

# 面向对象第五次课程作业指导说明

## 一、程序功能

如指导书中所说，该程序设计了三部智能又智障的电梯，若输入正确，则按照规则自动输出哪条请求、所到楼层，运行方向，运行完成时间；若输入有误或者不满足题目限制要求，则输出 `System_time : INVALID` 和该条输入（去除空格之后的输入）；若出现题目中所说的同质请求，则忽略，并输出 `System_time : SAME+该请求`。

## 二、程序运行所需环境和使用规范

运行环境：与课程提供的 JDK 和 Eclipse 相同即可。

运行步骤：将全部.java 导入工程文件，从属 Elev 包，该工程文件中需要有 JRE System Library，之后运行 run Begin 文件，即可在控制台输入电梯控制信息。

## 三、正确输入格式说明

本程序中，可能产生正确结果的输入格式仅为：

`(FR,floor,UP/DOWN)` 或者

`(ER,#number,floor)` 或者

`END`

其中，`floor` 表示长度为 1-9 位的十进制数字（可支持+号、前导 0），`number` 为电梯号只可为 123，/表示几种情况均可，支持任意位置的空格（即 `E N D` 也会终止程序输入）。

1. 系统启动时间，非程序开始运行时间，而是第一条指令输入时间，不管指令合法与否。直到单独出现 `END`，之后不可继续输入。可以没有有效输入，直接 `END`，结果为空。

2. 应指导书响应，每行指令不能超过 10 条，并且不能超过 50 行，无视硬性要求导致程序可能输出错误将无法解释，但一般情况下超出部分会被当做无效输入。

3. 对于 `;` 的作用，分号前后将会被分割成两个请求，并且没有其他字符也会被当做空输入。比如，`;` 直接回车，会输出两个空 `INVALID`；`(ER,#1,1);` 会输出一个空 `INVALID`。

4. 对于边界时间问题，由于硬件设备计算速度等问题，不同硬件下可能结果不同，像助教团队所说，不应以之作为判断正误的标准。

## 四、输出结果情况及内容

1. 上述式子在满足题目要求、题意正确性的前提下，均会给出正确结果。

其中，正确结果是指，对于满足题意及正确性的输入，会判断并执行相应操作，并给出形如

ST:[Request,T]/(#Number,Floor,Direction,Finish\_time)的输出。

2.若输入时出现：超出规定范围、非法字符等，如会报 INVALID。

3.若不按格式输入，如（ ）、(xxx 、xxx) 等都会报 INVALID。

4.单行无输入，如直接 Enter，同属 3 中不按格式输入，会报 INVALID。

5.程序无有效输入，直接以 END 结束程序，什么都不会输出。

6.其余不符合上述正确输入说明要求的，且在输入阶段就可以判断的输入，均报 INVALID；相应的，同质请求会报 SAME。

7.输出顺序与指导书要求相同：对于不可捎带请求，按照输入顺序执行；对于可以捎带请求，按照电梯运行方向顺序执行，并且运动量少的电梯优先选择，可捎带电梯优先选择；对于同层可一起捎带请求，按照输入顺序同时执行。

8.对于巨量测试数据，考虑到 OJ 测试压力，限定有效请求不超过指导书说明，否则会报 INVALID，且结果不一定正确。

9.输出的 Request 说明：对于合法请求或同质请求，返回 Request 为最有效的部分，即不包含空格、前导零、正负号、括号等无用信息的请求，如输入为 (FR,001,UP)，样例输出为 [FR,1,UP]；对于非法请求，为只去除空格的剩余输入，(FR,-00 1,UP)，样例输出为 [(FR,-001,UP) ]。

10.由于同运动量时捎带为随机选择，在发现结果不同时希望能够动手理清一下，防止出现多种正确结果可能，毕竟多线程的事我们也说不清楚 TAT。

## 五、类的介绍

1.Begin 类：属性为电梯、楼层、控制器、请求列表，方法为信息输入，main 函数用于程序的执行。

2.Controller 类：属性为现在请求、现在时间、之前时间等，方法为执行下一个请求、控制电梯并输出结果、忽略同质请求。

3.Elevator 类：属性为现在所在楼层，方法为生成请求、移动、显示当前楼层。

4.Floor 类：方法为生成请求。

5.Request 类：属性为标志、目的地、方向和请求时间，方法为返回标志、目的地、方向、时间。

6.RequestList 类：属性为存放请求的动态数组，方法为添加请求、返回请求、移除请求和是否为空。

7.Scheduler 类：继承 Controller 类，使用其属性，重载方法。属性为主请求，捎带请求，计数器等，方法为选下一个捎带请求、下一个主请求、判断同质请求、执行请求、改变电梯方向、删除请求等。

8.Elevator\_method 接口：规定了 Elevator 类应该具有的方法。

9.Element 类：用于存放常量。

10.Mult\_Scheduler 类：继承 Scheduler 类，但是基本 Override，用于分配请求给各个电梯。