OO**第十次作业指导书**

# 作业背景和目标

本次作业在第九次作业基础上，进一步对出租车模拟系统的功能进行扩充，并重点训练类规格、过程规格的规范书写和相应的检查。

# 核心概念定义

## 城市地图

1. [新增]为交叉路口（包括十字交叉和丁字交叉）增加红绿灯（为简化问题，不设置黄灯，只有红、绿两种灯）控制。针对全部交叉路口，要求为不少于30%的路口添加红绿灯。对于未添加红绿灯控制的路口，出租车行驶规则保持不变。红绿灯控制的路口信息通过一个独立文件输入，文件内容为 80行字符串，每行有80个字符，每个字符为0或1（字符之间允许出现空格或制表符）。0表示无红绿灯控制，1表示有红绿灯控制。
2. [新增]所有路口的红绿灯变化间隔在初始化时随机确定，为500ms~1000ms之间的一个数值。每个路口有两组灯，南北方向为一组，东西方向为另一组。同组灯的颜色始终相同，不同组之间颜色始终相异。初始化时随机决定南北或东西方向的灯为绿色，然后每经过变化间隔时间，两个方向的灯同时变换颜色。

## 出租车

1. [新增]针对有红绿灯控制的路口，要求出租车必须遵守如下图所示和相应文字解释的行驶规则（不存在现实中的违规可能性）。图中假设此时南北方向为绿灯，东西方向为红灯，绿色表示可行驶路线，红色表示禁止行驶路线。其中由东西到南北的左转弯路线（如图中的5）与南北直行路线（如图中的4）出现交叉，属于可接受情况（因为我们没有专门设置左转红绿灯控制）。

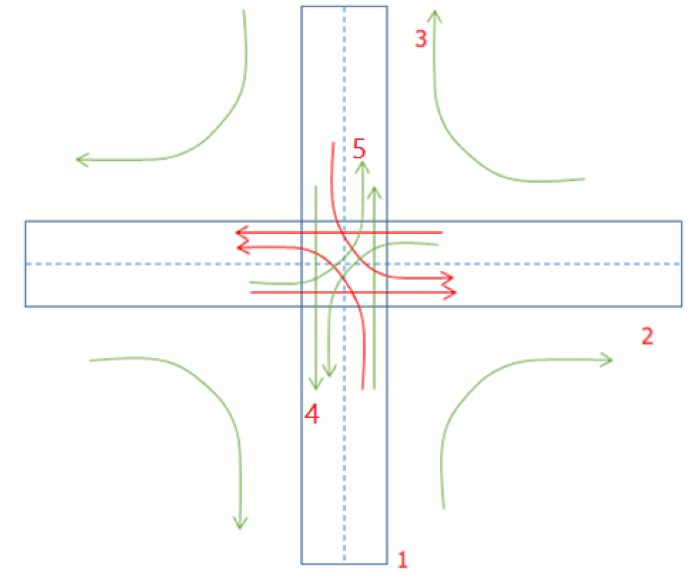


图1：红绿灯控制路口的出租车行驶规则示意图

* 1. 当出租车车头指向的灯为绿灯时，可直行；
  2. 当出租车车头指向的灯为红灯时，可左转；
  3. 不论出租车车头指向的灯为何颜色，均可右转时（如图中的四条右转弯线）；
  4. 处于路口等待的出租车，不存在故意延误时间不走的情形。

提示：出租车在搜索最短路径时依然采用之前的策略，无需考虑红绿灯变化带来的未来可能影响。出租车等红灯所消耗时间不纳入之前所定义的出租车状态转换中相关时间和一条边的行驶时间。举例来说，之前出租车行驶一条边为500ms，此时正好在这条边的一个端点处（假设正好是红绿灯控制的交叉路口）等红灯消耗了相应变换间隔时间（如600ms），则出租车实际行驶这条边所花时间为1100ms。

# 设计要求

## 对设计者的要求

1. [新增]要求使用一个线程类来模拟交叉路口的红绿灯控制。
2. [新增]要求每个类都实现repOK方法，且必须与不变式逻辑一致。

## 对测试者的要求

1. [新增]要求针对每个类都进行不变式满足情况的检查测试：每个方法被至少调用一次，在满足规格要求的前提下，测试不受限制。以方法为单位，只要方法执行后导致后置条件不满足或者所在对象的不变式不满足，记为一个wrong类型的bug。

## 建议和说明

1. 每个路灯的切换时间统一处理（每个灯的切换时间相同）
2. 不能使用Lambda表达式
3. 红绿灯处如果前进道路都被关闭，掉头不受限制
4. 红绿灯停车位置为红绿灯所处位置
5. 红绿灯一旦设置，不因为道路的开关而改变
6. 出租车等待红绿灯的时间不计入等待状态的20s
7. 内存问题：要确保整体100个车都能正常运行

# 关于GUI程序包的使用说明

1. 针对路口红绿灯，调用void SetLightStatus(Point p, int Status)来进行设置，其中Status取值：0 没有红绿灯 1 东西方向为绿灯（南北向红灯） 2 东西方向为红灯（南北向绿灯）

# 关于测试接口

1. [新增]命令行增加一个新的指令Load Filename（如：Load D:\test.txt），文件格式如下：

其中：#No 9 Test File#为文件标记符

#map和#end\_map中间是地图文件，地图联通由测试者保证，如果中间内容为空，则保持当前地图不变。

#light和#end\_light中间是红绿灯文件，红绿灯数量数量需要满足指导书要求。

#flow和#end\_flow中间是指定道路流量的基础值，其他的道路流量为0。如果中间内容为空，则保持当前道路流量不变。

#taxi和#end\_taxi中间是指定出租车的状态、信用和位置信息，编号0-99，出租车状态：服务状态取值为0，接单状态取值为1，等待服务取值为2，停止状态取值为3。未说明的出租车位置随机。

#request和#end\_request中间是同时发出的多个请求，格式与正常输入格式相同。如果中间内容为空，则不需要响应请求。

在文件指令结束之后，可以继续由控制台进行请求输入。

该文件格式正确性由测试者保证，不测试文件格式正确性

|  |
| --- |
| #No 9 Test File#  #map  #end\_map  #light  #end\_light  #flow  (x1,y1) (x2,y2) value  #end\_flow  #taxi  No. Status Credit (X, Y)  No. Status Credit (X, Y)  No. Status Credit (X, Y)  #end\_taxi  #request  [CR, (X1, Y1) (X2, Y2)]  [CR, (X3, Y3) (X4, Y4)]  #end\_request |