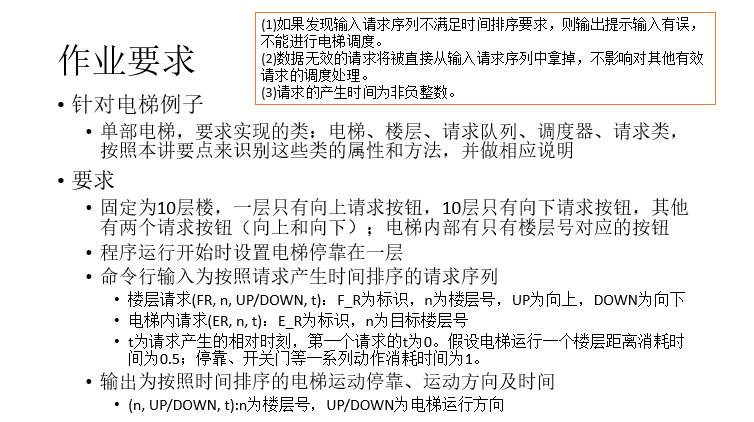
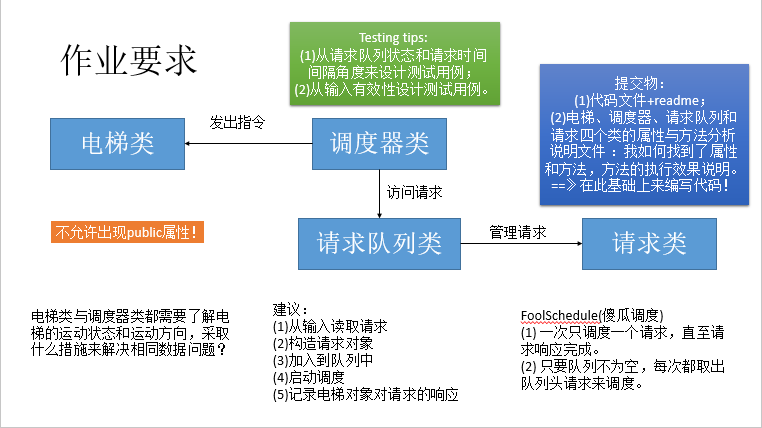
OO第二次作业要求

基本法镇楼！以下红色字体为强制规定，请各位同学注意！





# 输入规范

用户输入为按照请求产生时间排序的请求序列。

请求分为两类：一类是楼层请求，一类是电梯内请求。

楼层请求(FR, n, UP/DOWN, t)：F\_R为标识，n为楼层号，UP为向上，DOWN为向下。

电梯内请求(ER, n, t)：E\_R为标识，n为目标楼层号。

t为请求产生的相对时刻，第一个请求的t为0。假设电梯运行一个楼层距离消耗时间为0.5；停靠、开关门等一系列动作消耗时间为1。合法的输入时间为非负整数。

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。附韬神的输入样例：

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

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

**(ER,1,2)**

**(ER,6,4)**

**run**

**(ER,7,1)**

**(ER,2,90)**

**(ER,2,8)**

**run**

**reset**

**(ER,7,1)**

**(ER,3,+90.0)**

**(ER,2,-8.1)**

**(FR,+12,DOWN,100)**

**(FR,2,DOWN0,95)**

**(ER,7,1)**

**(ER,3,+90.0)**

**run**

**exit**

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.

# 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。

# 输出规范

输出实际上为电梯停靠的层数，即在每次电梯有运动转为静止时进行输出。

输出为按照时间排序的电梯运动停靠、运动方向及时间：

格式为：(n, UP/DOWN, t)

其中 n为楼层号，UP/DOWN为电梯运行方向

其他未规定的地方可由编程者自行决定。

# 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，不能进行电梯调度。

2. 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。

任何情况下，程序都**不应crash**，要正常结束（exitcode=0）。

# 其他强制规定

1. 固定为10层楼，一层只有向上请求按钮，10层只有向下请求按钮，其他有两个请求按钮（向上和向下）；电梯内部有只有楼层号对应的按钮
2. 程序运行开始时设置电梯停靠在一层
3. 必须要实现电梯、楼层、请求队列、调度器、请求类，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。（老师看）
4. 仅要求实现傻瓜调度，即一次只调度一个请求，直至请求响应完成。

由于时间仓促，此文档可能尚有疏漏。另如有其他问题，鼓励大家在讨论区中提问。你的问题可能也是大家的问题，老师们和课代表也好统一解决，谢谢大家。

2016/3/8