**《软件测试》**

**实验报告一 ——JUnit测试方法**

**姓 名： 徐静 学 号： 2020112825**

**院 系： 计算机与信息学院 专 业：**  **数字媒体技术**

**实 验 室： 实验日期： 2021/3/19**

**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：

Eclipse2020,JUnit

### 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）；
3. 测试用例数据可以参考naiveX.xls，并运行naiveX.xls;(p24页第8题）；

### **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** | 0,4,5 | 输入错误 | 输入错误 |
| **2** | 3,4,5 | 不等边三角形 | 不等边三角形 |
| **3** | 4,4,4 | 等边三角形 | 等边三角形 |
| **4** | 4,4,6 | 等腰三角形 | 等腰三角形 |
| **5** | 3,4,7 | 非三角形 | 非三角形 |

1. **代码实现：**

|  |
| --- |
| **实现代码：**  package inittaskdemo3.app;  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 inittaskdemo3.app;  import static org.junit.jupiter.api.Assertions.assertEquals;  import org.junit.jupiter.api.DisplayName;  import org.junit.jupiter.api.Test;  class TriangleTest {  @Test  @DisplayName("输入错误")  void parameters\_error\_test() {  Triangle triangle = new Triangle();  String type = triangle.classify(0, 4, 5);  assertEquals("输入错误", type);  }  @Test  @DisplayName("不等边三角形")  void scalene\_test() {  Triangle triangle = new Triangle();  String type = triangle.classify(3, 4, 6);  assertEquals("不等边三角形", type);  }  @Test  @DisplayName("等边三角形")  void dengbian\_test() {  Triangle triangle = new Triangle();  String type = triangle.classify(4, 4, 4);  assertEquals("等边三角形", type);  }  @Test  @DisplayName("等腰三角形")  void dengyao\_test() {  Triangle triangle = new Triangle();  String type = triangle.classify(4, 4, 6);  assertEquals("等腰三角形", type);  }  @Test  @DisplayName("非三角形")  void notATri\_test() {  Triangle triangle = new Triangle();  String type = triangle.classify(3, 4, 7);  assertEquals("非三角形", type);  }  }  测试结果：    覆盖率： |

### **五、结论分析与体会**

这次实验室是我第一次接触和使用Junit5和Gradle，从前也只是停留在听说过Maven创建程序，所以起步还是比较困难的。其中，我觉得这次实验的重难点其实并不是代码的理解和编写，而是配置两种工具的使用环境和深刻理解两种工具的实际作用。从最开始的gradle安装，到后来的包报错，完善代码的编写，我查阅了很多很多资料，包括两个工具的各个方面，最后终于通过自己的努力完成了这次实验。所以，困难不应该成为我们学习的止步石，它是阻碍，但并不是放弃的理由。

### **六、仓库地址**

https://github.com/cdwvcf/rjce1.git