

README

1 程序功能说明

模拟出租车的乘客呼叫与应答系统。

2 运行环境

Eclipse Oxygen.2 (4.7.2)

java version "1.8.0_144"

运行主类 Main

3 一些说明

按照指导书要求，请测试者保持地图的连通性。

输入格式参照指导书，读入的请求信息同样按照指导书的要求，标准格式为[CR, src, dst]，其中 src 和 dst 均为(x, y)的形式，除了空格外不允许出现其它的字符。例如合法的请求：

[CR, (80, 80), (1, 1)]。

当输入的乘客请求被接受时，会在控制台提示“**Get request**”。
当输入的乘客请求被拒绝时，会在控制台提示“**Invalid Request**”。
当输入的乘客请求相同时，会在控制台提示“**Same Request**”。

对于一个被系统接受的请求当其未被相应时，会提示“**No Answer**”。

打开一条道路的标准格式为[OPEN, src, dst]，关闭一条道路的标准格式为[CLOSE, src, dst]，其中 src 和 dst 均为(x, y)的形式，除了空格外不允许出现其它的字符。控制台会输出一些提示信息，仅供参考。

车辆状态说明：**0**:停止 **1**:服务 **2**:等待接单 **3**:接单前去拉客。

每个被接受的请求的具体处理过程会在 **data.txt** 中显示。

输出文件样例：

```
-----
乘客请求内容----发出时刻：1524633230600 请求坐标：(5,7) 目的地坐标：(3,10)
窗口结束
无出租车接单
-----

乘客请求内容----发出时刻：1524633230800 请求坐标：(46,26) 目的地坐标：(32,19)
窗口结束
出租车信息----车辆编号：48 所在位置：(43,27) 车辆状态：2 车辆信用信息：1
出租车信息----车辆编号：64 所在位置：(48,30) 车辆状态：2 车辆信用信息：1
-----

开始派单
车辆编号：48 坐标：(43,27) 派单时间：1524633233831 乘客坐标：(46,26) 目的地坐标：(32,19)
-----
出租车48 途径坐标：(44,27) 途径时间：1524633233835
-----

乘客请求内容----发出时刻：1524633230900 请求坐标：(6,6) 目的地坐标：(2,9)
窗口结束
无出租车接单
-----

出租车48 途径坐标：(45,27) 途径时间：1524633234035
出租车48 途径坐标：(45,26) 途径时间：1524633234235
出租车48 途径坐标：(46,26) 途径时间：1524633234435
出租车48 到达乘客所在地时间：1524633234635
出租车48 途径坐标：(45,26) 途径时间：1524633235637
出租车48 途径坐标：(44,26) 途径时间：1524633235838
出租车48 途径坐标：(43,26) 途径时间：1524633236038
出租车48 途径坐标：(42,26) 途径时间：1524633236238
出租车48 途径坐标：(41,26) 途径时间：1524633236438
出租车48 途径坐标：(40,26) 途径时间：1524633236638
出租车48 途径坐标：(39,26) 途径时间：1524633236838
出租车48 途径坐标：(38,26) 途径时间：1524633237038
出租车48 途径坐标：(38,27) 途径时间：1524633237238
出租车48 途径坐标：(37,27) 途径时间：1524633237438
出租车48 途径坐标：(36,27) 途径时间：1524633237638
出租车48 途径坐标：(35,27) 途径时间：1524633237838
出租车48 途径坐标：(35,26) 途径时间：1524633238038
出租车48 途径坐标：(34,26) 途径时间：1524633238238
出租车48 途径坐标：(34,25) 途径时间：1524633238438
出租车48 途径坐标：(34,24) 途径时间：1524633238638
出租车48 途径坐标：(33,24) 途径时间：1524633238838
出租车48 途径坐标：(33,23) 途径时间：1524633239038
出租车48 途径坐标：(33,22) 途径时间：1524633239238
出租车48 途径坐标：(33,21) 途径时间：1524633239438
出租车48 途径坐标：(33,20) 途径时间：1524633239638
出租车48 途径坐标：(32,20) 途径时间：1524633239838
出租车48 途径坐标：(32,19) 途径时间：1524633240038
出租车48 到达目的地时间：1524633240238
```

接口说明：

在 Car 中可以通过以下方法得到信息

```

////////////////////////////////////
    public synchronized int getstate() {
        return state;
    }

    public synchronized Point getpos() {
        return pos;
    }

    public long gettime() {
        return new Date().getTime();
    }

    public synchronized int getcredit() {
        return credit;
    }
////////////////////////////////////

```

在 Main 中可以通过以下方法按状态查找出租车

```

public static Vector<Integer> statetaxi(int state, Car[] cars) {
    Vector<Integer> taxiId = new Vector<Integer>();
    for (int i = 0; i < cars.length; i++) {
        if (cars[i].getstate() == state) {
            taxiId.add(cars[i].getid());
        }
    }
    return taxiId;
}

```

关于文件测试接口说明：格式如 Load test.txt

大部分要求参照指导书

此接口只用于初始化

```

1  #Test File#
2
3  #map
4  #end_map
5
6  #light
7  light2.txt
8  #end_light
9
10 #flow
11 (0,0) (0,1) 3
12 #end_flow
13
14 #taxi
15 3,0,45,(0,0)
16 4,0,45,(0,0)
17 5,2,45,(0,0)
18 #end_taxi
19
20 #request
21 [OPEN,(0,0),(1,0)]
22 [CR,(7,7),(0,0)]
23 [CR,(15,68),(0,0)]
24 [CR,(57,42),(7,4)]
25 #end_request

```

文件内容及格式如上图所示。

其中#taxi 的指令格式为

No, state, credit, (x, y)

或者(不写位置时车位置随机)

No, state, credit

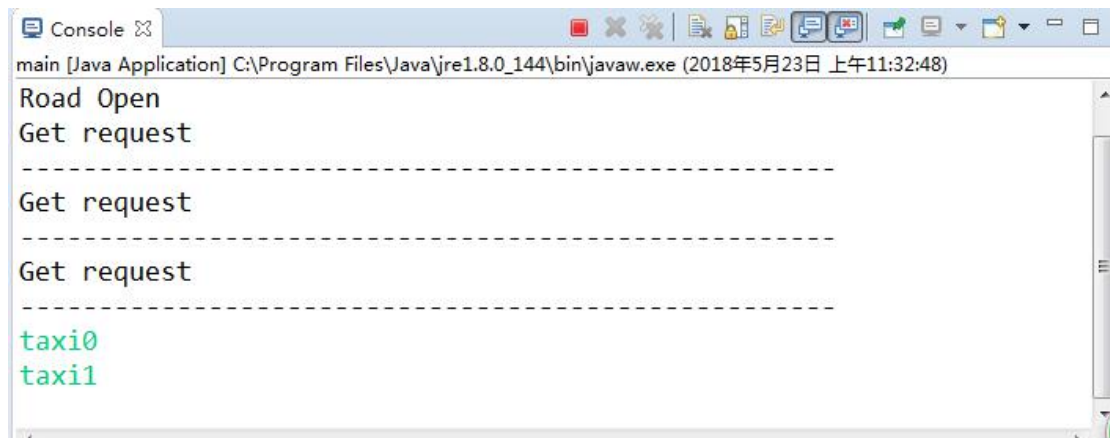
车辆状态说明：0:停止 1:服务 2:等待接单 3:接单前去拉客。

若在无接单随意设置出租车状态，出租车会罢工，并在控制台输出提示信息。

Gui 仅供参考，由于 **gui** 自身的原因，更改出租车位置和状态会有显示的延迟，更改基础流量也不会再 **gui** 中进行显示。清除流量的显示也不太对。

#light 与**#light_end** 间可以不填

需要查看可追踪出租车在控制台输入例如 **taxi0** 这样的即可输出至文件 **taxi0.txt**。(0-29 号为可追踪出租车)



```
main [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (2018年5月23日 上午11:32:48)
Road Open
Get request
-----
Get request
-----
Get request
-----
taxi0
taxi1
```

也可通过调用迭代器在测试线程中查看：

迭代器在 **Car** 类中，分别为

next() **hasNext()** **previous()** **hasPrevious()**

使用方法如下图，可自行在测试线程中调用

```
public void run() {  
    while(true) {  
        try {  
            // your code here  
            sleep(1);  
            if(cars[3].hasNext()) {  
                System.out.println(cars[3].next());  
            }  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

好人一生平安,望手下留情。