OO-第二次作业-Readme

2017/3/13

# 程序功能说明

模拟电梯运行，用傻瓜式调度策略。

# 程序运行所需环境

java环境：

java version "1.8.0\_73"

Java(TM) SE Runtime Environment (build 1.8.0\_73-b02)

采用的IDE是Eclipse Java Neon.

# 电梯运行规则

主要参考《OO第二次作业要求8.1》的2.3 电梯基本运行规则，**下**面强调几点内容。

* + 1. 允许一个楼层按钮同一时刻的上行下行请求。例如(FR,3,UP,0)

(FR,3,DOWN,0)

run

我们认为都合法并在第二条输出(3,STILL,3.0)。

* + 1. 如果有两条完全一样的指令，我们认为后一条出现的是可以忽略的，因此程序不处理也不输出。即使这两条指令不相邻。例如(FR, 9 , UP , 0)

(FR,9,DOWN,0)

(FR,9,UP,0)

(FR,9,DOWN,0)

run

则后两条请求都忽略。类似的，对于作业要求中2.3(2)、2.3(3)说明的实质相同的指令，也对后出现的采用这样的处理方法，即不输出直接跳到下一条继续执行。

* + 1. 第一条指令如果时间不是0则是无效输入，但程序不会立即退出。继续读入一直等到其都入到一个时间为0的或者读入终止。
    2. 规定读入的总请求数（即不包括run这一行）不能超过100000。如果请求 的总条数大于100000，则输出“Too Many Requests”并终止程序。

# 输入输出具体说明

## 3.1 程序的输入说明

大部分主要参考作业要求，**加黄色的内容为我自己强调说明的重要内容。**

用户输入为按照请求产生时间排序的请求序列（注意：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；

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。

括号内的逗号采用ASCII字符集中的逗号“,”，而不是中文字符逗号“，”。请求之间可以使用空格、退格和换行分隔。请求内部元素之间可以有空格。

如果读入是空行，则视该行为无效输入。

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。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（为32bit整数，即0~4294967296），n，m为1~10之间含两端的正整数。读入的数字允许有前导0。

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。

程序的输入通过特殊命令“run”来结束输入。逐行读入请求，每次读完一行请求后，立即判断该行的读入是否合法。如果不合法，则立即输出"Line i :Invalid input"，其中i代表第几行的读入有误。注意的是，这里判断的是读入是否合法，但对于合法但是多余的要求，此时不处理也不输出。

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，则该请求直接被丢掉，继续处理下一个请求。

## 3.2 输出响应信息

在读入的时候，每读入一行我们都判断一下这一是否合法，若不合法则立即输出"Line i :Invalid input"。

当读入完毕后，我们对合法的请求进行处理后输出。

对于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，我们忽略这条请求，但不输出（提示）任何东西。相当于跳过这一条处理下一条。

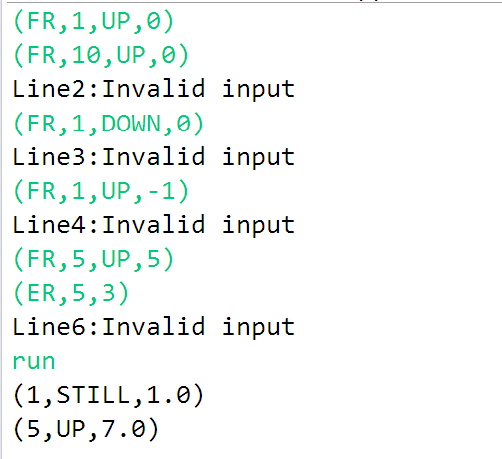
## 3.4错误处理

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”Invalid input”，并忽略不满足要求的请求，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），输出”Invalid input”后，继续处理下一个输入请求直至结束。
  3. 格式有误或数据无效(如楼层超过10)的请求将被直接从输入请求序列中拿掉，并输出”Invalid input”，不影响对其他有效请求的调度处理。
  4. **遇到需要忽略的请求，则直接忽略，不输出任何语句，继续处理接下来的指令。**
  5. 指令（即不包括run这一行）超过100000条时，输出“Too Many Requests”并终止程序。

## 3.3 输入输出示例

如果是一行行的读入，由于我是每读入一行，不合法的话就输出。因此当是一行行的读入的时候，不合法的错误信息与读入是混在一起的。例如下图：



如果是直接所有的复制粘贴过去，则输出大致如下：

