****

**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**

**姓 名 郭皓华**

**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）

1.设计一个程序，给定一个大小为7的整型数组和一个数字，判断用数组中的任意个元素相加能否得到该数字，可以的话返回true，否则返回false。

2.为该程序设计一个测试程序并用Junit测试。

1. 概要设计（简单描述设计思路，配合UML图）

用给定的数字减去数组中的元素，判断能否得到0。

1. 详细设计（详细描述具体如何实现，附代码及说明）

用给定的数字依次减去数组中最大的元素，直到得到3以下的数字，返回true，否则返回false。

package service;

public class Triangle {

public static boolean ifconsist(int x) {

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

if(x <= 83 && x > 0) {

if(x >= 50) {

x = x - 50;

if(x == 0) {

return true;

}

}

if(x >= 20) {

x = x - 20;

if(x==0) {

return true;

}

}

if(x>=5) {

x = x - 5;

if(x == 0) {

return true;

}

}

if(x >= 5) {

x = x - 5;

if(x == 0) {

return true;

}

}

if(x <= 3) {

return true;

}

}

return false;

}

设计测试程序以及测试用例：

package service;

import static org.junit.Assert.assertEquals;

import java.util.Arrays;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.junit.runners.Parameterized;

import org.junit.runners.Parameterized.Parameters;

import service.Triangle;

@RunWith(value = Parameterized.class)

public class TriangleTest {

private boolean result;

private int testNumber;

public TriangleTest(boolean result, int testNumber) {

this.result = result;

this.testNumber = testNumber;

}

@Parameters(name="{index}: ifconsist({1}) = {0}")

public static Iterable<Object[]> data1(){

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

{false,-1},

{true,2},

{false,9},

{true,11},

{false,14},

{false,18},

{true,28},

{true,58},

{true,83}

});

}

@Test

public void test() {

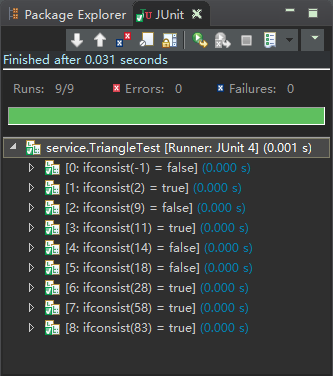
assertEquals(result,Triangle.ifconsist(testNumber));

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

设计的测试用例在运行时与预期的结果不符，输入-1得到的结果应该是false，结果得到的是true，原因是triangle程序的判断前提并没有要求输入大于0，于是在最开始的判断中加了大于0的前提判断。

1. 测试结果（描述输入和输出）



1. 总结

学会了使用Junit来测试程序，Junit确实是很方便的测试工具。在设计测试用例时一定要尽可能考虑各种情况，否则测试就没有意义。