OO 第三次作业要求

2018

1. 作业目标

本次作业是在第二次作业（单部电梯傻瓜式运行控制系统）的基础上，增加一部分功能，通过引入继承机制实现具备更多功能的电梯控制系统，通俗地说，新增的功能是具备“捎带功能”。

2. 作业内容和成果物

2.1 作业电梯系统基本描述

此部分内容继承自第二次作业。

详细内容参考《OO 第二次作业要求》中 2.1 节第 1）、第 2）部分和 2.2 节的说明。

2.2 电梯基本运行规则

本次作业的电梯控制系统具备“捎带”功能，即电梯在去往目标楼层过程中，如果在电梯前进方向上有新的同向乘梯请求，控制系统需要判断是否响应该请求。

1） 程序运行开始或重置时设置电梯停靠在一层；

2） 电梯的状态定义：

a） **电梯运行状态**：从电梯启动时刻（此时速度>0），到电梯运动停止时（此时速度刚刚为 0）的运行状态，包括上行（**UP**）和下行（**DOWN**）状态。（启动时刻，运行停止时刻]，期间运行速度始终大于 0。

b） **电梯开关门状态**：电梯静止时，从门打开时刻（门开始动作），到门完全关闭时刻（门刚刚停止动作）时的状态。（门打开时刻，门关闭时刻]

c） **电梯停留状态**：电梯停在某层，且门长时间处于关闭状态。（门关闭时刻，门准备打开时刻或电梯准备启动时刻]

d） **STILL 状态**：此时电梯处于运行速度为 0 的状态。（电梯运行停止时刻，电梯准备启动时刻]

3） 一个楼层**同一时刻**只能发出**一个上行**或**下行请求**。电梯未到本楼层的时候，某个请求按钮被按下变亮后，再按不会产生实际效果（即不被响应）。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。在电梯到达某楼层处于开关门状态时，该楼层的多个同向请求只认为是一个请求。当电梯关门动作完成后（含门完全静止的那一刻），可以再产生新的上下行请求；

4) 电梯内的一个目标楼层按钮被按下后只能发出前往某个目标楼层的请求，一旦发出某个目标楼层请求后，在电梯到达该楼层并完成关门动作前（包括关门完毕时刻），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。当电梯关门结束后，可以再发出任意目标楼层请求。

5) 所有请求按照时间上的先来先服务策略（First Arrived First Served，FAFS）作为基本调度原则，具体含义是，在没有其它策略时，按照 FAFS 来响应。

6) 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。

7) 本次作业的电梯系统采用 ALS_Schedule (A Little Smart Schedule)调度策略：

(1)只要队列不为空，每次都取出队列头请求来调度（同傻瓜调度策略）；

(2)电梯在运动过程中不能突然改变运动方向；

(3)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，电梯要响应所有满足“顺路捎带”条件的请求，关于捎带的详细定义参考 2.3。

2.3 关于“顺路捎带”请求的说明

设电梯当前状态为 e=(e_n,sta,n)，即当前所处楼层为 e_n，运动状态为 sta（包括 UP，DOWN，STILL 三种状态），当前运动的目标为楼层 n，则：

(1) (e.sta = UP → 10>=e.n>e.e_n) || (e.sta = DOWN→1<=e.n<e.e_n) || (e.sta = STILL→1<=e.e_n<=10)[1]

注释：本段所述的可捎带条件可以这样理解：①电梯状态向上运行时，发出向上请求的楼层处于当前楼层和 10 层之间；②电梯状态向下运行时，发出向下请求的楼层处于当前楼层和 1 层之间；③在电梯静止状态，都可以捎带（1~10 层之间）

[1] 符号“→”是逻辑蕴含的意思

(2) 对于任意一个楼层请求 r=(FR, n, dir, t)，如果电梯当前是运动状态，则顺路捎带请求一定有：

(r.dir=e.sta) && ((r.dir=UP→(r.n <= e.n)&&(r.n>e.e_n)) ||

(r.dir=DOWN→(r.n>=e.n)&&(r.n<e.e_n)))

注释：本段所述的可捎带条件可以这样理解：电梯状态向上或向下运行时，新楼层请求的运动方向与当前方向一致，而且新请求发生的楼层在当前所处楼层和目标楼层之间。

(3) 对于任意一个电梯内运行请求 r=(ER,n,t)，如果是电梯当前运动状态下的顺路捎带请求，则一定有：

(e.sta=UP→(10=>r.n > e.e_n)) ||

(e.sta=DOWN→ (1<=r.n<e.e_n))

注释：本段所述的可捎带状态可以这样理解：电梯内请求的目标楼层在电梯的前进方向上。

(4) 对于 e.sta = STILL 状态分为 2 种情况。一是在电梯处于停留状态，此种情况没有“顺路捎带”请求，因为此时请求队列为空；二是电梯处于 STILL 状态但是非停留状态，这时满足(2)和(3)条件的请求，同样是顺路捎带请求。（**注意 STILL 状态仅用于输出表达，输入只有 UP/DOWN 两个状态**）

- 如有例 1：

  (FR, 1, UP, 0) //楼层 1 在 0 时刻发出上行请求。

  (ER, 8, 1) //电梯内在 1 时刻发出去 8 层的请求，此时在 1 层，预期在时刻 4.5 到达 8 层。

  (FR, 4, UP, 2) //楼层 4 在时刻 2 发出上行的请求，此时电梯向上运行且尚未到达 4 层，则调度器将该请求判断为顺路捎带请求，应做出响应，在时刻 2.5 到达 4 层，完成开关门 1s，预期在时刻 5.5 到达 8 层。

  RUN

- 在例 1 基础上有例 2：

  前 3 条请求如例 1，后续请求为：

(ER, 6, 4)　　//电梯内在时刻 4 发出去 6 层的请求，判断为顺路捎带请求。电梯在楼层 6 完成一次开关门动作，预期时刻 6.5 到达 8 层。

RUN

- 在例 1 基础上有例 3：

  前 3 条请求如例 1，后续请求为：

  (ER, 5, 4)　　//电梯内在时刻 4 发出去 5 层的请求，此时电梯已在时刻 4 达到 5 层，不能捎带，操作见(5)和(6)。

  RUN

- 在例 1 基础上有例 4：

  前 3 条请求如例 1，后续请求为：

  (FR, 5, DOWN, 3)　//楼层 5 在时刻 3 发出下行请求，但因方向不同，不能捎带。

  RUN

(5) 由于电梯不是傻瓜调度，应**满足所有“从主请求发出时刻起（包括发出时刻），到电梯到达主请求要求的目标楼层开门止（*但是不包括门打开时刻和开门过程时间*）的可捎带请求”**（所谓主请求，就是可以捎带其它请求的请求）。

(6) 一个请求完成后，其附带的顺路捎带请求可能未完成，此时**按照时间顺序**，将未完成的**最先顺路捎带请求，“升级为”**可以捎带其它请求的主请求（如果时间相同则取先输入者），并重新判断与其它请求的顺路捎带关系；没有顺路捎带请求时，此次响应过程结束。

1) 在例 1 基础上有例 5：

前 3 条请求如例 1，后续请求为：

(FR, 9, UP, 3)

(ER, 10, 3)

(ER, 9, 3)　　//在时刻 3 时，当前执行的主请求为(ER, 8, 1)，当前楼层为 4，因此(ER, 9, 3) (ER, 10, 3)均可被捎带，而(FR, 9, UP, 3)则不能被捎带。电梯到达楼层 8 响应完主请求

后，当前还有(ER, 10, 3)　(ER, 9, 3)两个顺路捎带请求未完成，因此根据规则选择(ER, 10, 3)为主请求，重新计算后(FR, 9, UP, 3) (ER, 9, 3)为顺路捎带请求，根据运行规则停靠 9 层、10 层后，此次响应过程结束。

RUN

(7) 电梯在某层处于开关门状态时，如果当前主请求与主请求的捎带请求集当中有多个当前层的请求，则可视为在一次开关门过程中同时执行。如果某请求不在该集合内，即使是当前层的请求，也不会执行。

如有例 6：

(FR, 1, UP, 0)

(ER, 10,0)

(ER, 4, 2)

(FR, 4, DOWN, 2)

(FR, 4, UP, 3)

RUN

在执行完第一条请求后，第二条请求成为了主请求，下面的请求中只有(ER, 4, 2)可以捎带，所以在 2.5S 时电梯到达 4 楼后，开关门 1S。继续在 6.5S 时到达 10 楼，停靠开关门 1S 后主请求执行完毕，此时应选择队列中未执行的第一条请求(FR, 4, DOWN, 2)作为主请求，此时队列中还剩(FR,4,UP,3)。但是该条请求不可被当前主请求捎带，因此电梯在 10.5S 时到达 4 楼，开关门使用 1S 执行完(FR,4,DOWN,2)，再选择最后一条请求(FR, 4, UP, 3)作为主请求，开关门 1S 后最后在 12.5S 时执行完全部请求。

3. 作业内容和成果物

3.1 作业内容

实现一个符合本文档第二章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；

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：可以输入时间相同的两个请求，排在前面请求被优先执行），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。

楼层请求格式为：**(FR, m, UP/DOWN, T)**，其中 **FR** 为楼层请求标识，**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。（注释：相当于请求者在楼道里的某楼层按“上行”或“下行”键）

电梯内请求格式为：**(ER, n, T)**，其中 **ER** 为电梯内请求标识，**n** 为请求前往的目标楼层号，**T** 为发出时刻。（注释：相当于人在电梯里按一个目标楼层号）

所有的逗号应采用 ASCII 字符集中的逗号“,”，而不是中文字符逗号“，”。**请求之间必须通过换行进行分隔，两条请求之间不允许有空行**。一条请求的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(4 字节非负整数)，**第一个请求必须为(FR,1,UP,0)**（若违反本限制条件将导致自动测试得到不正确的结果，从而拿不到分数！）。

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；**有效的 FR 请求会带来一个开关门动作**。合法的请求产生时刻 T 为**非负整数（4 字节非负整数，支持前导 0 和正号），1 =< n，m <= 10（正整数，支持前导 0 和正号）**。

不正确的标识符，不正确的方向，不正确的数字范围，多余的其他字符，均认定为不合法输入，即无效输入，无效输入不应被程序接受。

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。

在一行内只能输入一条请求，一行内输入多条请求将导致 OJ 无法识别！输入全部请求后，必须再键入 RUN 表示输入结束，回车后程序开始执行调度。

本程序中的 T 和 t 均为模拟值，即不要按照 T 和 t 的值作为真实时间来控制运行输出。

附一个正确输入样例（例 7）：

(FR,1,UP,0)

(ER,2,2)

(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**在输出该请求存在时间为乱序情况的提示信息后，将其从请求队列中删除，继续处理下一个请求。**（*Tip：本次作业时间上靠后的请求有可能被先响应！想想比较远的楼层先提出请求，而后比较近的楼层提出可捎带请求的情况*）

要求程序能够过滤掉相同的请求，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效果相同的请求。详情见**第二次作业指导书中**的规定。

被过滤掉的请求需要**输出该请求提示信息并从请求队列中删除**。

例如（例 8）：

(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 9）：

(ER,3,3)

(FR,3,DOWN,10)

(FR,3,DOWN,1000)

第三条请求执行时，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。

如果一条请求即可被当成捎带请求，又是相同请求，那么将被判断为相同请求，并在输出提示信息后从请求队列中删除。

如有下例（例 10）：

(FR,1,UP,0)

(FR,10,UP,1)

(FR,10,UP,3)

(FR,10,UP,6)

执行完第一条请求后，第二条请求成为主请求，此时第三条请求既为同质请求也为与主请求相同的请求，此时应判断为实质上相同的请求从请求队列中删除。第四条请求为与主请求相同的请求但是不可被主请求捎带，因此同样判断为实质上相同的请求从请求队列中删除。所以电梯仅在 10 层开关门一次。

本次作业不要求楼层请求和电梯请求的顺序符合现实场景（所谓现实场景，即现实生活中电梯内如果没有乘客，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**粗体**和**红字**表明的是强制规定，但只是强制规定的一部分。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

本次作业有两种输出：

1) 对于无效请求，要输出该请求为无效的信息，即使进行容错处理也要输出相应的字符串。

   格式为：INVALID［*request*］

2) 实质上相同的请求，要输出该请求为相同请求的信息

   格式为： SAME［*request*］

3) 每个有效请求执行完毕的输出请求内容和请求执行结果，分两种情况：

i. 电梯停靠信息为**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：[*request*]/(n, UP/DOWN, t)

本输出为一个对偶输出，前一部分是[*request*]，为有效请求的字符串，用“[]”包含。中间使用“/”分割。后一部分该请求的执行效果，包括： 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）。

ii. 同层请求时输出为：[*request*]/(n, STILL, t)，本输出为一个对偶输出，前一部分是[*request*]，为有效请求的字符串，用“[]”包含。中间使用“/”分割。后一部分包括： n 为楼层号，STILL 代表静止，**t 为考虑开关门用时后的时刻。**

例 11：

[FR,1,UP,0] / (1,STILL,1.0)

[FR,4,UP,2] / (4,UP,2.5)

[ER,8,1] / (8,UP,5.5)

iii. **如果一次停靠执行了多条请求，那么需要分行输出**

例 12：

[FR,1,UP,0] / (1,STILL,1.0)

[FR,4,UP,1] / (4,UP,2.5)

[ER,4,1] / (4,UP,2.5)

4) 输出要求按照请求执行完的时间进行排序。无效请求或实质上相同的请求也许输出信息，可以插在中间。

5) 输出格式要求所有字符为英文符号。

6) 其他未规定的地方可由编程者自行决定。

5. 其它说明事项

5.1 设计要求

1) 使用**继承**机制，不要覆盖或删除第二次作业的 scheduler 代码，重写(override)调度方法来实现捎带响应请求。

2) 使用 interface 来归纳电梯的运动方法。

3）**重载 Object 的 toString** 来获得电梯运行状态和时刻的观察。

5.2 测试要求

1）设计请求序列，重点检查是否违背电梯运动的两个原则；

2）针对新增的三个设计要求进行检查；

3）未实现，作为 incomplete 类 bug 报告，需要写清楚那个要求不满足，相应的类和方法也要报告。

5.3 处理原则

电梯在响应过程中需要捎带当前运动方向上能够响应的请求，完成响应之前不得改变运动方向，否则视为程序 wrong。

请求队列为空时，电梯不存在捎带关系。

5.4 错误处理原则：

1) 如果发现输入请求序列**不满足时间排序要求**，则输出相应不满足排序要求的请求（作为无效输入），并忽略该请求，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），**提示输出为无效输入后**，继续处理下一个输入请求直至结束。
3) 任何情况下，程序都**不应 crash**，要正常结束（exitcode=0）。

5.5 Tips

- 第二次作业的 bug 一定要修复，否则引入 ALS 调度后会导致产生很多新的 bug，甚至被报告第二次作业就存在的 bug（且不能进行申诉）
- 开发和测试时一定要抓住电梯状态和请求队列中的请求序列。

6. 其他规定

1) 文档中**粗体**和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；

（5）所提交的任何文档（程序文件、说明文件）内包含了编程者的个人信息。