放在前面：

readme中没有提到的一些细节，请以指导书和issue为准。上一次测试中我有一个bug是不能打开已存在的道路。助教明确了，**不打开本来不存在的道路这个是测试者来保证的，如果测试者不保证，只要不crash就行，所以这个bug是不存在的**。我已经把助教的澄清截图给他了，但是仍沟通无果，已经进入仲裁。希望同学不要再报这个，倒时候申诉时候扯皮没意义。

测试环境：

请使在搭载 windows10 系统的非 MAC 系列电脑上 JDK 版本 jdk-8u171-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（load属于文件调用）和命令行输入。

输入：

1. The system initing! 之后可以在sonsole输入Load 指令。
2. 系统加载地图等需要一定时间，请在"The system ready!"字样出现后，在输入请求。
3. Load指令在程序最开始进行初始化的时候在eclipse控制台输入，**必须使用且仅能使用一次**。如果其中的map部分出错会直接结束整个程序，其他的出错会进忽略。如果load文件名出错或不存在，会进行反馈，并进行等待，直到正确的load指令输入，再进行初始化。
4. Load的map必须是文件。
5. 本次指导书没有明确规定等的输入形式，也没有规定必须有默认灯图，所以没有准备默认灯图。请务必在Load的file中指定load的灯图文件 。
6. 再load之后仍然可以通过输入请求来增加订单。
7. 再load之后仍然可以通过输入指令来查看车的状态。
8. CheckTaxi指令以$开头，后面直接跟数字。不$开头的进入后面的判断。
9. CkeckTaxi指令输出的是出租车上一次移动后的状态，第一个时间是现实时间，第二个时间是上一个运动周期的出租车时间。
10. Load指令请务必按照正确的格式输入，如果里面的文件输入格式错误，后面发生的bug可能无法预料，但不会crash，也有可能直接触发系统shut down。
11. Flow的输入请严格遵守(x1,y1) (x2,y2) value的格式即value和第二个点之前都有且仅有空格，括号内无空格。具体可以参见文末的样例。
12. 请求的输入支持前导0，不支持前导+和负数，但是请不要输入>=80或<0的位置坐标。
13. 出发点目的地一致的请求算作无效请求。触发INVALID。
14. 除非您计算机性能特别强大，否则请不要尝试一次运行太多指令，毕竟理论上有可能计算不完。
15. 如果想要关闭程序只需要将GUI的窗口点击关闭即可，程序会直接自动关闭。
16. 由于本次测试不测格式，所以不是所有的报错都会有输出，有可能直接忽略而不显示，也有一定可能直接触发系统结束。
17. 程序的所有输入数值请不要大于100000，不要有前导+。
18. 总请求数大于300时，输入线程结束
19. 所有的文件名及路径中，请不要出现空格。
20. **默认地图的文件名为map\_ori.txt，如果##之间没有指定文件，则调用map\_ori.txt请保证此文件存在。否则当没有指定文件时系统结束。**

输出：

1. 输出时，为每个有效的（包括有无被派单的请求）建立一个日志（也就是一个文件）里面记录与该请求有关的信息。具体的路径为，与src文件夹和map.txt同级。
2. 输出格式请同学测试一组查看，格式很直观，恕时间紧张不再多做解释。大概顺序依次为：请求本身，曾经抢过该单的所有出租车，派单的情况，前往乘客的路程，搭上乘客前往目的地的过程，最终到达的时间
3. 输出的参与抢单的车的状态会被认为是指最后派单时候的状态。
4. 出租车移动过程在节点上中输出的状态，除了接到乘客后休息一秒结束的节点以外。均值的是在该节点进行完状态转换之后的状态
5. 出租车移动过程在节点上中输出的时间是绝对时间，所以都没有划归到100的倍数，其主要价值在于差值。
6. 乘客上車后（1s之后），才输出接到乘客的日志。到达乘客地点的时候不会产生日志。
7. 同质请求（100ms内的起点终点相同者）会算作一个请求计入输入的请求数，分派其一个ID，但是不会算数console中会输出同质提示，也不会为其建立日志。
8. 如果load的地方特殊，可能sonsole有一些输出优先于The system ready!。这从事情请不要计较，毕竟console本来也不在检查范围。
9. 幽灵GhostRequest所有车统一输出到request0.txt。但是由于整个过程不完整里面会缺失一些东西，请同学不要在意。
10. 为了防止信息输出不全，在request输出中，边界时间节点（状态进行变化）的输出会稍有一些累赘冗余。边界状态也不好把握，请同学见谅。

light行为解释：

1. Light必须在load中输入，**请保证此文件存在**，如果没有，系统自动结束。
2. 允许空格和tab
3. 所有灯的颜色一致。这种行为允许且合理，请勿扣分！
4. 等必须建在度大于2的节点上，即丁字路口和十字路口。否则输出错误，且不建立此灯。

出租车行为解释：

1. 休息时间不会被打断。
2. 出租车的20s一睡，由于等红灯大概率会延长，但是除去等灯，保证总时间是20s，即与指导书一致。
3. 判断抢单范围时是如果在半路上，视为在原路口才进行抢单。
4. 当出租车仍在某条路上（非路口）的时候，如果被派单，将继续完成此路段（即不会突然掉头）再进入服务状态，也因此派单时间（窗口关闭时间）和最终到达的时间的差很可能不是500的倍数。
5. 由于等灯，输出的两次节点信息之间可能不是500。
6. 如果出现幽灵乘客的现象，根据状态执行[CR,(20,0)(10,0)]请求，此条请求的编号为request0，在request0.txt中输出所有车辆的情况。（由于中途派单，故输出内容信息不完整。）同时，此条请求不存在分派等过程，永远不会消失。
7. 运行过程从道路塌陷，出租车将继续行驶至写一个路口。

SetRoadStatus行为解释：

1. SetRoadStatues时，请在console输入命令，格式请严格遵守#(x1,y1) (x2,y2) status的格式，即status和第二个点之前都有且仅有空，括号内无空格，具体可以参见文末的样例。
2. 只有第一个字符为‘#’时才会进入SetRoadStatues命令的判定，否则系统按照请求进行识别。输入不合法的时候会报错，同时请务必保证开闭之后整张图仍是连通的，否则会出现奇怪的错误，比如直接程序结束。（显然这已经超出题目的要求）
3. 新开通的路径不会影响已经进入服务状态的出租车的路线，其最短路径不会更新。
4. 车在接上乘客和接到的单发生效果的时候，以及碰到现有最短路径上出现断路的时候，才会重新计算最短路径。
5. 对于已经进入服务状态的出租车，如果其最短路径上某条路断了，只有当出租车到达该断掉的路径时，才会重新计算新的最短路。

派单行为解释：

1. 7.5s窗口结束后，派单时，如果一辆车处于等待状态之外的状态，即使他是最合适的或者唯一抢单的，也不会被派单。
2. 当信用和距离一致时随机选择队伍中的第一个（约等于随机）

GUI解释：

1. GUI不能对Load产生的初始流量进行显示（其固有缺陷），但是出租车会对初始化的流量进行响应，即第一个500s内车能够观察到初始化的流量。
2. GUI中不写JSF

写在最后：

人生艰难，OO不易，求同学手下留情OTZ。文明六系，和谐OO。

#map

map\_new.txt

#end\_map

#light

light1.txt

#end\_light

#flow

(11,2) (12,2) 5

(3,23) (2,23) 10

(25,10) (25,11) 15

(45,46) (45,47) 20

#end\_flow

#taxi

No.1 0 2 (1,1)

No.2 0 2 (2,0)

No.3 0 2 (3,3)

No.4 0 2 (4,4)

No.5 0 2 (5,5)

No.6 0 2 (6,6)

#end\_taxi

Consule输入样例：

请求：  
[CR,(48,8)(6,53)]

[CR,(16,56)(59,1)]

[CR,(27,77)(26,57)]

checkTaxi查询：

$5

$6

$7

道路通断：(这两条本来不一定是通的，取决于地图)

#(7,10) (8,10) 0

#(7,20) (8,20) 0