

# 00 第二次作业要求

2017/3/10

## 1. 作业目标

通过设计一个简单的**单部电梯**运行控制系统，以面向对象的设计方式来实现电梯控制系统的具体功能要求。

## 2. 作业内容和成果物

### 2.1 业务背景

#### 1) 电梯定义

电梯是服务于规定楼层的固定式升降设备。垂直升降电梯具有一个轿厢，运行在至少两列垂直的或倾斜角小于  $15^\circ$  的刚性导轨之间。台阶式自动扶梯和滚梯不在本次作业考虑范围之内。

#### 2) 电梯的简单使用方法（不考虑紧急情况和复杂情况）：

载人电梯都是微机控制的智能化、自动化设备，不需要专门的人员来操作驾驶，普通乘客只要按下列程序乘坐和操作电梯即可。

- 1) 在乘梯楼层电梯入口处，根据自己上行或下行的需要，按上方向或下方向箭头按钮，只要按钮上的灯亮，就说明你的呼叫已被记录，只要等待电梯到来即可。
- 2) 电梯到达目标楼层停下开门后，先让轿厢内人员走出电梯，然后呼梯者再进入电梯轿厢。进入轿厢后，根据你需要到达的楼层，按下轿厢内操纵盘上相应的数字按钮。同样，只要该按钮灯亮，则说明你的选层已被记录；此时不用进行其他任何操作，一旦电梯门关闭之后，电梯就会开始运动，只要等电梯到达你的目的层停靠即可。
- 3) 电梯行驶到你的目的层后会自动开门，此时按顺序走出电梯即结束了一次乘梯过程。

### 2.2 作业电梯系统基本描述

- 1) 固定为 **10 层**楼，楼层计数采用中国式计数，即 1 层显示为 1，2 层为 2，依次类推，直至顶层显示为 10。

- 2) 每个楼层都有电梯请求按钮（下面简称为楼层按钮），告知乘客想要去的楼层与发出请求楼层之间的方向关系。如果乘客按下了向上请求按钮，则表示想搭乘电梯去往上面的楼层；反之，则表示想搭乘电梯去往下面的楼层。**规定最底层只有向上运行请求按钮，最顶层只有向下运行请求按钮，中间楼层同时有向上和向下两个方向请求按钮。**
- 3) 一个电梯箱体（剩余文档简称电梯）内有对应每个楼层的目的地请求按钮（下面简称目标），一般显示相应楼层的编号。
- 4) 此外，电梯还会有开门、关门按钮，以及其他的如报警处理等相关按钮。本次作业不要求在设计中体现这些按钮的设置和功能，但后续作业会逐渐引入这些按钮和功能。

## 2.3 电梯基本运行规则

- 1) **程序运行开始或重置时设置电梯停靠在一层；**
- 2) 一个楼层按钮同一时刻只能发出一个上行或下行请求，在电梯未执行完毕，即到达该楼层并开关门完毕前（**含开关门完毕时刻**），该楼层的多个同向请求只认为是一个请求（相当于实际生活中，电梯未到本楼层的时候，请求按钮变亮后，再按不会产生实际效果，但是发出上行请求后可以再发出下行请求，反之亦可，这视为两个不同的请求，执行完一个后另一个仍需执行）。当电梯开关门结束后，可以再产生新的上下行请求；
- 3) 电梯内的一个目标楼层按钮只能发出对应目标楼层的请求，一旦发出请求后，在电梯未执行完毕，即到达该楼层并开关门完毕前（**包括开关门完毕时刻**），目标楼层与该按钮楼层相同的多个电梯内请求被认为是一个请求。当电梯开关门结束后，可以再发出任意目标楼层请求。
- 4) 所有请求按照请求发出的时间顺序被电梯系统管理和调度，按照时间上的先来先服务策略（First Arrived First Served, FAFS）进行调度。
- 5) 如果电梯同时收到了电梯内请求和楼层请求时，则按照输入时的请求排列顺序执行。
- 6) 本次作业的电梯系统采用傻瓜式调度策略：不断扫描队列，按照 FAFS 策略取出待响应请求，只有当该请求被执行完毕之后，才会尝试调度下一个请求。**请注意此处不要按照常见的电梯运行调度策略进行优化设计，不要合并“顺**

路”的请求，如从 2 层去 8 层期间，未到 6 层时有 6 层的上行请求，应先处理完 2 层到 8 层的请求，再处理 6 层的请求；或者 2 层到 8 层时，中间又发出了去 6 层的请求，同样先执行完到 8 层的动作，再处理去 6 层的请求。

### 3. 作业内容和成果物

#### 3.1 作业内容

实现一个符合 2.2 节和 2.3 节所描述的电梯运行调度 java 程序。

#### 3.2 提交内容

- 1) java 语言程序 (java 程序文件)；
- 2) 程序说明文档 (Readme)，内容包括：
  - a) 电梯调度策略和程序功能说明；
  - b) 程序运行所需环境和运行指令规范；
  - c) 程序的输入说明，包括标准输入格式、输入限制和遇见输入错误时的响应信息；
  - d) 程序计算结果的输出规格，以及可预见的运行错误响应信息；
- 3) 程序中若干类的说明文档 (Readme, word 文件)，详见 4.4 节。

### 4. 作业要求和限制

#### 4.1 输入规范

用户输入为按照请求产生时间排序的请求序列（注意：如果两个请求产生时间相同，则谁在前面都是合法输入），序列通过字符串表示；

请求分为两类：一类是楼层请求，一类是电梯内请求。注意，本次作业不支持开门和关门请求的处理。

楼层请求格式为：(FR, m, UP/DOWN, T)，其中 FR 为标识，m 为发出请求的楼层号，UP 为向上请求，DOWN 为向下请求，T 为发出时刻。

电梯内请求格式为：(ER, n, T)，其中 ER 为标识，n 为请求去往的目标楼层号，T 为发出时刻。

括号内的逗号应采用 ASCII 字符集中的逗号“,”，而不是中文字符逗号“，”。请求之间可以使用空格、退格和换行分隔，这一点请在 readme 说明。请求内部元素之间可以有空格，要求程序能够自动过滤。

T 为请求产生的相对时刻（非负整数），第一个请求的 T 值要求设置为 0。设电梯运行一个楼层距离消耗时间为 0.5；达到楼层后停靠、开关门等一系列动作消耗时间为 1。合法的请求产生时刻为非负整数（建议为 32bit 整数），n，m 为 1~10 之间含两端的正整数。

不正确的标识符，不正确的方向，不正确的数字范围，多余的其他非允许字符，均认定为不合法输入，即无效输入。

特别地，对于 FR 标识符，1 楼的 DOWN 和 10 楼的 UP 也认为是无效输入。

至于是必须在一行输入全部请求，回车后电梯运行，还是通过特殊命令结束输入，在此并不做出限定，但是要求在说明文档中说明。附可能的输入样例（例 1）：

**(FR,3,DOWN,0)**

**(FR,1,UP,1)**

**(ER,1,2)**

**(ER,6,4)**

run

本次作业要求一次性将所有请求输入，然后执行程序进行电梯调度并输出结果。标准输入的请求是按照时间排序的，如果遇到一个乱序的请求，即请求产生时间小于前面一个请求产生时间，则该请求直接被丢掉，继续处理下一个请求。

要求程序能够忽略相同的请求，包括产生时刻相同的相同请求和产生时刻不同但是实质上相同的请求。详情见 2.3.2 及 2.3.3 中的规定。

例如（例 2）：

**(FR,3,DOWN,0)**

**(FR,3,DOWN,1)**

这里第二条请求发出时第一条请求还没有执行结束，相当于楼梯按钮仍处于按下状态，第二条请求相当于按下了按下的按钮，所以与第一条请求实质上相同。

如有下例（例 3）：

**(FR,3,DOWN,0)**

**(FR,3,DOWN,1000)**

第二条请求执行时，一般来说电梯已到 3 层，该请求相当于是同层请求，应执行一次开关门动作。对于电梯内请求的类似情况同理。

本次作业不要求楼层请求和电梯请求的顺序符合真实情况（所谓真实情况，即实际情况中电梯中如果没有乘客，不会产生电梯内请求，但本次作业做简化考虑，任何顺序的楼层和电梯内请求都可被接受）

本次作业不允许使用文件作为输入。

除红字表明的强制规定外，对于更多的细节的输入规范，如与文档冲突，请在 readme 说明，若没有说明且与文档的冲突，测试者有理由质疑。

## 4.2 输入方式

依旧为控制台或命令行输入，输入方式由程序设计者决定，但是要求在说明文档中加以明确说明。

## 4.3 输出规范

程序的输出为按照时间排序的电梯运动停靠楼层、停靠前的运动方向及停靠时刻（即电梯刚到达目标楼层由运动转为静止状态，尚未执行开关门的时刻）：

格式为：(n, UP/DOWN, t)

其中 n 为楼层号，UP/DOWN 为电梯运行方向；t 为相对于第一个请求发生的时间（浮点数）。

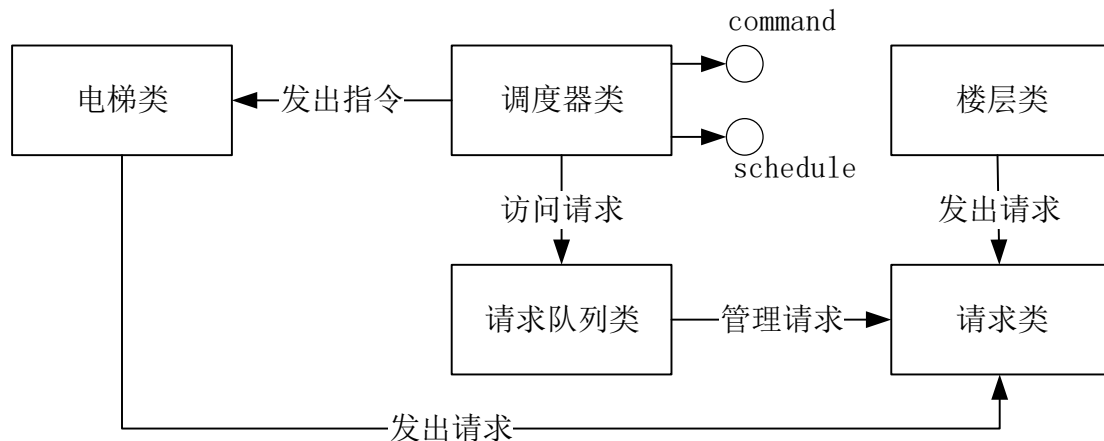
同层请求时输出为：(n, STILL, t), 此处 t 应考虑开关门时间。

输出格式要求采用 UTF-8 标准。

其他未规定的地方可由编程者自行决定。

## 4.4 设计要求

必须要实现电梯、楼层、请求队列、调度器、请求这五个类，且类中不允许出现 Public 属性。其中楼层类可能相对简单一些，不做具体要求，但是其他四个类的具体实现必须单独编写说明文件，内容包括类属性和方法的简要说明。



几个类之间的协作关系参考图

## 5. 其它说明事项

### 5.1 设计建议

- (1) 从输入读取请求
- (2) 构造请求对象
- (3) 加入到队列中
- (4) 启动调度
- (5) 记录电梯对象对请求的响应

### 5.2 错误处理

处理原则：

- 1) 如果发现输入请求序列 **不满足时间排序要求**，则输出 **提示输入有误**，并忽略不满足要求的请求，继续处理下一个输入请求。
- 2) 遇到无效请求（包括格式或内容不符合要求的），**提示为无效输入后**，继续处理下一个输入请求直至结束。
- 3) **格式有误或数据无效**(如楼层超过 10)的请求将被直接从输入请求序列中**拿掉**，不影响对其他有效请求的调度处理。
- 4) 任何情况下，程序都**不应 crash**，要正常结束（exitcode=0）。

### 5.3 Tips

3 个对象：电梯箱体、楼层、调度器，它们的定义是什么，有什么（属性）；它们能做什么，在什么情况下做（方法）。同理，分析另外 2 个内部管理类。

## 6. 其他规定

- 1) 文档中红色字体部分为强制要求。
- 2) 无效作业，以下四种情况视为无效作业。
  - (1) 程序不能编译和运行；
  - (2) 未使用 Java 语言；
  - (3) 所编制的程序不是本次作业的内容。
  - (4) 无法通过任何一个可以输出正常结果的公共测试案例；