放在前面：

本程序将 100 个出租车放入一个线程进行运行。但是也满足了多线程要求（经过询问学长，这种方法是允许的，请同学不要误判，或误认为无效等，感谢！另外，readme 中没有提到的一些细节，请以指导书和 issue 为准。

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 必须是内容而不是文件。
5. 再 load 之后仍然可以通过输入请求来增加订单。
6. 再 load 之后仍然可以通过输入指令来查看车的状态。
7. CheckTaxi 指令以$开头，后面直接跟数字。不$开头的进入后面的判断。
8. 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。
9. 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。
10. 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。
11. 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。
12. 出发点目的地一致的请求算作无效请求。触发 INVALID。
13. 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。
14. 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。
15. 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。
16. 程序的所有输入数值请不要大于 100000，不要有前导+。
17. 总请求数大于 300 时，输入线程结束

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。边界状态也不好把握，请同学见谅。

出租车行为解释：

1. 休息时间不会被打断。
2. 出租车不会中途掉头，所以出租车的所有活动时间都在距离初始时间 200 的整数倍点上
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 200 的倍数。
5. 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。
6. 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. 3s 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）

派单行为解释：

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。

Load 样例
#map.txt##

#map
1223211333333321323113231311322333111131221222132313333131233331331113212333333312
1133112321312221313111211221322132211131231111211123322211131313111233132223211 2
3113222321232122112122131312132322311231222321313212212332331112321221323313133 2
1212111221311121113221132333121212333112223231321121311332313212222311222233333 2
3121322222231333312211212233232222211332213232333312211323111221112323222132131 2
3123332131212312333233322321333211233313133323331231221321321111213123322311313112
2123131113213212321331231113112111132333321311231212131122322222333133131332322 2
1231112322223122331333122222321111231131331122213233113231123313132333222312331 2
1221331313222122333111331232322112222331231133113212312222322123211121211213332 2
2223323131122133121322332332111311221123133311111133233213222213331333323122211 2
2313113321123133313132332321212221331311212231122232211231331232232221322212113 2

133311223313221232211313223312221331321212323213113132223132121313121113131
21122
113311112213131121233133122311321313112131132312122331131322221223113331
13122
313123112113122321233311323233221133133333323322313132321222331223313121211
33232
332333323231223112121132232111231321212223111332213332311232211223221223211
13122
322123331323121131322221123213212222322213212113211311232323121111113233231
12212
211222122332211231233331121321313131122232221311132312122123213332221133322
21232
331323122232132132213212111121132121231111112331323313123211112122321313212
11322
322133212213221333312211132323113331332223111331322131222321111322121313211
33212
223122313221313123223332113221231132313313131112121113231313331212113332311
22212
113311231133322111231233123123112113112113112212112231213231221311231111312
33132
322122123333212122221133233132312123221121311213222323123312132213131223233
13132
121222312313223211332231231322322132313121221221233331213221221231232132312
23212
112313123331222311122132211122222212123232232111211232133223212213221122312
23132
221321313223333221313333132332113231213313232132123313312233123311313313233
31332
231323221313332212122111111313112123222322123223121213111132313121123231332
12322
333211331311333312322132213122113211312113233131211231131212111333223231223
33112
332121332213131312131333322222133123223113111233321222332123232121133231233
22212
321322123121323313132231323212233122312312212321113312311311233131132333333
32322
121132212121131212221331213332111121231323211133212323312112232121331221231
23212
332133222123323311123322111323233231332231211132222123111213113332211213132
13232
223233111211213122213222233132231333131212333231111223232222131333111232333
23212
122211323313113223232332131333311313212233213232221212111231111221212131312
21312

211121233133132322111111212331113323132123222311111133213113323311132113321
31312
323321133323132313332121332122332322313121113132233331322333131113222123111
13222
112131122211212332122132323222311213332222311112111312121312211212233112122
33212
132123233121331232332313213312223212222232112231111313223232123321113113132
31232
233312213121231112113213111313312321322312213332221323121313111311322311232
11332
112331232313321123132123213223232111332312232321121212233321321311222111222
32232
131312112232133331111333323223233222322312122222213211331313331123113123312
11112
312311111312313332232221333232323312331113213122211312113112111311111131212
32112
232131122232233331112133322221223311122133322323221311122121232313123211213
11332
322213311231333323232122213333123233333333333233233312323123222213121233122
12112
213131111312321233111223213323222123122111112213222121323121322122111221121
12122
312132321331313113123233313332121122113333232132333231221213122223122123222
12132
111231131212221312222212123132112222222323221131132322311123231232212131122
11222
331231131231232312333221132233233321123112221212323333233233332111313112233 1
32112
232311233332212123232223312332112231323233113213223333333333312222113331222
12232
222111313332213311122223322313121331322221221321111121222333321211223221131
31132
233312231332321323212122213311121321123212322212132231323313323231212133321
33312
332233322122111321123211123223212112123331312122232221312321133321332123322
21222
112221223322223123132111311111232333221132213323112323313322223122323133111
12112
211113211221221212221211223111232133123312132331313132223111123323132212221
31312
112123131212332221122312133131331132232222212322323113323111312211321212221
21122
112121113332133112112112131232222123231233221123311333211323233323113111332
33122

221212211321332333233212223221113113122323333233321312332321132122123222112
11312
122233122112133311321221312132213113233331233123212212111223232111212133112
13222
122112133313333313322323313311322213311333121223221111112331133232223222233
32132
221111131133213233123133322222321233132212112221122332232223211212232231313
31332
332212121213311211131211322122112311312332211211223323321323322213131113213
33132
311312321311311123112313122331332322313131131132131321222321133332222121122
22132
332333113133232113132123131133133223122222331323223232322332231122113312311
33232
332112233312121212222323322213221121322332223311322133113331313231321133223
33112
133113122311221231331211122211222111112232311331211222333121223121321132213
23132
112331111111322232212111122332113121213333112232122223113131111221331121333
32232
113132213113333223333231113222222221131122121311133311313133111223321321113
31232
213213111312331212131222232112331132132113333211211222121313323321311332132
33322
113322331211333133223333123131123113322313332132222122211332133212231221122
23322
121212111321123123113322233212221333232232131213133123222113122122233133 1
11112
133312123233312232121131332122322113333212213112223132132232111333323323322
32132
333333121212123212323122123313121222132231111322332132121222111131232231213
11212
312331133333112232231131211111232322113122322122333321223122213331212121323
12232
122213323312233323111311223331312321212213133213132112321223231112212232131
13222
331233212232221333223231221212323121313132211232131321331332122112111112132 2
32132
112223333132231233111131311132313113321131132213311331321221113333212223333
21232
112322232222311122322133232111123323333232122111123123232123231133221212211
12112
312333131113211132213232311333131221312122221131231222131132133332223132222
33122

133333133323323231112233131332312221313133131212221121233311223321322112323
11332
111312232213111322111132113212232313212313321311123121312233132112231121333
33122
111111111111111111111111111111111111111111111111111111111111111111111111111
11110
#end_map

#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)
#(7,10) (8,10) 0
#(7,20) (8,20) 0