

# 第十三次作业指导书

## 一、 作业内容和要求

### 1. 作业内容

本次作业的内容基于第三次作业的需求（即支持捎带的单电梯系统），以及所完成的设计和代码，需要完成的任务包括：

- 1) 根据第三次作业中的电梯相关功能和性能要求，按照JSF规范来补充关于电梯、调度和请求队列的相关类的规格，确保规格与代码实现逻辑一致。
- 2) 针对代码实现中的每个类，使用Junit4编写自动化单元测试代码，可以使用相关工具自动生成相应的框架代码，甚至是测试数据。
- 3) 对于补充了规格的类，要求按照规格来设计相应的测试用例。
- 4) 要求对第三次作业被报告的bug（公测和互测）编写测试用例来检查和确认bug是否继续存在或者已修复。
- 5) 执行所有设计实现的所有测试用例，并报告自己所发现的bug，并提供累积的覆盖率统计，并要求语句覆盖率为100%，分支覆盖率不低于85%。

### 2. 作业要求

- 1) 鼓励对第三次作业的代码进行重构，但也可以不进行重构。
- 2) 不论是否重构，都必须在本次作业的git库中commit第三次作业的代码，并在同一个项目中编写测试代码。注意，我们仍然会对本次作业的代码（包括第三次作业的代码和本次编写的测试代码进行查重）。
- 3) 必须逐个整理第三次作业中被报告的功能型bug，而且每个bug至少要有有一个测试用例来试图复现它，复现结果可能包括：
  - 如果未对代码重构，一定能够确认bug仍然存在某个或某几个方法中；
  - 如果重构了，bug可能已被修复，或者仍然存在某个或某几个方法中；
  - 注意，第三次测试所报告的bug是功能性bug，意味着多个类和方法中都有与相应功能bug相关的代码bug，构造测试用例的目标是展示导致相应功能执行不正确的代码层次bug；
  - 务必在试图复现bug的测试用例处使用注释标清楚所针对的功能bug（即

测试被报告的bug)。

注：针对一个被测方法而言，所谓一个测试用例就是针对该方法，带有@Test标签的那个测试方法。

4) 要求使用Junit4提供的功能来统计覆盖率，并提交原始的覆盖率统计文件；

5) 可不必对其中的get和set形态方法，以及第三次程序中的main方法编写测试代码。所谓get形态方法，就是简单return对象属性取值的方法（即无额外的分支处理）；所谓set形态方法，就是简单对象属性进行赋值的方法（即无额外的分支处理）。

## 二、 提交物与要求

1) 第三次作业对应的代码（不论是否重构）和所补充的规格；

2) 基于Junit4框架所设计实现的测试程序；

3) 针对所提交电梯程序的测试运行覆盖率报告；

4) Readme文本文件（注意避免泄露个人信息），要求至少包括测试用例及测试结果的概述（以列表方式给出每个方法对应的下述数据）：

- 所测试的方法名
- 测试用例数目
- 分支覆盖率
- 语句覆盖率
- 所复现的bug数
- 新发现的bug数

5) 上述所有提交物要求都存放在本次作业的git库中，作业系统将自动拉取相应的提交物。

## 三、 测试检查（互查）

1) 除get型、set型和main方法外，检查所有方法的语句覆盖率是否为100%，分支覆盖是否大于等于85%。如果不满足要求，则报告1个incomplete类型bug。

2) 基于所提供的规格或者代码的设计信息来补充测试用例，从而发现更多的bug：

- 如果所提供的测试程序已经发现了相应bug，不允许重复报告；

- 如果确认为由测试者自己所发现的新bug，则务必提交测试用例代码和相应的运行结果，按照wrong类型报告bug，注意不允许重复报告相同的bug；
- 如果提交的新bug报告中没有相应的测试用例代码，则自动视为无效bug报告，且不接受申诉。

#### 四、 有关自测与互测的说明

- 1) 每个同学通过 junit 测试程序找到自己第三次作业中的任何 bug 都不会导致本次作业被扣分；
- 2) 测试者所提交的任意一个 bug 报告，都必须是新的 bug，且是开发者自己未发现的 bug；
- 3) 会在服务器后台自动运行 junit 程序，对本次作业的覆盖率统计结果将进行复检，若确认与自测者所提交报告中的两种覆盖率出现较大差异，**判自测者作弊！**