程序运行环境说明

请使在搭载 windows10 系统的非 MAC 系列电脑上 JDK 版本 jdk-8u161-windows-x64 环境中，使用 eclipse-java-oxygen-2-win32-x86_64 运行该程序。如果以上要求不满足，运行时有可能出现未知错误。注：仅支持控制台输入，不支持使用文件输入和命令行输入。

程序功能说明

在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，即新增捎带功能。

电梯调度策略

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,到电梯运动停止时的运行状态,包括上行（UP）和下行（DOWN）状态。此状态电梯速度大于 0。（启动时刻，运行停止时刻）。
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。此状态电梯速度等于 0。[门打开时刻，门关闭时刻]。
c) 电梯停留状态（WFS 状态，Wait for Service）：电梯停在某层，且门长时间处于关闭状态。此状态电梯速度等于 0，且请求队列为空。（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]。（注意 STILL 状态仅用于输出表达）
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)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。

(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（WFS 状态），此种情况没有“顺路捎带”请求，因为此时请求队列为空；（注意 STILL 状态仅用于输出表达，输入只有 UP/DOWN 两个状态）

(5) 可捎带请求的先决条件是非同质请求。

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

(6) 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）。

(7) 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。

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

(8) 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。

输入格式及标准

1. 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，先输入的请求被优先执行），序列通过字符串表示；
2. 请求分为两类：一类是楼层请求，一类是电梯内请求。
3. 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)
4. 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)
5. 使用除以上两种以及 RUN 命令的任何多余或错误的输入，均会判定为无效输入。

6. 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，空行视为一条错误输入**。一条请求的内部元素之间可以有空格，程序能够自动过滤。
7. T 为请求产生的相对时刻，第一个请求的 T 值必须设置为 0，否则视为无效输入！等到第一个合法的 T 值为 0 的输入，才正式作为第一条指令接受指令。
8. 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，判定为无效输入。
9. 设电梯运行一个楼层距离的时间消耗为 0.5s；达到楼层后一次开关门动作时间消耗为 1.0s。
10. 合法的请求产生时刻为非负整数（要求最大为 4 字节的非负整数，关于符号和前导 0 请看补充事项中的叙述），n，m 为 1~10 之间（包含）的正整数。
11. 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。
12. 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。
13. 在一行内只能输入一条请求，一行内输入多条请求，整条被视为无效输入。将输入全部请求后，必须再键入 RUN 表示输入结束，回车后程序开始执行调度。 多于 100 的情况请看补充事项中的叙述。
14. **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，不考虑有没有人。**

输入示例：

(FR,3,DOWN,0)
(FR,1,UP,1)
(ER,1,2)
(ER,6,4)
RUN

输出格式标准

1. 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，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。此类输入为同质输入，同质输入在输出是会反馈如下：

   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。 t 是保留小数点后一位有效位的浮点

数，如果 t=3，则输出为 3.0；如果 t=0，则输出 0.0。

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。

2. 本程序要求一次性输入所有请求，然后执行程序进行电梯调度并输出结果。
3. 以上条目中的 request

   对于 INVALID 情况：去掉空格，非法字符符号和前导零等保留，即 request 为非法输入仅去掉空格的结果；

   对 SAME 情况即输入合法情况：输出去掉空格和左右括号，去除前导零，去掉不必要的符号，比如正数和零的符号不会输出。

   对于有效合法的请求，request 输出时去掉空格和左右括号，去除前导零，去掉不必要的符号，比如正数和零的符号不会输出。
4. 当出现特殊情况的时候，会爆出

#ERROR

输入输出补充事项

1. 所有无效输入在其回车后直接反馈 INVALID[request]。在第二次作业时，本人设计为：输入完毕后运转时，同质请求将在输出的对应位置输出。现由于捎带问题，现同质请求的#SAME 输出无明确含义。
2. 输入的请求时间和楼层必须为非零十进制整数，可以带或不带“+”，但是不能为负数。请求时间和楼层允许使用前导 0，但是应注意，符号以外的数字长度不得超过 50 位。
3. RUN 行中的空格也会被忽略，但是如果 RUN 行出问题会被划归为命令出问题（比如输入 run 或 Run，会爆 INVALID[request]）但不会停止输入，会继续等待输入正确的 RUN。
4. RUN（包括 101 强制转化为的 RUN）前面的输入均被判定无效与无输入直接 RUN 两种情况可以看作一种情况。映射到现实生活，可以理解为该电梯没人会用和无人使用，故 RUN 之后无任何输出（但是前一种情况中 RUN 之前的 INVALID[request]报错会正常报错）
5. 虽然禁止了输入空行（直接回车或只有空格的行），如果输入还是有空行，会被当作错误输入反馈为 INVALID[]，此外空行会算一条输入。即 6 中的 100 条限制，包含空行，空行也算作一条。
6. 包括 RUN 的输入总行数（正确和错误输入的总行数）<=100 时，可以完全准确的运行。当不包括的 RUN 的输入达到 100 时，电梯不会马上启动，输入第 101 行并回车后，电梯启动，从第 101 行输入起的任何输入系统不会做出响应。具体来说，如果你在 console

一条条输入，第 101 条是可以输入的，并且系统会直接将此条当作 RUN（即系统将强行将 101 行输入置为 RUN）。102 行将无法输入。对于不是一条条输入而是粘贴到 console 的输入。系统只对前 100 行响应，第 101 行视为 RUN。