**《软件测试》**

**实验报告二 ——边界值分析方法**

**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. 巩固 git 协作式管理工具的使用；
2. 巩固 gradle 项目构建 Java 应用的基本操作
3. 掌握边界值分析方法设计测试用例的方法
4. 掌握 Junit5参数化测试方法；

### 二、实验环境：

Eclipse2020,JUnit

### 三、实验要求：

1. 学习Junit5 参数化测试方法，参考[Junit5 五分钟教程](https://www.jianshu.com/p/3183fb69fd74" \o "Junit5 五分钟教程)
2. 针对实验一实现判断三角形类型程序Triangle，采用边界值分析方法设计测试用例，并采用Junit5 编写完成三角形类型程序测试用例，并提交到代码仓库
3. 按照[实验报告模板](https://star.jmhui.com.cn/u/cms/www/202203/06160105hi3m.docx) 编写实验报告，以“学号-姓名-软件测试实验二”命名，提交到雨课堂“软件测试实验二”

### **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”。

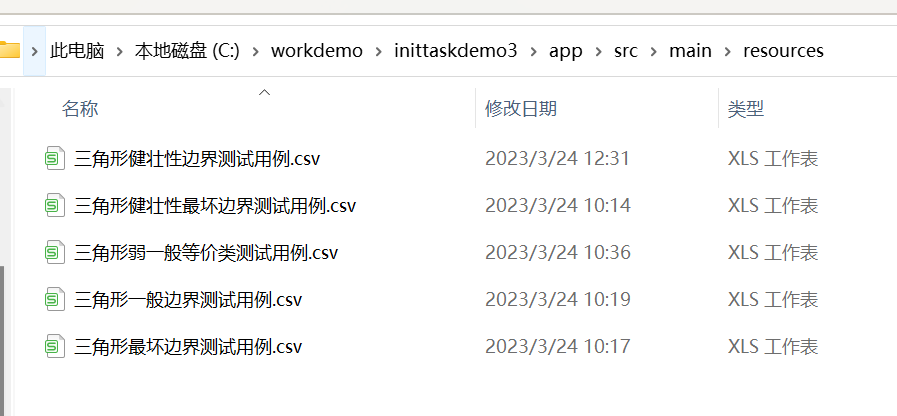
如果三条边相等，则程序输出为“等边三角形”。

如果恰好有两条边相等，则程序输出为“等腰三角形”。

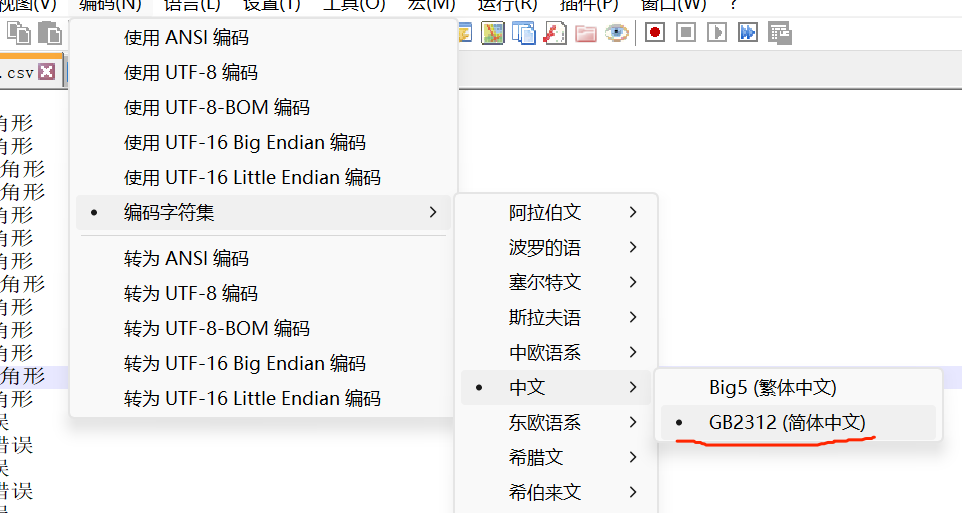
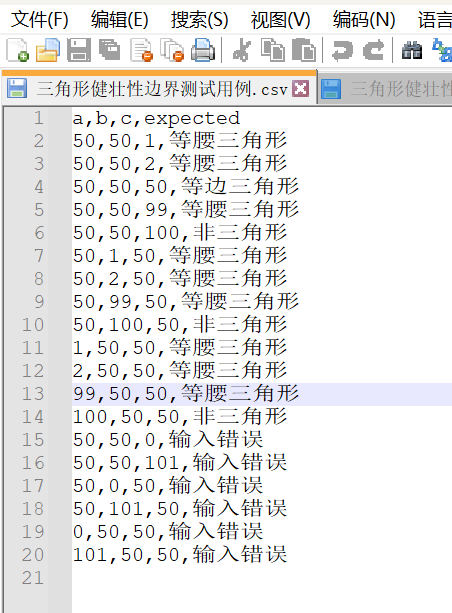
如果三条边都不相等，则程序输出为“一般三角形”。

1. **测试用例：**

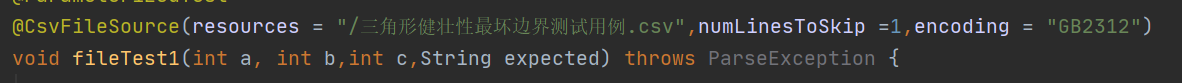
这里采用的是csv文件导入生成的边界测试用例，在resources文件夹中一共包括五个文件：



这里需要说明一下的是，csv文件是以逗号分割的文件：



对应的是Idea中的代码，注意代码中的encoding代表的是csv文件的编码格式，默认为GB2312，另，文件中每一行有三个整形数据和一个字符串类型的数据，所以代码中函数接收参数为：



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 org.junit.jupiter.api.DisplayName;  import org.junit.jupiter.api.Test;  import org.junit.jupiter.params.ParameterizedTest;  import org.junit.jupiter.params.provider.CsvFileSource;  import org.junit.jupiter.params.provider.CsvSource;  import java.text.ParseException;  import static org.junit.jupiter.api.Assertions.assertEquals;  class TriangleTest2 {  @Test  @DisplayName(value="不等边三角形")  void testTriangle() {  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 equallaterialTriangle() {  Triangle triangle = new Triangle();  String type = triangle.classify(3, 3, 3);  assertEquals("等边三角形", type);  }  @Test  @DisplayName("非三角形")  void notTriangle() {  Triangle triangle = new Triangle();  String type = triangle.classify(3, 3, 6);  assertEquals("非三角形", type);  }  @ParameterizedTest  @CsvSource({  "3,4,6,不等边三角形",  "3,3,3,等边三角形",  "3,3,6,非三角形"  })  void paramTriangle(int a, int b,int c,String expected) {  Triangle triangle = new Triangle();  String type = triangle.classify(a, b, c);  assertEquals(expected, type);  }  @DisplayName(value="三角形一般边界测试用例")  @ParameterizedTest  @CsvFileSource(resources = "/三角形一般边界测试用例.csv",numLinesToSkip =1,encoding = "GB2312")  void fileTest(int a, int b,int c,String expected) throws ParseException {  Triangle triangle=new Triangle();  String type = triangle.classify(a, b, c);  assertEquals(expected, type);  }  @DisplayName(value="三角形健壮性最坏边界测试用例")  @ParameterizedTest  @CsvFileSource(resources = "/三角形健壮性最坏边界测试用例.csv",numLinesToSkip =1,encoding = "GB2312")  void fileTest1(int a, int b,int c,String expected) throws ParseException {  Triangle triangle=new Triangle();  String type = triangle.classify(a, b, c);  assertEquals(expected, type);  }  @DisplayName(value="三角形健壮性边界测试用例")  @ParameterizedTest  @CsvFileSource(resources = "/三角形健壮性边界测试用例.csv",numLinesToSkip =1,encoding = "GB2312")  void fileTest2(int a, int b,int c,String expected) throws ParseException {  Triangle triangle=new Triangle();  String type = triangle.classify(a, b, c);  assertEquals(expected, type);  }  @DisplayName(value="三角形弱一般等价类测试用例")  @ParameterizedTest  @CsvFileSource(resources = "/三角形弱一般等价类测试用例.csv",numLinesToSkip =1,encoding = "GB2312")  void fileTest3(int a, int b,int c,String expected) throws ParseException {  Triangle triangle=new Triangle();  String type = triangle.classify(a, b, c);  assertEquals(expected, type);  }  @DisplayName(value="三角形最坏边界测试用例")  @ParameterizedTest  @CsvFileSource(resources = "/三角形最坏边界测试用例.csv",numLinesToSkip =1,encoding = "GB2312")  void fileTest4(int a, int b,int c,String expected) throws ParseException {  Triangle triangle=new Triangle();  String type = triangle.classify(a, b, c);  assertEquals(expected, type);  }  }  测试结果：    覆盖率： |

### **结论分析与体会**

这次实验相较于上次比较简单，主要是又了解和接触了新的知识：边界测试案例编写和csv文件，由于是引入文件数据进行边界测试，所以编码的问题就显得尤为重要，我也是在这里研究了很久，最后才有所了解：Idea中所有项目的代码默认为UTF-8编码，而csv文件默认为GB2312编码，所以我们在编写代码是需要特定指出引入文件格式。

### **六、仓库地址**

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