带流量出租车系统设计文档

1. 程序功能
   1. 程序能够实现带流量带红绿灯带特殊出租车的出租车系统，出租车能够根据流量判断运动路径遇到红绿灯可能会停下来(等待时间不计入出租车时间)。可以通过指令开启或关闭某条路径。特殊出租车无视已关闭的道路，但之前就不存在的道路仍然不能走。
   2. 关于程序计时的问题，程序会在启动时记录初始时间，此后的所有时间都是(当前系统时间 - 初始时间)/100ms，单位不好定。但出租车时间是独立的，以1ms为单位。
   3. Load测试文件指令并不是在程序初始时有效，程序初始时依然是随机设置出租车位置，程序启动后可以随时使用Load指令详细设置道路、出租车等的状态信息。
   4. 初始时，项目目录下必须有地图文件和红绿灯地图文件，文件名为map.txt和lightMap.txt。
2. 程序输入
3. 为了确保线程安全，程序在创建出租车线程时每创建一个线程都会sleep(10)，故请务必当控制台输出 “请输入乘客请求, 格式为 : [CR,(x1,y1),(x2,y2)]”，才开始输入请求
   1. 程序一共支持三种输入，可以实现三类功能
      1. [CR,(x1,y1),(x2,y2)] : 输入请求
      2. [ST,(x1,y1),(x2,y2),status] : status = 0 关闭道路 status = 1 开启道路
      3. Load filepath : 加载测试文件，filepath对应的文件必须存在，否则会报错
      4. TestStart：会启动测试类，从控制台打印特殊出租车记录的信息
   2. 格式检查
      1. CR类型指令 : 不支持空格和前置+，0 <= x1,y1,x2,y2 < 80，请求和目的地坐标不能相同
      2. ST类型指令 : 不支持空格和前置，坐标要求同上，两个坐标不能相同
      3. LD类型指令 : filepath对应的文件必须存在，格式必须同指导书中的要求，除了#map类型和#light类型外其他类型支持多条指令，多条指令每条指令占据文件一行，每一行有且只能有一条指令，格式如下：
         1. #map类型：必须是正确的绝对路径
         2. #flow类型：(x1,y1) (x2,y2) flow 中间有空格且不能少
         3. #taxi类型：No.id status credit (x1,y1) 中间有空格且不能少
         4. #request类型：格式要求同上CR类型指令，此处不支持格式检查！格式必须正确
         5. #light类型：必须是正确的绝对路径
      4. 项目文件夹下有样例测试文件this.txt，可以参考
      5. 项目文件夹下有样例红绿灯文件lightMap.txt，可以参考
4. 程序输出
   1. - 为了便于测试，程序会在当前目录下创建"log.txt"文件记录请求和抢单的所有出租车信息
   2. - 为了便于测试，程序会在当前目录下创建”detail.txt”文件记录出租车前往接单和送乘客到达目的地的过程的坐标和时间，此次作业仅对普通出租车有效，特殊出租车须通过测试类里的ite打印出信息
   3. - 每次测试前都会重置log.txt文件和detail.txt文件
   4. - 由于设计局限，派单的出租车信息会在控制台输出
   5. - 如果没有出租车可以响应请求，则会输出“无出租车响应该请求: 请求”
5. 如何测试
   1. 可以从控制台输入CR类型指令发出请求
   2. 可以直接编写测试文件并从控制台输入LD类型指令进行测试
   3. 测试类的使用：
      1. 测试类名为：TestCode.java
      2. 通过修改测试类中的全局变量taxiId来修改想要打印信息的出租车序号
      3. 测试类的构造函数不能修改，已构造好迭代器ite
      4. Ite的所有方法：
         1. Ite.hasNext()：判断是否有下一个
         2. Ite.hasPre()：判断是否有上一个
         3. Ite.next()：下一条记录
         4. Ite.previous()：上一条记录
      5. 测试类请自行编写，已通过注释给出一个简单的模板
      6. 由于无法预测测试者的行为，测试类不写JSF和overview
6. GUI的一些说明
   1. - 程序会积极的更新GUI界面，但由于GUI界面本身的缺陷会有卡顿现象，与程序本身无关
   2. - 为了确保GUI界面能够运行同时不致使程序崩溃，每次调用GUI界面的方法后都是使GUI界面进程sleep(100)
   3. - 你可以观察GUI界面来大概的知道运行结果，但还是要看log.txt文件内容和控制台输出获取具体信息
   4. - 由于GUI本身的缺陷，使用GUI不支持查看流量，你可以通过测试文件设置道路流量
7. 有关流量和道路和红绿灯和特殊出租车的说明
8. 程序并没有提供查询指定道路流量的方法，但是你可以通过编写测试文件直接设置道路的流量
9. 由于流量记录窗口过短，只有500ms，所以大部分道路的流量一般是0或1，可能会影响测试
10. 通过ST类型指令设置道路状态，道路状态改变不能改变地图的连通性。
11. 道路关闭后会通知出租车并重新检索最短路径
12. 所有红绿灯切换状态的时间间隔均为1000ms
13. 想要清晰的看到出租车是否受红绿灯影响，可以手动把红绿灯切换时间间隔调长，在TrafficLight类改变全局变量SWITCHGAP即可
14. 只有三岔路口和四岔路口可以有红绿灯，程序会自动过滤不能建立红绿灯的路口
15. 特殊出租车有30辆，编号为0-29且不会改变，从GUI里可以看到颜色为粉红色。
16. 关于出租车的一些说明
17. 出租车线程并不是同时启动的，为了确保线程安全，每启动一个出租车线程都会sleep，所以你会看到出租车并不是同步停止1s的(变黄)。
18. 你可以通过测试文件直接设置出租车的状态，但处于接单和派送状态的出租车状态请勿随意改变！
19. 出租车运行的时间间隔与系统时间无关，以1ms为单位，可以从detail.txt文件中看出
20. 由于出租车运行的不确定性，出租车并不是准确无误的每20s停止一次，可能会差几十毫秒，这个问题无法避免，比如接单过程中总不能sleep吧
21. 由于特殊出租车的存在，detail.txt里的信息已经没有意义，所以你需要通过TestCode里的迭代器打印出特殊出租车的接单信息

while (ite.hasNext()) System.out.println(ite.next());

1. 是否满足LSP规则

由于子类完全继承父类的所有方法，并且通过run()函数使用了父类的所有方法，并没有作出任何改变，只是在原有的基础上构造了迭代器使之能够通过迭代器访问父类中的信息，所以在任何父类型对象出现的地方使用子类对象都不会破坏user程序的行为，故满足LSP规则