|  |  |  |  |
| --- | --- | --- | --- |
| 学号： 221801337 | 姓名： 林浩然 | 专业：软件工程 | 班级： 3班 |

**《软件质量与测试》实验二：白盒测试（4学时）**

**实验目的：**掌握基于逻辑驱动覆盖测试的白盒测试用例的设计方法并使用Junit完成单元测试。

**实验环境：** Idea open

jdk 15.0.2 2021-01-19

Linux manjaro-PC 5.10.23-1-MANJARO #1 SMP PREEMPT Thu Mar 11 18:47:18 UTC 2021 x  
86\_64 GNU/Linux

**实验内容：**

1、白盒测试用例的设计

（1）新建java语言项目（工具为IDEA 或者 eclipse/myeclipse），在src下新建Shape.java文件,在该类中增加方法 String judgeTriangle(int a,int b,int c)实现以下程序要求：

**输入条件：** 要求输入三个整数a、b、c，必须满足以下条件：

条件1 1≤a≤100 条件4 a< b+c

条件2 1≤b≤100 条件5 b< a+c

条件3 1≤c≤100 条件6 c< a+b

**输出结果：**

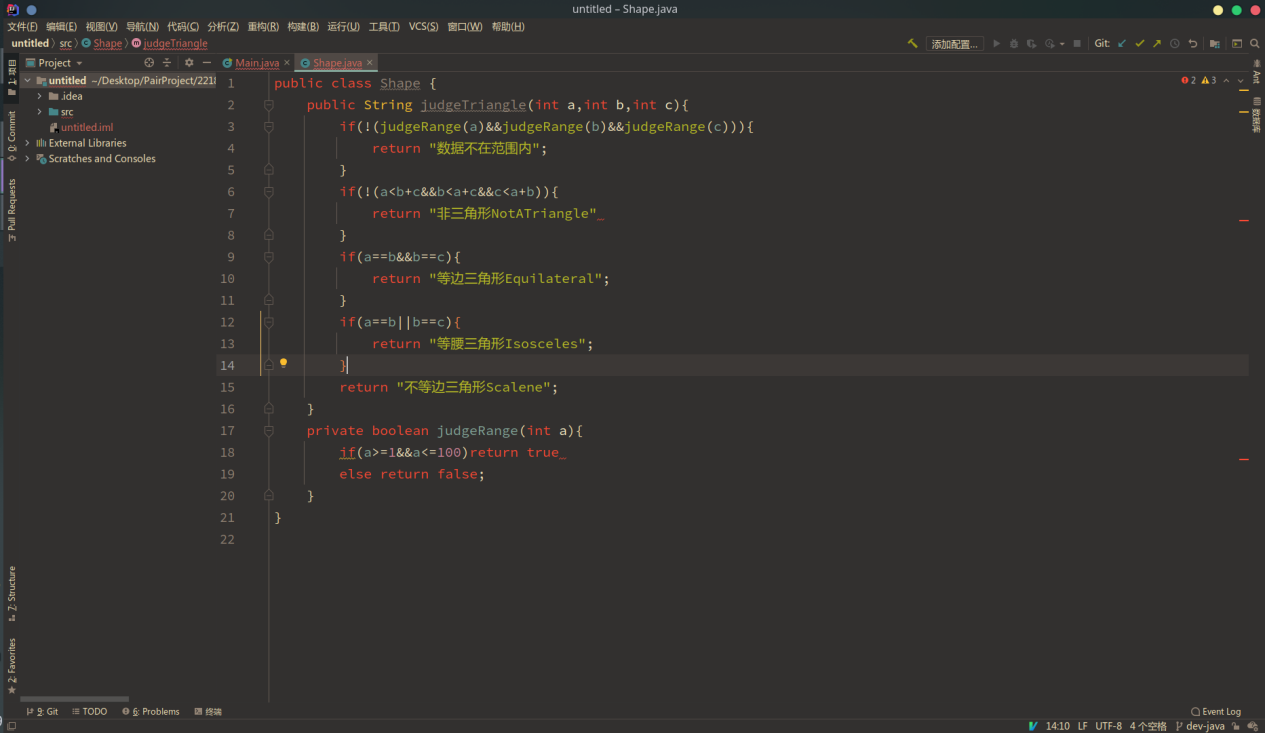
（a）如果不满足条件4、条件5和条件6中的一个，则程序输出为“非三角形NotATriangle”。

（b）如果三条边相等，则程序输出为“等边三角形Equilateral”。

（c）如果恰好有两条边相等，则程序输出为“等腰三角形Isosceles”。

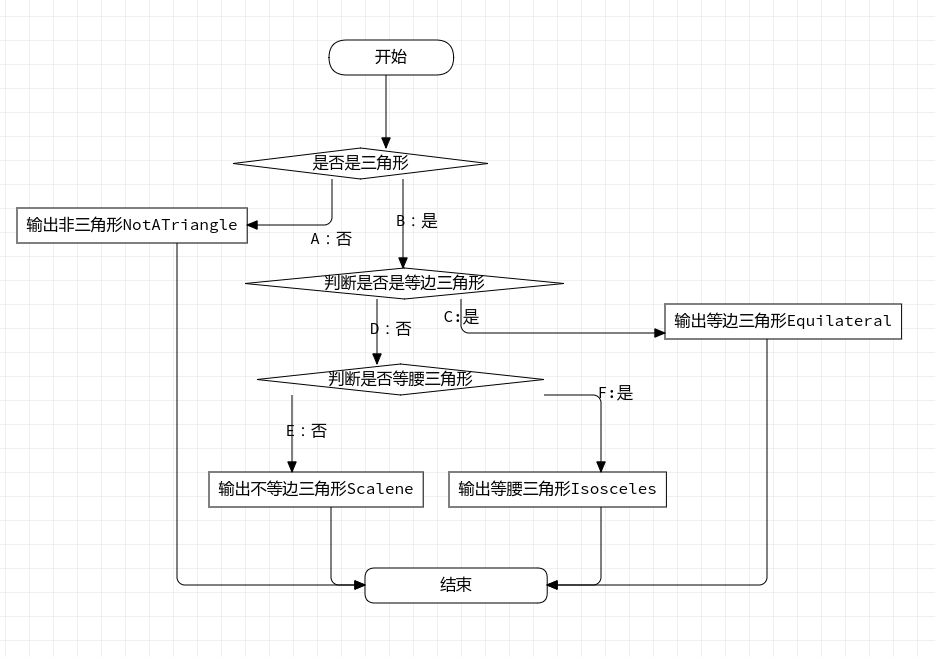
（d）如果三条边都不相等，则程序输出为“不等边三角形Scalene”。

完成的程序代码或截图：



（2）画出 judgeTriangle(int a,int b ,int c ) 方法的流程图并标注好路径以便填入步骤3表格（使用viso或者其他工具任选）。

流程图：.



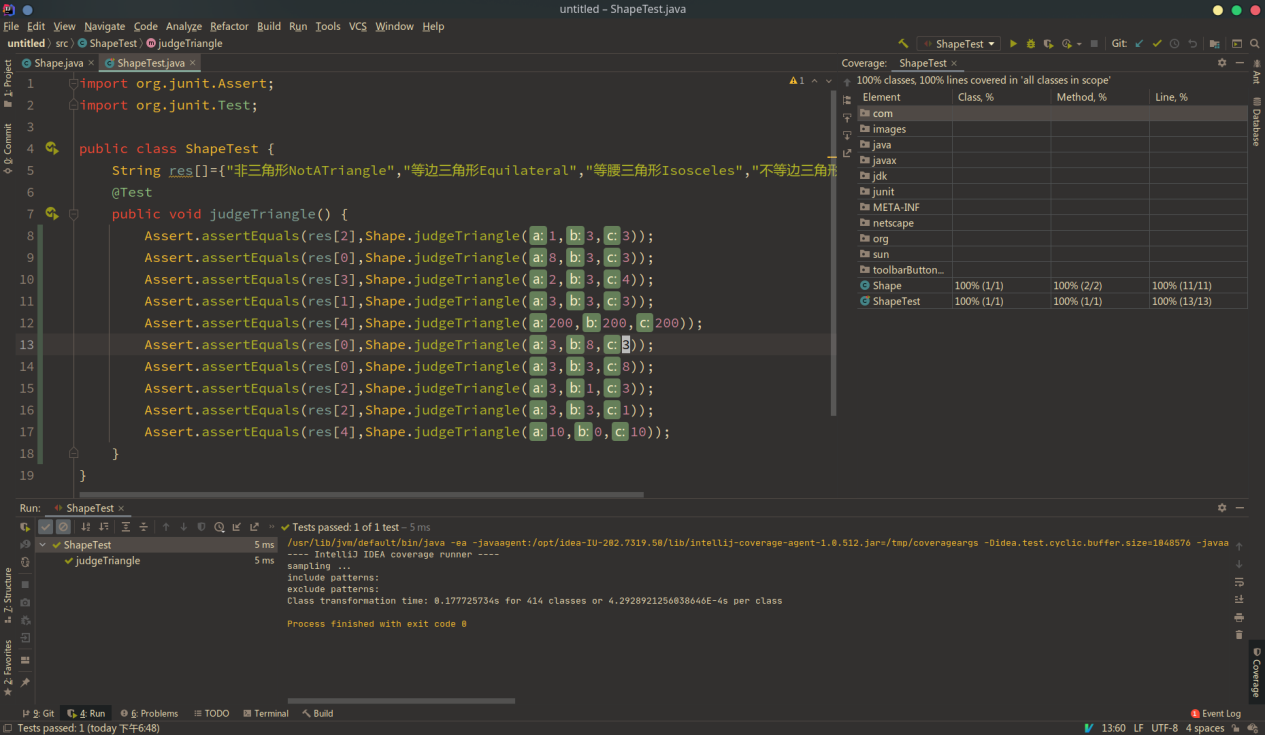
（3）针对judgeTriangle(int a,int b ,int c ) 方法编写白盒测试用例并填写在下面表格中，测试用例要求包含（语句覆盖，判定覆盖，条件覆盖，组合条件覆盖，判定组合条件覆盖，修订判定条件覆盖。每种覆盖根据自己的设计用例可能会有多行）。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **用例ID** | | FZU-S-TEST-LIB1-01 | | | | |
| **模块名称** | | 三角形判定 | | | | |
| **功能项（此列**  **按下文填写即可）** | **逻辑驱动标准** | **数据集** | **覆盖路径** | **期望的输出** | **实际情况** | **是否通过** |
| 三角形检测函数 | 语句覆盖 | a=1 b=3 c=3 | BDF | 等腰三角形Isosceles | 等腰三角形Isosceles | 是 |
| 三角形检测函数 | 语句覆盖 | a=8 b=3 c=3 | A | 非三角形NotATriangle | 非三角形NotATriangle | 是 |
| 三角形检测函数 | 判定覆盖 | a=2 b=3 c=4 | BDE | 不等边三角形Scalene | 不等边三角形Scalene | 是 |
| 三角形检测函数 | 判定覆盖 | a=3 b=3 c=3 | BC | 等边三角形Equilateral | 等边三角形Equilateral | 是 |
| 三角形检测函数 | 判定覆盖 | a=200 b=200 c=200 | A | 数据不在范围内 | 数据不在范围内 | 是 |
| 三角形检测函数 | 条件覆盖 | a=3 b=8 c=3 | A | 非三角形NotATriangle | 非三角形NotATriangle | 是 |
| 三角形检测函数 | 条件覆盖 | a=3 b=3 c=8 | A | 非三角形NotATriangle | 非三角形NotATriangle | 是 |
| 三角形检测函数 | 条件覆盖 | a=3 b=1 c=3 | BDF | 等腰三角形Isosceles | 等腰三角形Isosceles | 是 |
| 三角形检测函数 | 条件覆盖 | a=3 b=3 c=1 | BDF | 等腰三角形Isosceles | 等腰三角形Isosceles | 是 |
| 三角形检测函数 | 条件覆盖 | a=10 b=0 c=10 | A | 数据不在范围内 | 数据不在范围内 | 是 |

2、Junit 单元测试

（1）在当前的java项目 test文件夹下建立单元测试类 ShapeTest.java 在该类中对应上文的测试用例编写相应的测试方法，使用Junit执行以上测试方法（有兴趣的同学可以使用 maven 批量执行单元测试）。

完成的单元测试类程序代码或截图：



**实验总结：没有遇到什么问题**