**设计思路概述**

本程序采用了多线程的设计。其中，输入类和调度类是线程。每辆出租车都是一个线程，能够根据状态独立地进行运动；道路类是线程，能够计算流量；但订单类不是线程。

首先从文件中读入地图文件并解析生成地图，读入红绿灯文件并解析生成红绿灯。从控制台读入字符串，解析并生成订单，加入订单序列。遍历订单序列，将抢单的出租车加入到订单的抢单序列中，3秒后从序列中选出一辆，修改其状态，然后利用BFS算法计算订单发出点到目标点之间的最短路径，使出租车按最短路径（遇到红灯时在原地等待直到遇到绿灯）行驶完成订单。对于每一个订单都将运行结果输出到相应的文件。还可从控制台读入相应字符串来进行出租车状态和符合状态的出租车的查询。

与上次作业相比，本程序的NewTaxi类继承了Taxi类，可追踪出租车与普通出租车相比，增加了能够在已关闭的道路上运行和可以追踪乘客服务情况的功能。

**各类及各方法的功能**

1 坐标类 Points.java

主要属性：点的序号，坐标，文件中的数值，上下左右是否有连线（boolean）

主要方法：Initial() 根据文件中的点的数值对点进行初始化，形成地图

2 地图类Map.java

主要属性：保存地图点的数组，每两点间最短路径距离的数组和方向的数组，

主要方法：Map ParseMap() 解析文件并生成地图对象

ArrayList<Step> BFS() 求两点间最短路径的方法

short[] BFSdis() 求两点间最短路径距离的方法

3 输入类 Input.java extends Thread

主要属性：订单序列

主要方法：run方法 解析控制台输入的四种输入（订单、查询出租车状态、查询符合状态的出租车和查询可追踪出租车的乘客服务信息）

4 出租车类 Taxi.java extends Thread

主要属性：序号，种类，当前坐标，订单，状态，信用度

主要方法：void runRandom() 处于等待服务状态时的运行方法

void runShortest() 处于服务或接单状态时的运行方法

int calculateFlow() 根据车的当前位置和上一个位置计算道路流量

run方法 出租车作为线程的运动方法

5 订单类 Order.java

主要属性：发出时间，发出地点坐标，目标地点坐标，抢单的出租车序列，用于输出的文件

主要方法：Order ParseOrder() 用于解析从控制台读入的字符串并生成订单

void JudgeTaxi() 判断出租车是否抢单的方法

int ChooseTaxi() 选择接单的车的方法

6 订单序列类 Orderlist.java 用于让输入类、调度类和出租车类共享订单序列

主要属性：Order的ArrayList

主要方法：添加、删除序列中订单的方法

7 调度类Dispatch.java extends Thread

主要属性：地图，所有出租车，订单序列

主要方法：run方法 遍历订单序列并分配给出租车

8 系统类 Uber.java

主要属性：系统时间

主要方法：main方法，初始化并开启所有线程

9 步骤类 Step.java

主要属性：最短路径其中一步的起始点和目标点的坐标、方向和道路流量

10 道路类 Road.java extends Thread

主要属性：道路是否打开，道路流量

主要方法：void initial() 根据地图对每一条路进行初始化

void ParseOpenRequest()和void ParseCloseRequest() 执行开关路请求

run方法 每200ms计算一次道路流量

11 红绿灯类 Light.java extends Thread

主要属性：变化间隔时间，路口是否有灯，灯的颜色，车是否能通过路口

主要方法：Light SetLight() 解析文件并创建红绿灯对象

run方法 根据时间间隔改变红绿灯的颜色

12可追踪出租车类 NewTaxi.java entends Taxi.java

主要属性：序号，种类，当前坐标，订单，状态，信用度，乘客服务信息序列，迭代器

主要方法：与Taxi类相同

13 Gui接口类 gui.java 提供gui接口

**上次被报告的bug**

出租车到达请求发出地点和到达目的地后不会停止服务1s（已修复）。

不能自动创建testtaxi文件夹（已修复）。

**关于新增可追踪出租车类满足LSP原则的论证**

LSP原则：子类能够完全替换父类，而不会让调用父类的客户程序从行为上有任何改变。只要父类能出现的地方子类就可以出现，而且替换为子类也不会产生任何错误或异常。

**·子类必须完全实现父类的方法**

首先我将Taxi类中所有属性从private改成了protected，使得NewTaxi可以继承父类的所有属性和方法。

在Uber、Input、Dispatch等类中，有一个属性是Taxi数组，存放了100辆出租车，这100辆出租车中包含了普通出租车也包含了可追踪出租车。在这些类中遍历调用出租车的时候，都调用的是Taxi数组，使用的是父类，将两类出租车“一视同仁”。因此，NewTaxi可以完全实现Taxi的方法，没有违背LSP原则。

**·子类可以有自己的个性：**

在NewTaxi类中重写了构造方法、repOK()、runShortest()和run()方法，并新增了几个属性和方法，没有违背LSP原则。