

# Java 程序说明文档

## 1. 电梯调度策略和程序功能说明

本次作业设计了一套由 3 部电梯组成的多电梯调度系统，通过采用线程机制，在第三次作业所实现程序的基础上完成新的调度系统程序。本程序采用 ALS\_Schedule (A Little Smart Schedule) 调度策略和运动量平衡策略，在调度电梯完成一个（或一组执行时间有重叠）请求的过程中，电梯会响应所有满足“顺路捎带”条件的请求，同时 3 部电梯的累积工作量要尽量均衡，对“可捎带”和电梯运动量均衡策略的定义详见指导书。

## 2. 程序运行所需环境和运行指令规范

java 环境：

jdk1.8.0\_162 或 jdk-10

运行的 IDE：

Eclipse IDE for Java Developers

Version: Oxygen.2 Release (4.7.2)

Build id: 20171218-0600

在 eclipse 中导入 project 后直接运行 Main.java 文件即可，输入的字符串需要满足正确的格式否则会输出无效提示信息。

## 3. 合法输入说明

程序的输入在满足指导的所有输入规范的基础上作以下补充说明：

程序运行所需的输入由控制台输入，不允许使用文件输入。

本程序最多允许输入 50 行（包括有效请求，“END”和空行等其他不合法的请求），输入行数不足 50 行时程序会等待输入“END”来开始调度，输入行数等于 50 后程序将会关闭 System.in 不再读取输入，并直接开始调度并输出前 50 行的调度结果，不再等待“END”，故如果样例所给的输入多于 100 行时，程序仅会给出前 100 行的调度结果。

本程序每行最多允许输入 10 个请求，一行输入超过 10 个请求时程序仅会对前十条进行有效或无效判断，超出 10 条的部分直接视为无效并一次输出。对于空请求，程序

不会直接忽略，而是视为一个无效请求（如 `st:INVALID[T]`）。请求之间以分号为间隔，对于一行不超过 10 条请求的合法输入，请求的开头，结尾没有分号，如果加上分号，程序视分号另一侧为一个无效输入（空输入）；对于多个分号连续的情况，程序视相邻的分号之间为一个无效输入（空输入）。

对于直接输入 `END`，程序不会输出任何信息。

对于同质请求，程序在输入后马上会进行判断，判断为同质请求后输出时会加#号，格式见下文。

所有的输入中都可以含有空格（包括 `END`），程序会自动过滤空格，不报错。

对于两类请求的输入的楼层数，可以包括前零和“+”号，不包括“-”号，且楼层数 `floor` 包括前 0 在内最多为 8 位数字，数值为 1-20 之间。输入的电梯编号则只有 `#1`, `#2`, `#3` 三种形式，不支持前导 0 和正号。

#### 4. 非法输入说明

本程序会将包括但不限于以下几种输入情况视为非法输入：

- 1) 输入字符串中出现不合法字符，如 `'\t'`，`'\n'`，`'.'`以及中文字符等，故类似于 1.0 这样的请求楼层数为非法输入；
- 2) 输入字符串中合法字符不合理组合，如括号不匹配、加减号重复、逗号重复、一行输入过多请求等；
- 3) 输入字符串过少，如空输入，仅输入“`()`”等；
- 4) 输入字符串不合规定，如楼层数数字的长度超出指定范围，电梯编号格式有误等；
- 5) 楼层请求出现 1 楼有 `DOWN` 的请求，20 楼有 `UP` 的请求等；
- 6) 因其他原因导致程序崩溃。

#### 5. 程序输出格式说明

对于输出作如下补充说明：

- （1）对于本程序不能识别的无效请求，程序会立即给出错误提示信息：

`st:INVALID[request,T]`

（所有字符中间无空格，`request` 相对于输入时的字符串仅去掉空格，其他如前零、正号、括号、`'\t'`、`'\n'`、`'.'` 等字符都会原样输出）

- （2）对于每一条同质请求，本程序在处理时会对应输出：

**#st:SAME[request,T]**

（所有字符中间无空格，request 相对于输入时的字符串会去掉空格、前零、正号、括号等冗余信息）

（3）对于每一条合法请求，本程序在处理完成时会对应输出：

**st:[request,T]/(#电梯号,楼层,UP/DOWN,累积运动量,t)**

(t 是停靠时刻，即电梯刚到达目标楼层由运动转为静止状态，尚未执行开关门的电梯时间)

**st:[request,T]/(#电梯,楼层,STILL,累积运动量,t)**

(t 为考虑开关门用时后的时刻)

（所有字符中间无空格，request 相对于输入时的字符串会去掉空格、前零、正号、括号等冗余信息）

（4）对于在同一次开关门时刻完成的多个请求，或者对于同质请求和无效请求的输出，程序输出的先后顺序视多个线程的运行状态而定，具有一定的随机性，但不同的输出顺序不影响结果的正确性（可参考 issue#20）。

## 6. 补充说明

（1）第三次作业在公测及互测中并未发现 bug，因此没有对 bug 进行相关修复。

（2）本次作业为调度器，3 部电梯都各自新建了独立的线程来进行工作，主线程则处理输入请求，充当请求模拟器的角色，公共的队列 reqQueue 作为托盘供不同线程共享使用。对于某些需要等待的楼层请求，程序也各自新建了线程来处理。

（3）电梯时间以第一行输入为基准，不论该行输入是否有效。

（4）END 必须作为单独一行输入，分散在多行，或与其他请求混合在同一行，则视为无效输入。

（5）对于一个楼层请求，在同一时刻如果有多部电梯可以分配，程序一般会选择运动量较小的电梯，如果有多部相同条件的电梯可以分配，程序一般会选择编号较小的电梯（必须严格保证在相同的时刻有多部电梯符合要求，事实上由于多线程运行的不确定性，某些情况只能在理论上存在，比如实际运行时程序不可能精确保证多部电梯都在同一时刻突然空闲，总会有先后，所以边界情况下程序的调度结果不唯一，但都应视为正确结果）。

(6) 输入 **END** 进程不会立即结束，而是等待所以请求调度结束后再退出。

(7) 一条请求执行完后马上就会向文件输出结果，同时本程序也会在控制台输出结果，可以通过观察控制台输出来验证。

(8) 对输入时间的判定以输入时的  $t$  为准，例如 0ms 时输入指令(ER,#1,UP), 5980ms 时输入指令(ER,#1,UP), 6020ms 时又输入指令(ER,#1,UP)，因为可能在 6000ms 时实际第一条指令执行完成，尽管输出时显示两条(ER,#1,UP)的输入时间都为 6.0s，但仍有可能在同质指令的判断上出现不同结果，所以请不要将测试的重点放在这类边界上。

(9) 程序生成的 **result.txt** 文件在主目录下，采用 **utf-8** 编码，以 `\r\n` 为每一行结尾，在 **Windows** 系统下可直接通过记事本或 **SublimeText3** 打开。

(10) 在某些边界条件下，多线程的运行机制具有一定的不确定性，运行的结果也不能百分之百地准确预测，所以对程序的功能进行测试时请尽量避免可能出现不同结果的边界情况。

(11) 程序运行时 **CPU** 占用率较高，运行程序前请确保程序上一次运行的进程已经结束（可通过任务管理器查看），否则可能会出现卡顿或失去响应等状况。

(12) 仅允许输入英文状态下的字符，这一点要求测试者保证。