Readme

**1. 程序功能说明及电梯调度策略**

1.1本程序能够实现简单的单部电梯运行控制。

1.2 程序中各类的属性和方法

电梯共有六个类：Floor楼层类、Elevator电梯类、Request请求类、Requestlist请求队列类、Dispatch调度器类和EleSys电梯系统类。

楼层类：属性：boolean[][] floor，表示每层的UP和DOWN按钮是否被按下。

方法：无方法。

电梯类：属性：int PresentFloor，表示当前楼层。

Boolean[] floorbutton，表示电梯中每层的按钮是否被按下。

方法：各属性的基本get、set方法。

请求类：属性：int ERorFR,，表示请求是ER还是FR；0代表ER，1代表FR。

int UPorDOWN，表示请求是UP还是DOWN；1代表UP，-1代表

DOWN。

int m，请求为FR时的目标层。

int n，请求为ER时的目标层。

double T，发出请求的时刻。

方法：各属性的基本get方法。

返回值为Request的ParseRequest方法，逐行解析输入的字符串，判断

是否为合法输入，若合法就通过构造函数生成新的Request。

请求队列类：属性：ArrayList<Request> List，由许多Request构成的ArrayList数组，

用于添加和删除请求。

方法：返回值为Request的getRequest方法，返回指定的请求。

返回值为int的getsize方法，返回请求队列长度。

返回值为空的Add方法，用于在队尾添加请求。

返回值为空的Remove方法，用于删除指定位置的请求。

调度器类：属性：double Tsend，请求发出的时刻。

double Trun，电梯执行请求所需的时间。

double Tend，请求执行结束（开关门之前）的时刻。

int floor，请求的目标楼层。

int state，请求的运行状态，1代表UP，-1代表DOWN，0代表STILL。

方法：各属性的基本get方法。

电梯系统类：属性：无属性。

方法：返回值为int的JudgeIfSimul方法，用于检验无效请求。

包括了main函数，控制整个程序的运行。

1.3 本程序的主要思路如下：

包括main函数的电梯系统类逐行读入字符串，用请求类的ParseRequest方法解析，如果合法就生成一个请求，用请求队列类的Add方法加入到请求队列中。得到请求队列后分别在main函数中删除掉时间不合法的输入和无效输入，得到新的请求队列。遍历新的请求序列并生成调度器序列，判断如果本次请求的发出时刻Tsend晚于上次请求的结束时刻Trun+1，则将Trun+1的值赋给Tsend，从而可以正确得到每个请求执行结束的时刻。最终输出运行结果。

**2. 程序运行所需环境**

编写时使用了Eclipse。

**3. 输入规范**

3.1 用户输入为按照请求产生时间排序的请求序列（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；只能由控制台或命令行作为输入，不允许用文件作为输入。

3.2 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。

3.3 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“，”。请求之间只能使用换行分隔，输入“run”后电梯运行，如果一直不输入“run”则电梯一直不运行，可以接收新的输入。请求内部元素之间可以有空格。

3.4 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。输入的T值支持前导零，但不支持“+”号，包括前导零在内不得超过九位。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，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。请求之间额外的换行会被认为是非法输入而显示错误信息，即只输入回车将显示错误信息。

3.5 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。

3.6 本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（例如指导书上例2）。

3.7 全部请求（不包括run）不得超过1000行。

**4. 输出格式**

4.1 程序的正常运行输出结果：第一行为“The moves of the Elevator are:”，接下来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。

4.2 如果输入（不包括run）超过了1000行，将输出错误信息“There are more than 1000 lines of request.”

4.3 如果输入了非法请求（即不符合3.4和指导书的请求），将在第n行非法输入后输出错误信息"Line n of input is wrong."

注意：4.3的输出的位置与输入方式有关，有可能紧跟着每一行输入输出，也有可能在输入run后一起输出，具体请参考输入输出案例。

4.4 如果【合法输入】的第一行的时刻不为0，则输出错误信息“The first valid request doesn't begin at 0. The move is cancelled.”并直接退出程序，不运行电梯。

4.5 如果输入的请求中存在乱序的时间输入，则输出错误信息"The lines with time error have been removed."

4.6 如果在第n层同一时刻按下同一按钮，则输出错误信息“You pressed twice at the same button on Floor n.”

如果在电梯中同一时刻按下同一按钮，则输出错误信息“You pressed twice at the same button n in the Elevator.”

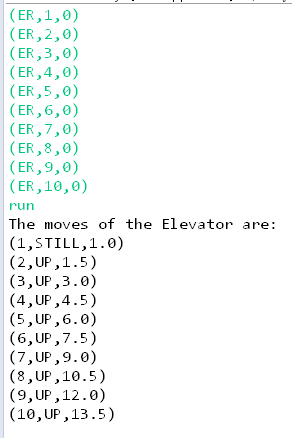
如果遇到了来自同一楼层（第n层）的产生时刻不同的无效输入（例如指导书上例2），则输出错误信息“The request from Floor n is invalid.”

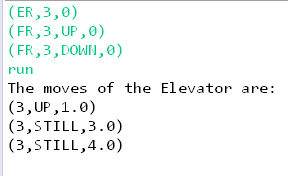
如果遇到了来自电梯的去往第n层的产生时刻不同的无效输入，则输出错误信息“The request from the Elevator to Floor n is invalid.”

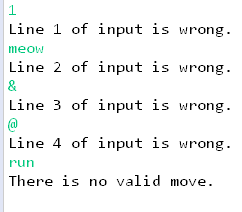
注：上述（4.6中）情况每检测到一条无效请求就输出一次错误信息，因此可能出现许多条同样的错误信息。电梯将忽略这些无效请求继续运行。

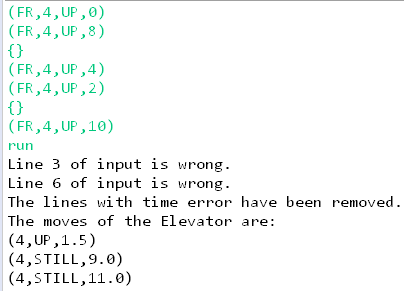
4.7 如果没有输入任何一条合法请求，则输出错误信息"There is no valid move."

1. **输入输出样例**

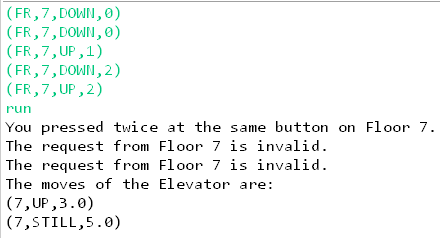
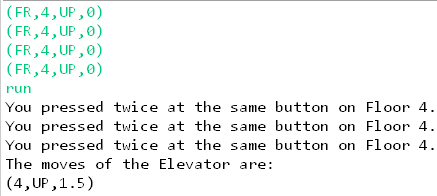
****

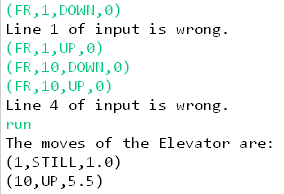
****

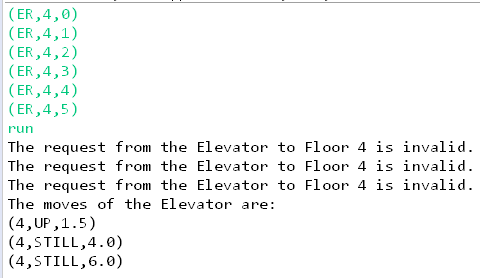
****

****

**1489502572(1)**

****

****

****