Readme

**1. 程序功能说明**

本程序能够实现模拟出租车的乘客呼叫与应答系统。与上次相比增加了统计车流量和开关路功能。

**2. 程序运行所需环境**

编写时使用了Eclipse。

**3. 输入规范**

3.1 在指导书的基础上做输入规范。

3.2 读入的地图文件路径默认为C://Users//Administrator//Desktop//map.txt，如需更改请到Uber.java第18行更改路径，注意每级之间只能用“//”连接。

如果地图文件每个点之间没有分隔符，则不做修改。如果用空格或制表符分隔，则需要在Map.java第45行在括号中添加空格或\t。

3.3 订单在控制台输入，输入格式为“[CR,(srcX, srcY),(dstX, dstY)]”，支持中间空格，支持前导零，支持数字前一个“+”号。。全部订单（包括无效订单）不得超过1000行。

3.4 一行可输入不超过10个有效订单，用“;”分割。由于解析订单算法较慢，只有同一行的订单才有可能被算作同一时刻的订单。（...）

3.5 可以从控制台输入指定出租车的序号（0~99）来查询该车的状态，输入的数值包括前导零不得超过两位数，否则算作非法输入。

3.6 可以从控制台输入指定状态来查询处于指定状态的出租车对象，输入格式为“StatusX”X的值**为1表示服务状态，为2表示接单状态，为3表示等待服务，为4表示停止运行**。除这四种外均为非法输入。

3.7【新增】可以从控制台输入指定请求来打开和关闭道路，输入格式为“[OPEN/CLOSE,(X1,Y1),(X2,Y2)]”，(X1,Y1)、(X2,Y2)为道路两端的点（顺序随意），**不支持空格**，支持前导零，支持数字前一个“+”号。输入的坐标值包括前导零不得超过两位数，否则算作非法输入。一次只能输入一个请求（一个OPEN或一个CLOSE）。

1. **输出规范**

4.1 如果目标路径地图不存在，控制台将输出“map.txt does not exist!”并直接退出。如果读入的地图文件不符合要求（每行80个数，数只能为0、1、2、3之一），控制台将输出“Invalid map!”并终止程序。

4.2 程序开始运行后，控制台出现“Ready for order!”（同时出现Gui界面）后才可以输入订单。

4.3 如果输入超过1000行指令，控制台将输出“There are more than 1000 lines of order.”并不再处理之后的输入。

4.4 如果输入不符合规范，控制台将输出“Invalid input: [输入内容]”；

如果输入能被判定为订单但是超出了地图范围，控制台将输出“Invalid order: [输入内容]”；

如果一行输入了超过10个有效订单，则对于10个之后的每个订单，控制台将输出“Invalid order(more than 10 orders in a line): [输入内容]”；

如果订单中的请求地点与目的地一样，控制台将输出“Invalid order(src=dst): [输入内容]”且不对该请求进行处理；

如果同一时刻在同一地点发出的去同一目的地的请求，视为相同请求，控制台将输出“Same order: [输入内容]”并忽略后一条请求。

4.5 当输入指定出租车的序号时，控制台的输出格式为：T: No.n at (x,y) status:s credit:c 其中**T为查询时间（单位ms），n为车的序号，(x,y)为车当前的位置坐标，s为状态（1表示服务状态，2表示接单状态，3表示等待服务，4表示停止运行），c为信用度**。

4.6 当输入指定状态时，控制台会将符合状态的出租车序号在一行内输出，用空格分开；如果没有符合条件的就不输出。

4.7【新增】当输入开关道路的指令时，

如果能被判定为订单但是超出了地图范围，控制台将输出“Invalid request: [输入内容]”；

如果输入的两个端点相同或者不是某条边的端点，控制台将输出“Invalid request(not a road): [输入内容]”；

如果输入的是OPEN指令，而两个端点构成的边已经打开，控制台将输出“The road is already opened!”；如果输入的是CLOSE指令，而两个端点构成的边已经打开，控制台将输出“The road is already closed!”

4.8 对于到文件的输出规定如下：

4.8.1每产生一个请求，都在D://testtaxi目录下新建一个命名为“T(srcX,srcY)-(dstX,dstY).txt”的文件。该文件需手动删除。(...)

4.8.2 输出请求发出时，处于以请求src为中心的4×4区域中的所有出租车状态、信用信息的格式为：（此时的信用度为接单之前的信用度）

**Taxi No.n: at (x,y) status:s credit:c**

4.8.3 输出在抢单时间窗内所有抢单的出租车的格式为：

**Taxi in compete: No.n at (x,y)**

4.8.4 输出系统选择响应相应请求的出租车的格式为

**Choose: No.n**

4.8.5 输出出租车响应相应请求过程中的实际行驶路径的格式为

**(x,y)->...->(srcX,srcY)** ←从车当前坐标到订单发出点的最短路径

**(srcX,srcY)>...->(dstX,dstY)** ←从订单发出点到订单目标点的最短路径

4.8.6 如果没有出租车相应该请求，则输出

**No taxi answer the order:(**

注：各数值定义同4.5

1. **Gui测试说明**

5.1 出租车为服务状态时，显示为蓝色；为接单状态时，显示为绿色；为等待服务状态时，显示为红色；为停止运行状态时，显示为黄色。

5.2 订单发出时会以订单发出的点为中心显示一个红色4x4的方格，但是请求消失后方格不会消失。

5.3 订单数量很多时，Gui界面可能会出现卡顿现象（出租车可能一次走好几格等）。

1. **上次作业被发现的bug**

6.1. 地图文件不存在时会crash。

6.2. 没有考虑输入指令的“+”。

6.3. 选择出租车的函数(ChooseTaxi)中，在删除序列中的出租车时导致了越界访问，但是try catch的时候没有printstacktrace，所以没有发现。

上述bug均已修正。

注：因为时间有限，jfs采用了中英文结合的方式。

求大佬轻扣qwq