面向对象电梯系列第一次指导书

摘要

本次作业，需要完成的任务为**单部多线程傻瓜调度（FAFS）电梯**的模拟

问题

电梯系统说明

本次作业需要模拟一个多线程实时电梯系统，从标准输入中输入请求信息，程序进行接收和处理，模拟电梯运行，将必要的运行信息通过输出接口进行输出。

本次作业电梯系统具有的功能为：上下行，开关门，每运行一层的时间为固定值，开关门的时间也为固定值。

电梯系统可以采用任意的调度策略，即上行还是下行，是否在某层开关门，都可自定义，只要保证在**系统限制时间内**将所有的乘客送至目的地即可。

电梯系统在某一层开关门时间内可以上下乘客，**开关门的边界时间都可以上下乘客。**

电梯运行基本概念

- 电梯系统时间：程序开始运行的时间
- 电梯楼层：1层到15层，共15层
- 电梯初始位置：1层
- 电梯数量：1部
- 电梯上升或下降一层的时间：0.5s
- 电梯开关门的时间：0.25s。开门0.25s，关门0.25s，共计0.5s，到达楼层后方可立刻开门，关门完毕后方可立刻出发。

电梯请求说明

本次作业的电梯，为一种比较特殊的电梯，学名叫做**目的选层电梯**（可以百度一下，确实是存在的，且已经投入了主流市场）。

大概意思是，**在电梯的每层入口，都有一个输入装置，让每个乘客输入自己的目的楼层。**电梯基于这样的一个目的地选择系统进行调度，将乘客运送到指定的目标楼层。

所以，一个电梯请求包含这个人的**出发楼层和目的楼层**，以及这个人的id（保证人员id唯一），请求内容将作为一个整体送入电梯系统，在整个运行过程中请求内容不会发生改变。

输入输出

输入

- 输入一律在标准输入中进行。
- 本次电梯作业使用以人为单位的请求模式，即包含一个人的id，出发楼层，目标楼层，出发楼层和目标楼层不能相同。

- 输入标准格式为`id-FROM-x-TO-y`，其中一条指令一行，id是人的唯一标识，x，y是属于楼层范围的一个整数，分别代表出发楼层和目标楼层。
- 课程组会下发对应的输入解析（ElevatorInput）程序，自动解析标准输入之中的内容并转换为`PersonRequest`对象返回给使用者。如果输入中有格式错误，或者存在id重复，或是出发楼层和目标楼层相同，解析程序只会读取到正确的请求，错误的将跳过并在标准异常中输出错误信息（不影响程序本身运行，也不会引发RUNTIME_ERROR）。具体的接口使用信息请参考**电梯第一次作业输入接口文档。**
- 本次的输入为实时交互，评测机可以做到在某个时间点投放一定量的输入。~~也就是传说中的，强制在线。~~

输出

- 输出一律使用课程组提供的接口进行输出，一条输出内容占一行，接口中会自动**附加输出时的电梯系统时间。**此外，任何其他信息**绝对不要**输出在标准输出中~~，更不要试图伪造假时间假输出蒙混过关~~，否则将会影响评测，且后果自负。
- 输出信息包括两部分，电梯的状态和人的状态
  - 电梯的状态，分为两种，OPEN(开门)，CLOSE(关门)
    - 开门格式：OPEN-楼层（在开门刚开始输出）
    - 关门格式：CLOSE-楼层（在关门刚结束输出）
  - 人的状态，分为两种，IN(进入电梯)，OUT(离开电梯)
    - 进入电梯格式：IN-id-楼层，这里的id是这个人自己的id
    - 离开电梯格式：OUT-id-楼层
- 具体输出时，使用者需要将格式字符串作为对应的参数传入官方输出接口中，接口内部会自动加上时间戳输出到标准输出。简单来说，和println使用方法大致类似，具体请参考**输出接口文档。**

样例

#	输入	输出	说明
1	[0.0]1-FROM-1-TO-2	[0.0100]OPEN-1 [0.0130]IN-1-1 [0.5110]CLOSE-1 [1.0120]OPEN-2 [1.0120]OUT-1-2 [1.5120]CLOSE-2	显然
2	[0.0]1-FROM-1-TO-2 [0.0]2-FROM-1-TO-2	[0.0120]OPEN-1 [0.0160]IN-1-1 [0.5120]CLOSE-1 [1.0120]OPEN-2 [1.0120]OUT-1-2 [1.5120]CLOSE-2 [2.0140]OPEN-1 [2.0140]IN-2-1 [2.5140]CLOSE-1 [3.0160]OPEN-2 [3.0170]OUT-2-2 [3.5160]CLOSE-2	电梯可以采用FAFS调度策略，因此先去2层送乘客1，再回到1层接乘客2，这样的结果会被判定为正确
3	[1.5]1-FROM-1-TO-2	[1.3870]OPEN-1 [1.3910]IN-1-1 [1.8930]CLOSE-1 [2.3940]OPEN-2 [2.3940]OUT-1-2 [2.8940]CLOSE-2	输入的起始时间可以不从0.0开始 **（此处存在时间计测同步性问题，在样例说明部分会有详细的说明）**

#	输入	输出	说明
4	[2.3]1-FROM-2-TO-5 [6.9]2-FROM-7-TO-10	[2.7070]OPEN-2 [2.7100]IN-1-2 [3.2080]CLOSE-2 [4.7090]OPEN-5 [4.7090]OUT-1-5 [5.2090]CLOSE-5 [7.8020]OPEN-7 [7.8020]IN-2-7 [8.3020]CLOSE-7 [9.8020]OPEN-10 [9.8020]OUT-2-10 [10.3020]CLOSE-10	起始楼层可以不是1层 **（此处存在时间计测同步性问题，在样例说明部分会有详细的说明）**

#	输入	输出	说明
5	[0.0]1-FROM-3-TO-5 [10.0]233-FROM-1-TO-7 [66.6]19260817-FROM-9-TO-2	[1.0110]OPEN-3 [1.0140]IN-1-3 [1.5110]CLOSE-3 [2.5120]OPEN-5 [2.5130]OUT-1-5 [3.0120]CLOSE-5 [11.8660]OPEN-1 [11.8660]IN-233-1 [12.3660]CLOSE-1 [15.3660]OPEN-7 [15.3660]OUT-233-7 [15.8660]CLOSE-7 [67.4690]OPEN-9 [67.4700]IN-19260817-9 [67.9690]CLOSE-9 [71.4690]OPEN-2 [71.4690]OUT-19260817-2 [71.9690]CLOSE-2	乘客id为int范围内的非负整数，电梯的运行总时间不超过200s **（此处存在时间计测同步性问题，在样例说明部分会有详细的说明）**

说明：

- 为了方便说明，指导书上提供的输入为这样的格式：`[x.xxxx]yyyyyyyy`。
- 意思是在`x.xxxx`这个时间点（相对于程序运行开始的时间，单位秒），输入`yyyyyyyy`这样的一行数据。
- 关于上面说到的时间计测同步性问题，在这里解释一下：
  - 直观地说，可能会观察到输入指令时间晚于做出反应的时间这一现象。
  - 这个实际上不是程序的错误，而是由于评测机投放数据的系统、和程序输出接口，这两边的时间可能存在不同步导致的。
  - **这一问题是由于多线程测试不可避免的波动性导致的计时同步性误差，属于正常现象，不会被认定为错误。**（~~当然，也不会出现故意提早的情况，因为程序本身就是实时交互，不可能做得到预知未来~~）

关于判定

数据基本限制

- 可输入电梯系统的指令数上限：30条。
- 电梯系统总执行时间上限：200s，这里的总执行时间是由课程组下发的`datacheck`程序确定，具体是指电梯执行完最后一条指令关门时的时间。
- 在互测时，为了防止边界情况的出现，规定互测输入数据的基准时间不得超过170s。
- 类似的，强测数据也会保证正常的程序不会超过时间上限，也**会避免使用对边界情况较为敏感的数据。**

正确性判断

> 电梯系统运行的正确性判断只根据以下三点：

- 电梯运行逻辑合理，即
  - 电梯在两个楼层之间的运行时间**大于等于**理论运行时间
  - 电梯开关门的时间**大于等于**理论开关门时间
  - 电梯**必须停下才能开门**，开门到关门期间，禁止移动，电梯**关门完毕后才可以继续移动**
  - 电梯在开门到关门的时间闭区间内，才可以进行上下人的行为
  - 而且，上下人行为的输出顺序必须介于开门和关门之间
    - 即对于这样的例子，即便输出时间上符合要求
      - `OUT-1-1`
      - `OPEN-1`
      - `CLOSE-1`
    - 也是属于错误的，`IN`、`OUT`**必须在一对相邻的**`OPEN`、`CLOSE`**之间**
- 人员运行逻辑合理，假设请求为`ID-FROM-X-TO-Y`，则
  - 发出指令时，人员处于电梯外的X层
  - 人员进入电梯后，人员将处于电梯内，且必须是**电梯外的当前停靠层的人员才可以进入电梯**
  - 人员离开电梯后，人员将处于电梯外的当前停靠层，且必须是**电梯内的人员才可以离开电梯**
  - 人员不在电梯内的时候，不会有自行移动的行为。即，可以理解为人员会一直待在原地
- 在电梯运行结束时，所有的乘客都已经到达各自的目标楼层电梯外，且电梯都已经关上了门。
- 故如果有程序运行结果正确但总时间超过了电梯系统总执行时间上限，或者未达到性能下限要求（对于部分中测数据点），那么会视为错误。
- **满足上述四条要求的一切调度结果都是正确的**

性能下限要求

- 我们以电梯系统的总运行时间作为性能指标。
- 本次电梯系统的参考调度策略为FAFS(先来先服务策略)，我们提供的`datacheck`模拟器就是使用这一策略，`datacheck`模拟出的运行时间作为基准时间T_{base}。
- 定义一个值，$T_{max} = \max(T_{base} + 5, 1.1 \cdot T_{base})$。
- 在中测时会有一部分标注出来的时间测试点，运行结果正确但总时间超过T_{max}的程序将也会视作未通过该测试点。
- 关于这个T_{base}的求解方式，即为根据输入来进行FAFS策略的模拟。不过值得注意的是，**我们会在模拟的时候随机加入一些扰动**（比如电梯运行一层的时间随机加长一定量，请求接收的时间也随机前后波动一定量等。**这些扰动在实际运行时也都是多少会存在的，不可能做到绝对意义上的精确，这也正是加入扰动的原因**），然后进行500次运行，取一个最大值，向上取整后作为基准时间T_{base}。
- 本次作业无性能分部分。

互测输入要求

互测样例输入数据需要额外附加时间项，评测系统将使用请求发射器在对应的时间将请求送入标准输入。

即互测时的输入格式为`[time]id-FROM-x-TO-y`，其中time是一个保留一位小数的浮点数，比如`0.0`,`1.5`,`2.2`等

互测样例输入基本限制总结

- 输入严格符合标准格式，不存在格式错误
- 整个输入序列中不存在重复的人员id，且人员id为int范围内的非负整数
- 整个输入序列中不存在出发楼层和目的楼层相同的请求指令
- 整个输入序列中的时间必须单调递增（不一定严格递增，但是禁止出现递减的情况）
- 满足指令数上限要求
- 互测时为防止特殊边界情况，限制样例的`datacheck`基准时间必须小于170s。
- 课程组下发的`datacheck`程序将会检测是否符合如上6条基本限制，如果不符合的样例将会禁止进入互测。

提示

- 如果还有人不知道标准输入、标准输出是啥的话，那在这里解释一下
  - 标准输入，直观来说就是屏幕输入
  - 标准输出，直观来说就是屏幕输出
  - 标准异常，直观来说就是报错的时候那堆红字
  - 想更加详细的了解的话，请去百度
- 关于FAFS调度策略
  - 指的是**按照请求进入系统的顺序，依次按顺序逐个执行运送任务**
  - 本次作业无需考虑实际电梯运行中顺路捎带等情况
  - 当然，以上只是建议，如果有自己觉得更好（更好写、或效率更高）的调度方式，欢迎自行探索和研究
- 关于这次的架构，可以采用这样的模式：
  - 主线程进行输入的管理，使用ElevatorInput，负责接收请求并存入队列
  - 开一个线程，用于模拟电梯的活动，负责从队列中取出请求并进行执行，执行完后继续取继续执行
  - 构建一个队列，用于管理请求，且**需要保证队列是线程安全的**
  - 以上只是建议，如果有自己觉得更好的设计，欢迎自行探索和研究
- 当然，对于很快秒了这次作业的大佬们，欢迎自行探索如何进行更高效的调度，之后会用得上的。

- 关于接口的一些使用，可以在idea里面，按Ctrl+Q。将可以看到类和方法的使用格式以及注释（本次官方提供的接口均提供了中文版javadoc注释，Ctrl+Q可以直接查看）