****

**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 16**

**姓 名 邓杰**

**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）

写一个java程序完成问题，并使用junit完成对代码的测试。问题要求如下：

你的口袋里有一张50,一张20和两张5元的纸币，以及三枚1元的硬币。编写一个程序来验证你能否给出规定的钱数（x）。

1. 概要设计（简单描述设计思路，配合UML图）

源程序：

第一步，初始化一个数组录入口袋所有的纸币种类和数量并按从大到小排列

第二步，检查用户输入是否正确，如果超过总金额或者输入为负则提示异常

第三步，运用循环逐步将需要支付金钱的顺序排列出来；

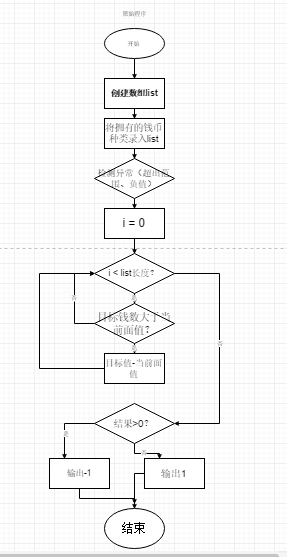
第四步，如果可以实现目标结果，则返回1，否则返回0。

测试程序：

第一步，输入测试用例；

第二步，输入期望输出结果，得到输出。

仓库url：https://github.com/popboykingdjko/softwareTest



1. 详细设计（详细描述具体如何实现，附代码及说明）

详情见代码注释。

源文件代码：

**package** main.java;

/\*\*

\***@author** popboy

\***@createDate** 创建时间：2019年3月19日上午11:03:32

\*\*/

**public** **class** WalletCash {

**int**[] money = {50,20,5,5,1,1,1};//从大到小输入现有的金钱种类

**public** **int** cash(**int** goal) {

**int** remain = 0; //口袋里还剩余的钱

**for**(**int** i = 0;i <money.length;i++) {

remain += money[i];

} //循环计算自己拥有的总金额

**if**(goal > remain) {

**return** -1;

} //输入钱的数量大于自己拥有的，返回错误

**else** **if**(goal < 0){

**return** -1;

} //输入的钱为负数，返回错误

**for**(**int** i = 0;i <money.length;i++) {

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

goal -= money[i];

}

} //贪心法则求解

**if**(goal == 0) {

**return** 1;

}//若能成功给出目标数值，结果为0，返回1

**else** {

**return** -1;

}//若不能成功给出目标数值，结果不为0，返回-1

}

}

测试文件代码：

package test.java;

/\*\*

\*@author popboy

\*@createDate 创建时间：2019年3月19日上午11:05:01

\*\*/

import org.junit.Assert;

import org.junit.Before;

import org.junit.Test;

import main.java.WalletCash;

public class WalletCashTest {

WalletCash c = null;

int[] num = {-20,0,6,9,21,49,52,53,54};//录入测试用例

int[] equal = {-1,1,1,-1,1,-1,1,1,-1};//录入期望结果

@Before

public void testBeforeClass(){

c = new WalletCash();//定义好测试前所需的类

}

@Test

public void testcash(){

for(int i = 0;i < num.length;i++) {

int result = c.cash(num[i]);

Assert.assertEquals(result,equal[i]);

}

//输入测试用例以及期望，进行测试

}

}

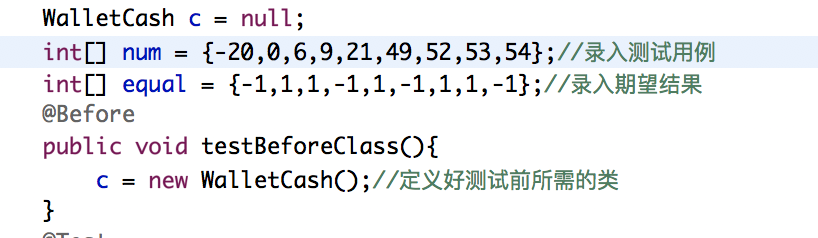
1. 调试分析（在实验过程中遇到的问题以及如何解决）

1在一开始编写测试程序时，我在Test中直接产生目标类，但是实际情况中这种做法会造成测试资源的浪费，所以将其单独列入before

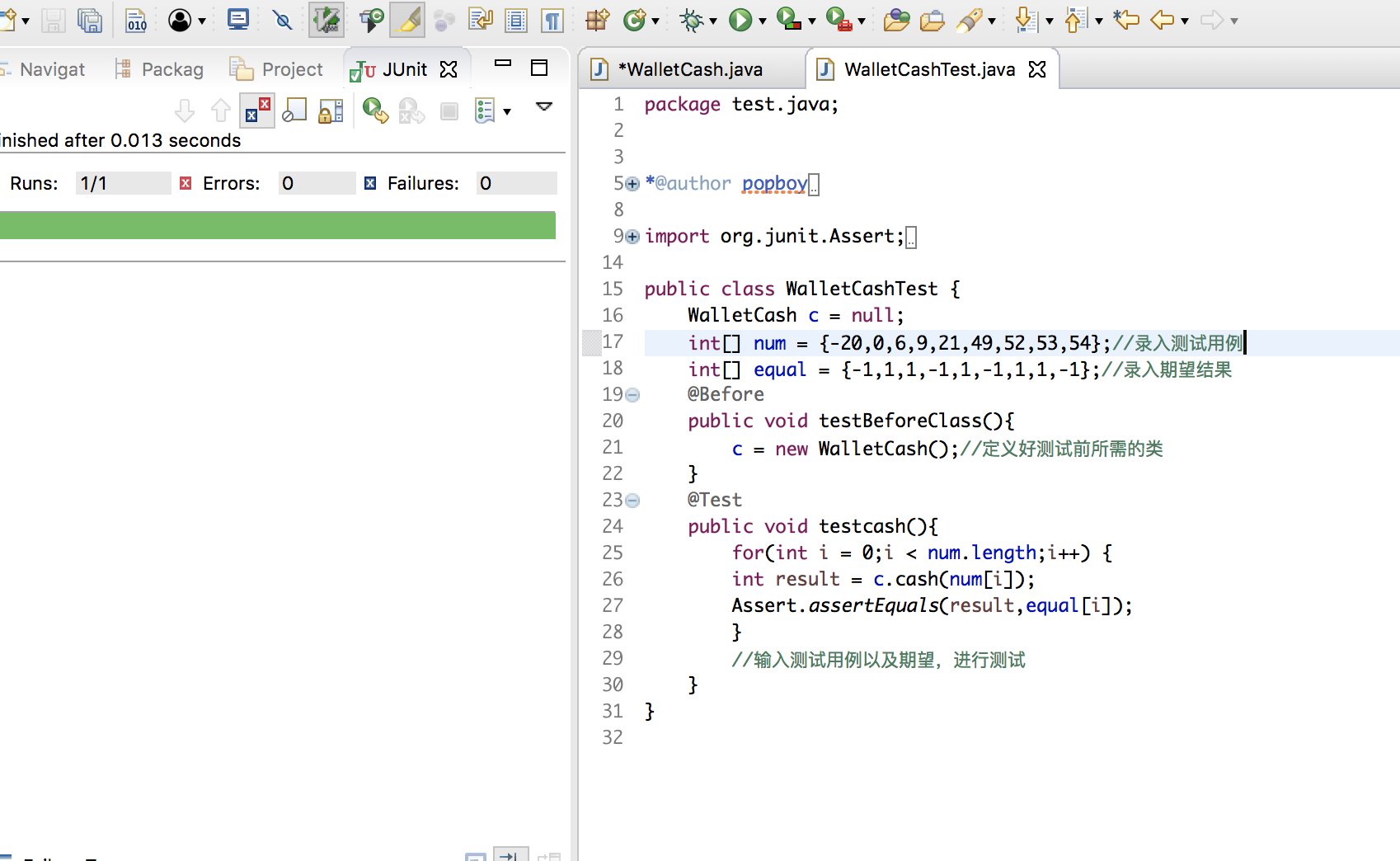
2 开始没有对金钱进行排序，造成了判断复杂度过高，后面通过运用贪心法则，很好的优化了算法

1. 测试结果（描述输入和输出）

测试样例与期望（输入）：



结果：



1. 总结

这次上机实验让我学会了去很好的在单独的文件目录下测试自己的类而不影响代码结构，打开了新的思路，受益匪浅。

仓库url：https://github.com/popboykingdjko/softwareTest