**《软件测试》**

**实验报告一 ——JUnit测试方法**

**姓 名： 汪成肸 学 号： 2020105002**   **院 系： 计算机与信息学院 专 业：计算机科学与技术**

**实 验 室： 实验日期： 2023/03/08**

**总评成绩： 审阅教师：**

### 一、实验目的：

1. 练习基本的编程能力；
2. 学习JUnit自动测试框架，掌握非参数化和参数化测试方法；
3. 理解以下三个题目，这是后面讲解黑盒测试方法和白盒测试方法的基础案例；

### 二、实验环境：

IntelliJ IDEA 2022.2.2

### 三、实验要求：

1. 使用Java语言实现以下三个程序；
2. 使用测试用例验证程序的正确性（JUnit）；

### 四、实验步骤与内容

1. **题目：三角形题目**

**输入**3个整数a、b和c分别作为三角形的三条边，要求a、b和c必须满足以下条件：

1、整数

2、3个数

3、边长大于等于1，小于等于100

4、任意两边之和大于第三边

**输出**为5种情况之一：

如果不满足条件1、2、3，则程序输出为“输入错误”。

如果不满足条件4，则程序输出为“非三角形”。

如果三条边相等，则程序输出为“等边三角形”。

如果恰好有两条边相等，则程序输出为“等腰三角形”。

如果三条边都不相等，则程序输出为“一般三角形”。

1. **测试用例：**

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试用例** | **预测结果** | **实际结果** |
| **1** | -1, -1, -1 | 输入错误 | 输入错误 |
| **2** | 3,3,3 | 等边三角形 | 等边三角形 |
| **3** | 3,3,6 | 非三角形 | 非三角形 |
| **4** | 3,3,5 | 等腰三角形 | 等腰三角形 |
| **5** | 3,4,5 | 不等边三角形 | 不等边三角形 |

1. **代码实现：**

|  |
| --- |
| **实现代码：**  package org.wcx.code;   public class Triangle {  public String classify(int a, int b, int c) {  if (a <1 || a>100 || b<1 || b>100 || c<1 || c> 100) {  return "输入错误";  }   if (!((a + b > c) && (a + c > b) && (b + c > a))) {  return "非三角形";  } else if (a == b && a == c && b == c) {  return "等边三角形";  } else if (a != b && a != c && b != c) {  return "不等边三角形";  } else {  return "等腰三角形";  }  } }  测试代码：  package org.wcx.code;  import org.junit.Assert; import org.junit.Test; import org.junit.jupiter.api.DisplayName;   public class TriangleTest {   private final Triangle triangle = new Triangle();   @Test  @DisplayName("输入错误")  public void errorParam() {  Assert.*assertEquals*("输入错误", triangle.classify(-1, -1, -1));  }   @Test  @DisplayName("非三角形")  public void errorTriangle() {  Assert.*assertEquals*("非三角形", triangle.classify(3, 3, 6));  }   @Test  @DisplayName("等边三角形")  public void equilateralTriangle() {  Assert.*assertEquals*("等边三角形", triangle.classify(3,3,3));  }   @Test  @DisplayName("不等边三角形")  public void scaleneTriangle() {  Assert.*assertEquals*("不等边三角形", triangle.classify(3, 4, 5));  }   @Test  @DisplayName("等边三角形")  public void isoscelesTriangle() {  Assert.*assertEquals*("等腰三角形", triangle.classify(3, 3, 5));  }  }  测试结果： |

### 五、结论分析与体会

通过这次实验，我明白了Junit的优点，可以直观的判断数据的正确与否，避免了后面的数据出错，从而方便了程序员进行修改的麻烦性，而我在这次实验中也学到了此插件的方便性，练习了基本的编程能力，学习JUnit自动测试框架，掌握参数化测试方法，在以后的实验 中我会考虑多种测试用例，做到多方面考虑。

### 六、仓库地址

https://github.com/shuaishuaixi03/software-testing/tree/main/project-0308/code