new Object[][]{

{66}, {50}, {0}, {11} // 测试用例

});

}

@Test // 判断是否成功

public void TestisOk() {

assertTrue(bill.isOk(input));

}

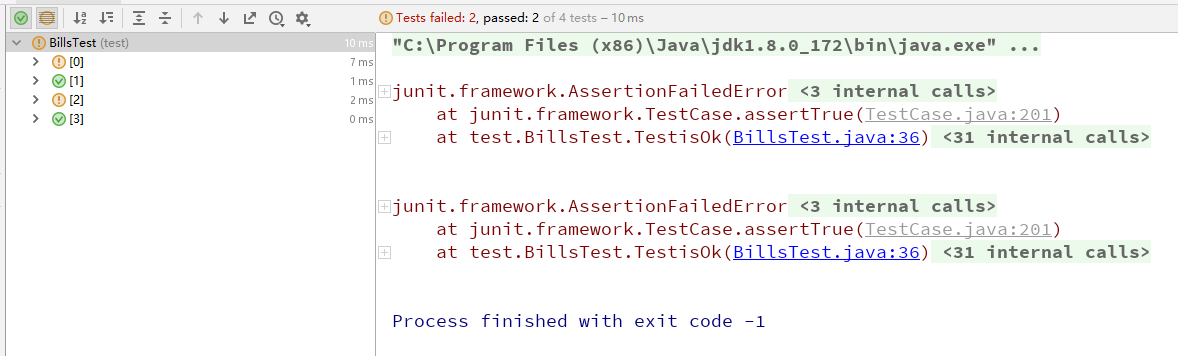
}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

遇到的问题：第一次写程序的时候，是在一个测试类(@Test)中同时运行了多个测试用例，测试用例不能同时测试，报错只显示第一个。

解决方法：使用@Parameterized.Parameters可以同时测试多个测试用例。

1. 测试结果（描述输入和输出）



1. 总结

第一次真正的使用了Junit，感觉功能很强大。学习了软件测试的流程，以及测试用例的设计、测试覆盖原则等知识。