**《软件测试》**

**实验报告一 ——构建软件测试实验环境**

**姓 名： 贾坤键 学 号： 202110120407**

**院 系： 计算机与信息学院 专 业：大数据**

**实 验 室： j312 实验日期： 2024/2/27**

**总评成绩： 审阅教师：**

### 一、实验目的：

1. 掌握 git 协作式管理工具；
2. 掌握 gradle 项目构建基本操作
3. 锻炼基本代码编程能力

### 二、实验环境：

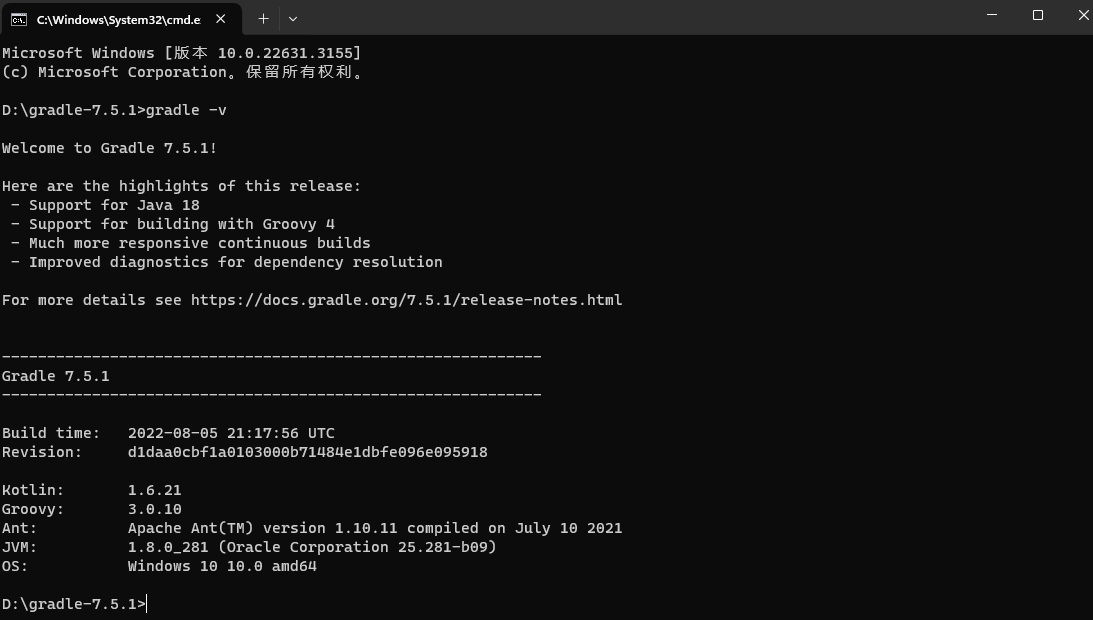
Eclipse2020,JUnit

### 三、实验要求：

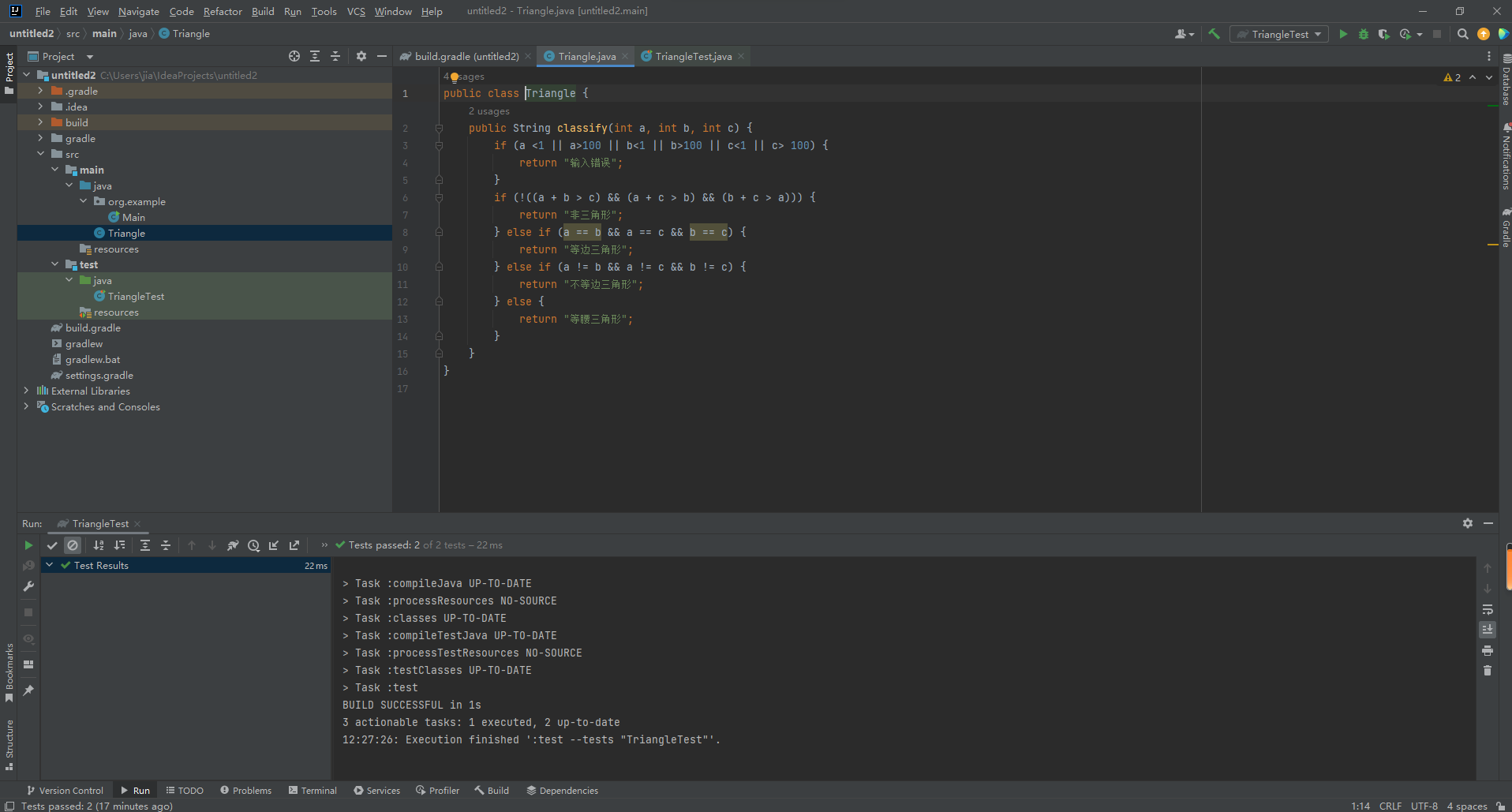
1. 学习使用 JUnit5 编写测试用例，参考[Junit5 五分钟教程](https://www.jianshu.com/p/3183fb69fd74" \o "Junit5 五分钟教程)或 使用 Pytest 编写测试用例，参考[PyTest学习指南](https://blog.csdn.net/qq_37806690/article/details/127218624?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-1-127218624-blog-130952327.235^v43^pc_blog_bottom_relevance_base9&spm=1001.2101.3001.4242.1&utm_relevant_index=4" \o "PyTest学习指南)
2. 使用 git 管理代码（gitee或GitHub均可）
3. 使用 gradle 项目构建工具，参考 [gradle 入门教程](https://www.cnblogs.com/hellxz/p/helloworld-gradle.html" \o "gradle 入门教程);或者使用 Maven 构建工具创建项目；或者使用其他 IDE 环境；
4. 按照以下实验要求，使用 Java 语言实现判断三角形类型程序Triangle，并采用Junit5 或PyTest 编写完成三角形类型程序测试用例，并提交到代码仓库
5. 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-软件测试实验一”命名，提交到雨课堂“软件测试实验一”

### **四、实验步骤与内容**

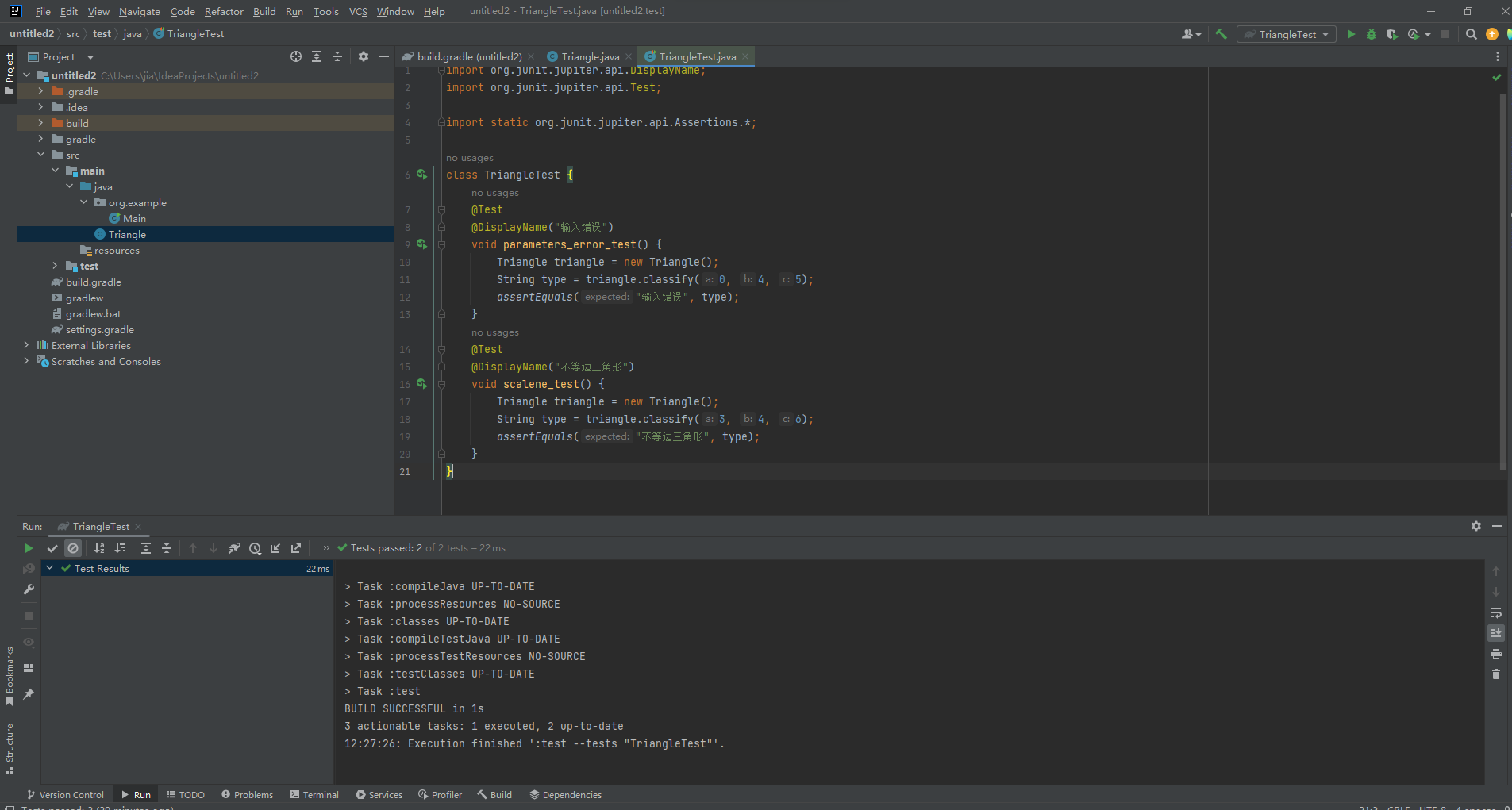
下载安装和配置 gradle 7.3.3



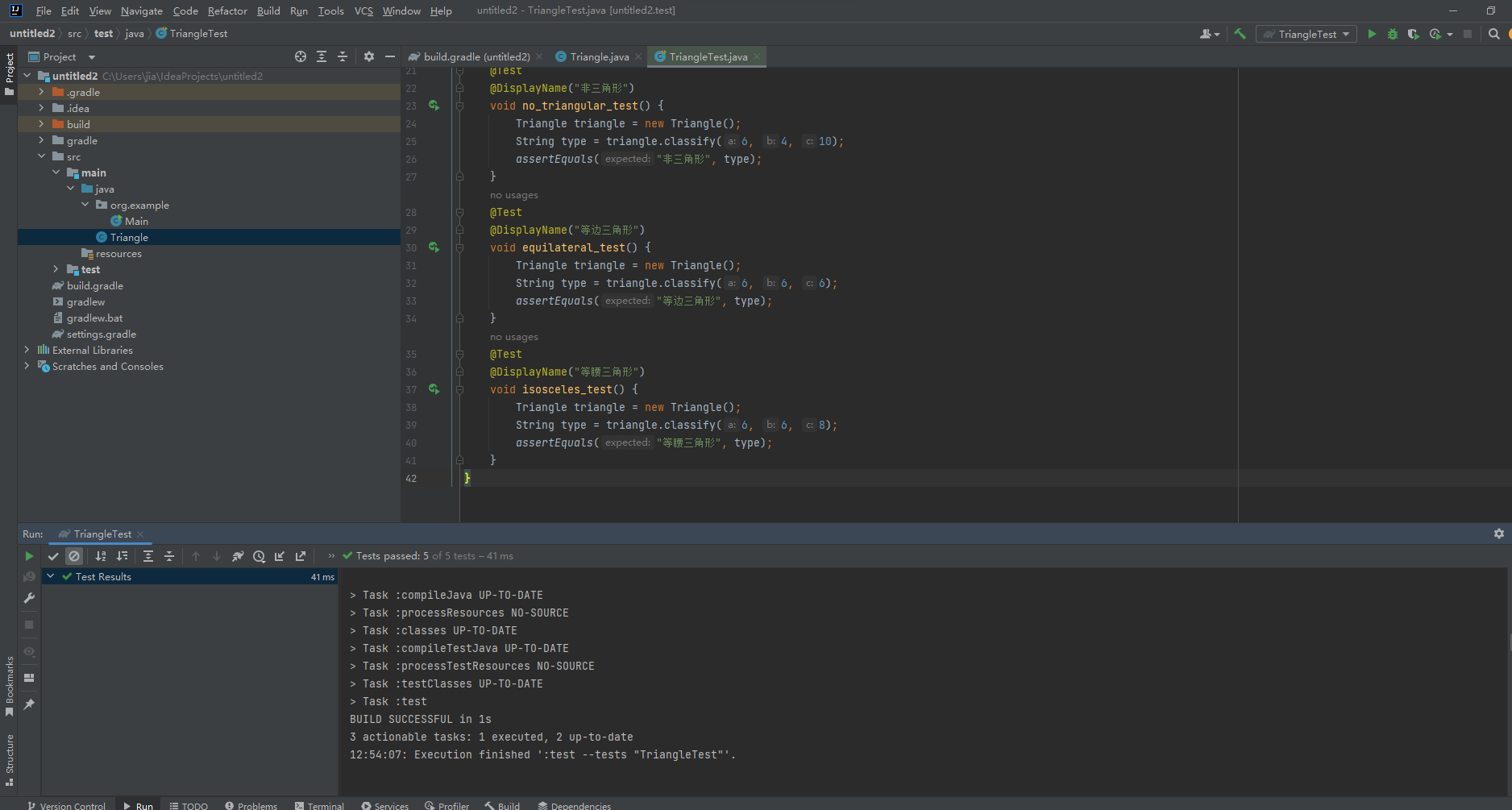
在结构目录src-main-java下创建Triangle类，并编写代码



右键-go to-test，idea在Test创建了TriangleTest类，右键运行



再添加3个实例，分别为非三角形，等边三角形，等腰三角形



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** | 3,4,3=6 | 不等边三角形 | 不等边三角形 |
| **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 |

1. **代码实现：**

|  |
| --- |
| **实现代码：**  package ***cn.edu.ctgu.junitTest***; /\*\*  \* @Title: Triangle.java  \* @Package cn.edu.ctgu.junitTest  \* @Description: 测试三角形是否为等边、等腰或者是不等边的  \* @author tiger  \* @version V1.0  \*/ public class ***Triangle* {** public ***String classify*(**int *a*, int *b*, int *c***) {** 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.junitTest***;  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;  /\*\*\*  \*   \* @author tiger  \* @date 2021年3月4日-下午8:43:13  \* @description 三角形测试用例  \*/ class ***TriangleTest* {** @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  void ***equallaterialTriangle*() {  *Triangle* triangle** = new ***Triangle*()**;    ***String* type** = **triangle**.***classify*(**3, 3, 3**)**;  assertEquals**(**"等边三角形", **type)**;  **}** @Test  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 = "UTF-8"**)** void ***fileTest*(**int *m*,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="三角形健壮性最坏情况测试用例"**)** @ParameterizedTest  @CsvFileSource**(**resources = "/三角形健壮性最坏情况测试用例.csv",numLinesToSkip =1,encoding = "UTF-8"**)** void ***fileTest1*(**int *m*,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="三角形健壮测试用例"**)** @ParameterizedTest  @CsvFileSource**(**resources = "/三角形健壮测试用例.csv",numLinesToSkip =1,encoding = "UTF-8"**)** void ***fileTest2*(**int *m*,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 = "UTF-8"**)** void ***fileTest3*(**int *m*,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="三角形最坏情况测试用例"**)** @ParameterizedTest  @CsvFileSource**(**resources = "/三角形最坏情况测试用例.csv",numLinesToSkip =1,encoding = "UTF-8"**)** void ***fileTest4*(**int *m*,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： |

### **五、结论分析与体会**

通过这次实验，我明白了如何创建gradle，并如何编写代码进行测试

### **六、仓库地址**

https://github.com/Jiakunjian4/school-homeworks/tree/main/Software%20testing/experment-one