**深 圳 大 学 实 验 报 告**

**课程名称：Java程序设计**

**实验项目名称： 作业6 JUnit应用**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师：卢亚辉**

**报告人： 林宪亮 学号：2022150130 班级：国际班**

**实验时间：2023年12月21日（周四）~2022年12月30日（周六）**

**实验报告提交时间： 2023年12月24日**

**教务部制**

|  |
| --- |
| **实验目的与要求：**  **实验目的：** 掌握JUnit应用。  **实验要求：**   * 学习测试用例的编写 * 使用JUnit进行代码测试  1. 参考https://www.liaoxuefeng.com/wiki/1252599548343744/1304048154181666，编写Factorial和FactorialTest，给出代码运行结果截图； （20分） 2. 参考https://www.liaoxuefeng.com/wiki/1252599548343744/1304049490067490，编写Calculator和CalculatorTest，给出代码运行结果截图；（20分） 3. 参考https://www.liaoxuefeng.com/wiki/1252599548343744/1304064312737826,编写异常测试代码，给出代码运行结果截图；（20分） 4. 参考https://www.liaoxuefeng.com/wiki/1252599548343744/1304073489874978，编写条件测试代码，给出代码运行结果截图；（10分） 5. 参考https://www.liaoxuefeng.com/wiki/1252599548343744/1304065789132833，编写参数化测试代码，给出代码运行结果截图；（10分）   报告写作。要求：主要思路有明确的说明，重点代码有详细的注释，行文逻辑清晰可读性强，报告整体写作较为专业。（20分）  **说明：**  （1）本次实验课作业满分为100分，占总成绩的比例（待定）。  （2）本次实验课作业截至时间2023年12月30日（周六）21:59。  （3）报告正文：请在指定位置填写，本次实验**需要单独提交源程序文件**（源程序单独打包在Blackboard中上传，不要包含外部导入的包）。  （4）个人信息：WORD文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。  （5）提交方式：截至时间前，请在Blackboard平台中提交。  （6）发现抄袭（包括复制&粘贴整句话、整张图），**抄袭者和被抄袭者的成绩记零分（含抄袭往届同学的作业）。**  （7）延迟提交，不得分；如有特殊情况，请于截至日期之后的48小时内发邮件到luyahui@szu.edu.cn，并在邮件中注明课程名称、作业名称、姓名、学号等信息，以及特殊情况的说明，我收到后会及时回复。  （8）期末考试阶段补交无效。 |

|  |
| --- |
| （1）  程序截图：  方法：对着网页的代码编写即可，实现斐波那契数列的计算。  测试程序：有一个测试方法（@Test标识），里面有五个测试点，测试fact方法。  运行结果：  这是测试成功的代码，不会输出信息。  如果程序运行成功则会输出具体的报错信息。  上图的意思是期望值是0，但是函数输出值是1，产生了错误。  （2）  程序截图：  方法：  对着网页上代码copy即可。就是实现加法和减法而已。  测试方法：  @BeforeEach：表示在每个测试方法执行之前执行的方法。setUp() 方法在每个测试方法执行前都会被调用。  @AfterEach：表示在每个测试方法执行之后执行的方法。tearDown() 方法在每个测试方法执行后都会被调用。  @Test：表示下面的方法是一个测试方法。  void testAdd() { ... }：这是一个测试方法，用于测试 Calculator 类的 add 方法。  assertEquals(expected, actual)：这是 JUnit 提供的断言方法之一，用于断言预期值与实际值是否相等。如果相等，则测试通过；否则，测试失败。  this.calculator.add(100)：调用 Calculator 类的 add 方法，并传入参数 100，然后与预期的结果 100 进行比较。  void testSub() { ... }：这是另一个测试方法，用于测试 Calculator 类的 sub 方法。  运行结果：  测试通过  待测试方法：  当输出的参数大于20时，丢出异常，其它部分和（1）相同。  测试方法：  新增一个测试方法testNegative，JUnit提供assertThrows()来期望捕获一个指定的异常。第二个参数封装了我们要执行的会产生异常的代码。当我们执行Factorial.fact(21)时，必定抛出ArithmeticException。assertThrows()在捕获到指定异常时表示通过测试，未捕获到异常，或者捕获到的异常类型不对，均表示测试失败。  运行结果：  如果测试成功：  如果测试失败：  会提示期望得到一个ArithmeticException，但什么异常也没有得到，或者可能得到别的异常。  （4）  代码截图：  如果是在windows系统，则会返回“C:\\”+filename。  如果是在mac或者linux系统，则会返回“/usr/local/”+filename。  测试程序：  @EnabledOnOs(OS.WINDOWS)：在满足条件（操作系统为 Windows）时才会执行该测试方法。  EnabledOnOs({ OS.LINUX, OS.MAC })：这个注解指定了在 Linux 或者 macOS 系统上才会执行这个测试方法。  @DisabledOnOs(OS.WINDOWS)：这个注解指定了在 Windows 系统上禁用这个测试方法。  @DisabledOnJre(JRE.JAVA\_8)：这个注解指定了在 Java 8 及以下版本禁用这个测试方法。  @EnabledIfSystemProperty(named = "os.arch", matches = ".\*64.\*")：这个注解指定了当系统属性 os.arch 包含 64 位信息时才会运行这个测试方法。  运行结果：  testWindows()，testOnJava9OrAbove() ，testOnlyOn64bitSystem()三个测试方法被执行，其他的测试方法都没有执行，不被执行的原因也输出了出来。  （5）  程序截图：  检查输入字符串 s 是否为空。如果为空（长度为0），则返回空字符串。  如果输入字符串不为空，使用 Character.toUpperCase(s.charAt(0)) 将第一个字符大写。  接着，使用 s.substring(1).toLowerCase() 将字符串的其余部分转换为小写。  最后，将大写的第一个字符与其余部分转换为小写的字符串连接起来，并返回结果字符串。  测试方法：  使用了三种参数化测试方法：  testCapitalize1 方法使用了 @MethodSource 注解，其目的是提供参数化测试所需的参数。它验证了 StringUtils.capitalize 方法在不同输入下的行为，并使用 assertEquals 来比较预期结果和实际结果是否一致。  testCapitalize2 方法使用了 @CsvSource 注解，以逗号分隔的字符串提供了测试参数。它也测试了 StringUtils.capitalize 方法的不同输入，并进行了预期结果与实际结果的比较。  testCapitalizeUsingCsvFile 方法使用了 @CsvFileSource 注解，从一个 CSV 文件中读取测试参数。该方法也测试了 StringUtils.capitalize 方法的行为，并进行了预期结果与实际结果的比较。  运行结果：  测试全部通过。  ++++++++++++++++++++++++++++++++++++++++++++++++++++++  **其他（例如感想、建议等等）。**  通过本次实验，我学会了使用JUnit对java的方法进行测试。本次实验相比与其它的实验还是比较简单的，只需要对着网页上的知识点和代码进行编写即可，所以也几乎没有遇到什么问题。唯一遇到的问题就是我的IDEA不兼容.csv文件的问题，需要在系统的文件夹里新建一个csv文件，然后输出测试数据。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  2021年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。