**《软件测试》**

**实验报告二 ——边界值分析方法**

**姓 名： 曹倜源 学 号： 2020112805**

**院 系： 计算机与信息学院 专 业：计算机科学与技术**

**实 验 室： J2301 实验日期： 2023/3/15**

**总评成绩： 审阅教师： 杨青**

### 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：

IntelliJ IDEA2020,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. 使用gradle创建 Java 工程，学习Junit5 参数化测试技术

参考 gradle 入门教程，构建基于 Java 的 Application 工程

为加快依赖包的下载速度，可以在build.gradle 中定义下载包的仓库地址，如

repositories {

maven { url 'https://maven.aliyun.com/repository/public/' }

}

常用gradle命令

gradle init

gradle build

gradle clean

gradle run

gradle test

gradle classes

主要学习ParameterizedTest，CSVSource、CSVFileSource，案例如下

@ParameterizedTest

@CsvSource({

"1,2,3,非三角形",

"-1,2,3,输入错误"

})

void testWithCsvSource(int a,int b,int c,String expected) {

assertEquals(expected, triangle.classify(a,b,c));

}

它也支持从文件导入，文件要放在类路径下，比如resource目录下，且以“/”开头定义相对路径，例如 @CsvFileSource(resources = “/two-column.csv”, numLinesToSkip = 1),如

@ParameterizedTest

@CsvFileSource(resources = "/triangle.csv")

void test\_csv\_file\_source(int a, int b, int c, String expected) {

assertEquals(expected, triangle.classify(a,b,c));

}

2. 编写代码实现

使用自己熟悉的 IDE 开发环境如 Idea 或 Eclipse 等，导入生成的工程，参考以下示例代码，实现三角形类型判断功能

/\*\*

\* @Title: Triangle.java

\* @Package cn.edu.ctgu.junitTest

\* @Description: 测试三角形是否为等边、等腰或者是不等边三角形，或不是三角形

\* @author tiger

\* @version V1.0

\*/

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";

}

}

}

3. 分别采用以下边界值分析方法判断三角形类型设计测试用例

3.1 一般边界值方法

3.2 健壮性边界值方法

3.3 最坏情况一般边界值分析方法

3.4 最坏情况健壮性边界值分析方法

4. 分别针对上述四种边界值分析方法，使用 Junit5 编写自动化单元测试代码。

5. 提交项目到 git 代码仓库

1. **代码实现：**

|  |
| --- |
| **实现代码：**  package cn.edu.ctgu;  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 cn.edu.ctgu;  import org.junit.jupiter.api.Assertions;  import org.junit.jupiter.params.ParameterizedTest;  import org.junit.jupiter.params.provider.CsvSource;  class TriangleByParameterizeTest {  @ParameterizedTest(name = "{index} => a={0}, b={1}, c={2}, expected={3}")  @CsvSource({  "1, 2, 3, 非三角形",  "2, 2, 3, 等腰三角形",  "2, 2, 2, 等边三角形",  "3, 4, 5, 不等边三角形",  "50, 50, 50, 等边三角形",  "100, 100, 100, 等边三角形",  "1, 1, 100, 非三角形",  "10, 20, 100, 非三角形"  })  public void testTriangleClassify(int a, int b, int c, String expected) {  Triangle triangle = new Triangle();  Assertions.assertEquals(expected, triangle.classify(a, b, c));  }  }  测试结果：    覆盖率： |

### 五、结论分析与体会

在本次实验中，我们使用边界值分析的方法设计了测试用例，然后使用Junit5框架编写了三角形类型程序的测试用例，并对程序进行了测试。根据测试结果，我们可以得出以下结论和体会：

参数化测试可以大大提高测试用例的覆盖率和效率，避免手动编写大量的测试用例，也能减少出错的机会。

边界值分析是一种常用的测试设计方法，可以针对不同的输入值设计出各种情况的测试用例，提高测试的准确性和可靠性。

在本次实验中，我们使用了Junit5框架进行测试用例编写，该框架相比较于Junit4有更多的新特性和改进，例如支持参数化测试、动态测试、条件测试等，具有更好的灵活性和扩展性。

在编写测试用例的过程中，需要充分考虑程序的各种可能输入情况，包括正常情况和异常情况，并对程序进行全面的测试，以保证程序的正确性和稳定性。

实验过程中需要充分理解和掌握相关知识和技术，包括边界值分析、Junit5框架的使用、代码仓库的操作等，这有助于提高实验的效果和学习的质量。

综上所述，本次实验让我们更好地了解了参数化测试方法和边界值分析的应用，掌握了Junit5框架的使用方法，同时也体会到了测试的重要性和编写测试用例的技巧。

### 六、仓库地址

https://github.com/Ctyring/soft-test.git