****

**软件测试技术第一次实验报告**



**学 院 软件学院**

**专 业 软件专业**

**年 级 2016级**

**姓 名 王冠**

**2019年 3 月 12 日**

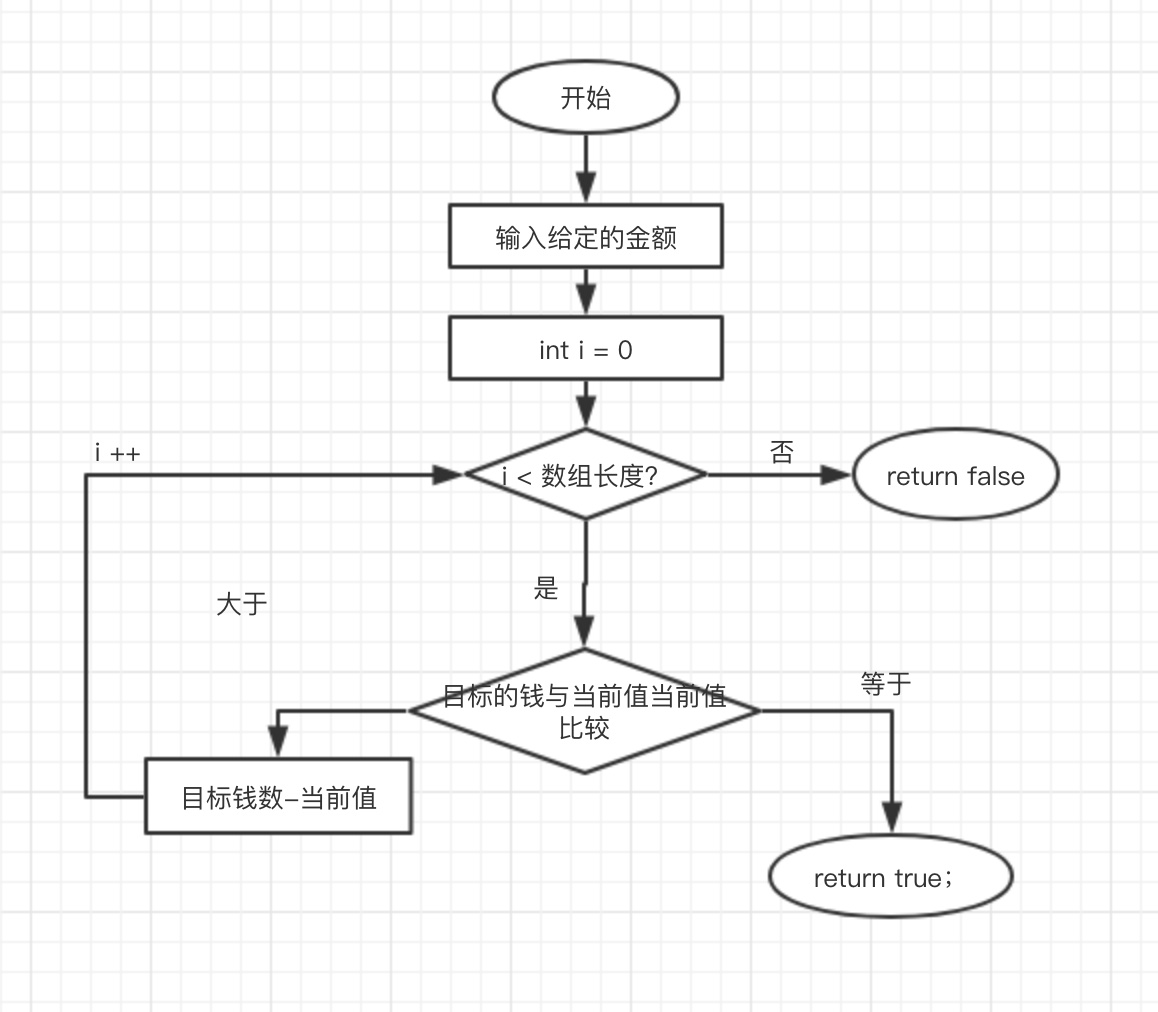
# 软件测试技术第一次实验报告

1. 需求分析（描述具体需求）
2. 在eclipse上安装Junit,Hamcrest。
3. 安装Eclemma。
4. 有一张50元,一张20元,两张5元, 三张一元在钱包里。写一个程序找出是否能给出指定的钱。并且使用Junit进行测试。
5. 概要设计（简单描述设计思路，配合UML图）

步骤一: 按照题目要求初始化一个数组，并按照纸币大小，按照从大到小排列。

步骤二: 写一个方法, 根据用户输入, 给出true,或者false。

步骤三: 新建一个测试用例，输入期望，测试该方法是否正确。



1. 详细设计（详细描述具体如何实现，附代码及说明）
2. 新建java类。

package lab1;

public class Wallet {

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

public boolean canGive(int givenMoney) {

int leftMoney = givenMoney;

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

if(leftMoney > money[i]) {

leftMoney -= money[i];

} else if (leftMoney == money[i]) {

return true;

}

}

return false;

}

}

1. 新建测试类。

package lab1;

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class WalletTest {

@Test

public void testCanGive() {

assertEquals(true,new Wallet().canGive(50));

assertEquals(false,new Wallet().canGive(100));

}

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

实验中先是手动输入测试类，代码写的不规范，不能运行。

解决: 后来发现可以对于指定类的指定方法，自动的生成测试用例，是代码更加的规范。

1. 测试结果（描述输入和输出）

*assertEquals*(**true**,**new** Wallet().canGive(50));

*assertEquals*(**false**,**new** Wallet().canGive(100));

使用此函数进行测试，输出无错误。

1. 总结

测试在写复杂程序时还是很重要的，可以使我们的代码更加正确，减少漏洞。

而使用Junit单元测试框架，是我们的测试更加规范，快速，可以很快发现定位到代码中的问题。