

○○ 第七次作业指导书

一、 作业背景和目标

本次作业模拟出租车的乘客呼叫与应答系统，开发相应的程序，继续训练线程安全设计方法，同时应用课堂所讲授的面向对象分析方法和设计原则来开展分析和设计。

该作业为系列作业，从初始就构造一个好的设计尤为重要。

二、 核心概念定义

1. 城市地图

- 1) 使用**网格**来模拟城市地图。所有的道路为水平方向或垂直方向的网格线，如果两个点之间有道路，则这两个点之间存在一条连接。
- 2) 城市地图通过ASCII编码的**文本文件输入**。文件内容为80行字符串，每行有80个数字字符（字符之间允许出现空格或制表符），每个数字字符为0到3之间的整数，表示一个 $A[80 \times 80]$ 的邻接矩阵。输入文件中除数字0, 1, 2, 3, 空格, 制表符和回车换行外，出现任何其他字符都可判定为**无效输入**。如果输入少于或多于80行，或者存在某行输入少于或多于80个0~3之间的数字，则可判定为**无效输入**。
- 3) 坐标：地图网格中的坐标点为水平线与垂直线的交叉点，坐标值为 (i, j) ，其中 i, j 为整数，并有 $0 \leq i < 80$ ， $0 \leq j < 80$ 。

第 i 行的第 j 个数字 $A_{i,j}$ 记录的是地图中 (i, j) 坐标位置的点到与右方坐标 $(i, j+1)$ 的点和下方坐标 $(i+1, j)$ 的点的连接情况，有：

- a) $A_{i,j} = 0$ 表示 (i, j) 与 $(i, j+1)$ 和 $(i+1, j)$ 均无连接。
- b) $A_{i,j} = 1$ 表示 (i, j) 与 $(i, j+1)$ 有连接，但与 $(i+1, j)$ 无连接。
- c) $A_{i,j} = 2$ 表示 (i, j) 与 $(i, j+1)$ 无连接，但与 $(i+1, j)$ 有连接。
- d) $A_{i,j} = 3$ 表示 (i, j) 与 $(i, j+1)$ 和 $(i+1, j)$ 均有连接。

注： (i, j) 与 $(i-1, j)$ 和 $(i, j-1)$ 的连接情况由 $A_{i-1,j}$ 和 $A_{i,j-1}$ 定义（其中 $i-1 \geq 0$ 和 $j-1 \geq 0$ ）。所有的连接均为双向。只有水平和垂直连接，没有对角线连接。

- 4) 文件需要确保地图上的**所有的点都连通**，即整个图是连通图。在图的外缘线上的所有点，不能与图外的点有连接，例如 $A_{79,0}$ 不能是 2 或 3， $A_{79,79}$ 只能是 0。由测试者保证此规则。

2. 出租车

- 1) 出租车限定为100个，出租车起始位置由设计者通过随机数生成器来决定。
- 2) 出租车行驶一条格子边（即由两个直接相邻的坐标点定义的边）的时间为200ms。
- 3) 每辆出租车的状态有四种：
 - a) 服务状态：搭载乘客从打车始发地前往目的地的过程状态。
 - b) 接单状态：车辆抢单后被派遣任务，车辆去往乘客请求出发地的过程状态。
 - c) 等待服务：无服务无接单的空车运行状态，此时可以抢单。
 - d) 停止状态：车辆处于停止服务状态，此时不提供任何服务。
- 4) 出租车的状态有以下转换规则：
 - a) 出租车处于等待服务状态持续20s后，进入停止状态，1s后再次运行立即进入等待服务状态。
 - b) 出租车在被选中响应乘客叫车请求后（即被派单），进入接单状态，需以最短路径（本次作业为最短距离，若有多条最短路径可以任选一条）前往乘客打车出发地。一旦到达用户等待位置，等待1s后（模拟乘客上车场景）立即进入服务状态。
 - c) 出租车完成当前服务（到达用户的目的地）后，进入停止状态，1s后再次运行立即进入等待服务状态。
- 5) 出租车在停止状态、接单状态和服务状态下**不能响应乘客请求**。
- 6) 出租车的行走方式有以下两种：
 - a) 在等待服务状态时，出租车如果遇到道路分支，可随机选择一条分支边行走。
 - b) 要求出租车的随机分支选择行为满足**独立同分布**特性，即每个出租车使用独立的、满足一定分布的随机数生成器来进行分支选择。
 - c) 在接单状态，需以最短路径前往乘客打车出发地。
 - d) 在运载乘客到目的地过程中，出租车必须按照**最短路径**行走。
- 7) 出租车有信用积累，初始所有车信用为 0，每参与抢单一次该出租车信用度立即加 1，每成功服务顾客一次（即把乘客从出发点成功送到目的地后）该出租

车信用度立即加 3。

3. 乘客叫车请求及响应

- 1) 乘客在任意一点 C 向系统发出叫车请求后，系统只把请求发送给在以 C 为中心的 4×4 网格区域里行驶的出租车。叫车请求包括点 C 的坐标和目的地坐标信息，如果目的地坐标无效，则系统拒绝响应该请求。如果请求中点 C 的坐标与目的地一样，则视为无效请求，不予响应。

注：这个 4×4 网格区域实际可通过点 C 上下左右均延伸 2 个最小格子来得到，含 4×4 网格区域的边界点，但不超出地图的外围线。

- 2) 若在同一时刻、同一地点发出去往同一目的地的请求有多个，则视为一个请求。
- 3) 系统一旦收到乘客的叫车请求，会设置一个时间长度为 3s 的抢单时间窗口，在窗口内系统向符合条件（满足本小节第 1 条规则且处于等待服务状态的车辆）的出租车广播发送乘客请求。在抢单时间窗口关闭时，系统在抢单的出租车中，按照本小节第 5 条规则来选择相应出租车响应乘客请求。

注：处于等待服务状态的出租车只要在乘客叫车请求抢单时间窗口内进入到以请求发出地为中心的 4×4 网格区域内就会收到该请求，收到请求后要求立刻抢单。

- 4) 如果在 3s 时间窗内无出租车抢单，系统则视为无车响应，告知乘客无可用出租车，并结束对该乘客请求的处理。
- 5) 如果有出租车响应，系统在抢单的出租车中选择信用度最高的那辆车来派单。如果抢单的出租车中有多辆信用度相同，则选择**当前**距离用户请求出发地最近的车租车；如果仍有多辆满足条件的出租车，则从中随机选择一辆。
- 6) 出租车一旦抢单，系统就会记录，即使出租车在抢单时间窗口关闭时已经离开了以请求发出地为中心的 4×4 区域也视为有效。
- 7) 一辆出租车在同一时刻可以抢不止一单，但每辆车一次服务只能够响应一个乘客请求。
- 8) 出租车在抢单后被派单之间这个时间段，出租车仍然是普通的等待服务状态。一旦系统选择一个出租车来派单，则通知所有其他抢单出租车此单

已派，无需等待派单。

- 9) 任何参与抢单的出租车一旦被选中派单，立刻进入接单状态，并按照相应规则去接乘客。

三、 设计要求

1. 对设计者的要求

- 1) 要求采用本讲介绍的方法进行分析和设计，并按照所提供的模板整理需求分析文档。
- 2) 要求使用多线程和线程安全设计。提供线程安全的乘客叫车请求队列，供测试使用。注意请求队列容量不得小于300个。
- 3) 程序可通过控制台来获得乘客请求，乘客请求格式为[CR, src, dst]，其中CR为标识符，src和dst均为(i, j)形式的坐标位置,表示乘客请求的发出地和目的地。任何超出地图范围的坐标都视为无效请求而被直接忽略，不影响对其他有效请求的处理。乘客请求的产生时间自动从系统获得。系统的基本时间单位 100ms。
- 4) 要求程序把对乘客请求的处理过程输出到文件中，作为测试判断的依据。针对每个乘客请求，需要记录的数据包括：
 - a) 乘客请求内容：发出时刻、请求坐标、目的地坐标；
 - b) 在抢单时间窗内所有参与抢单的出租车信息：车辆编号、车辆位置、车辆状态、车辆信用信息；
 - c) 被派单的车辆运行信息：车辆编号、派单时的车辆位置坐标、派单时刻、达到乘客位置的时刻、乘客位置坐标、到达目的地坐标、到达目的地时刻、途经分支点的坐标和经过时刻。
- 5) 要求提供**测试接口来按照出租车查询状态信息**，状态信息包括**查询时刻、出租车当前坐标、当前所处状态**。
- 6) 要求提供**测试接口来按照状态查询出租车**，返回所有在查询时刻处于相应状态的**所有出租车编号**。
- 7) 本次作业会提供一个GUI程序包，该程序包提供相应接口来展示地图和出租车的位置和运动，通过调用GUI程序包可在测试时直接在GUI界面上进行观察，便于进行测试判断（GUI的使用说明另行提供）。

8) 要求提供能够让测试者完成测试的测试接口，并提供相应的功能说明。

2. 对测试者的要求

1) 对被测试者提交的需求分析文档给出综合评价：

a) 很好(+3)：使用了所介绍的方法，分析完整和细致。

b) 好(+1)：基本使用了所介绍的方法，分析欠完整或较为粗糙。

c) 一般(0)：其他。

注：被评为很好的需求分析文档，课程组将考虑选择性的发布在班级博客，供大家借鉴。

2) 按照 SOLID 设计原则及讲义中所介绍的其它原则（见指导书附录二），对被测试者提交的程序进行检查，如果发现设计原则被违背，则记为一个 incomplete 类型的bug，但同一个原则不重复扣分。

3) 检查设计者提供的测试接口是否满足了关于测试接口的要求（即上述对设计者要求中的第5和第6项），如果不满足，则记为一个incomplete类型的bug。

4) 鼓励编写测试代码来复现基于界面或控制台测试所发现的问题，并在bug报告中提交所编写的测试代码。一旦确认为有效的bug，且测试代码可以复现相应的bug，则测试者获得相应bug类别的1.5倍得分，但不会对被测试者有额外的扣分。

注：建议向请求队列发送请求来模拟乘客呼叫出租车，并通过访问相关出租车对象的状态自动判断程序处理是否正确。

5) 提交bug报告时，需要提供必要的证据，包括但不限于：测试使用的**地图文件**、**界面截图**、乘客请求-响应过程的日志记录文件（参见设计者要求中的第4条）、**测试代码**。

3. 建议和说明

1) 由于100个车辆在6400网格内持续运动，人工测试和检查的难度很大，建议并鼓励通过写程序的方式进行测试。

2) 作为具有说服力的测试参考，应将各种状态结果输出到文件中，作为运行正确性评判的最重要依据。

四、关于GUI程序包的使用说明

本次作业会提供一个GUI包，为大家调试多线程提供和检查结果方便。GUI表现出来的问题不作为程序设计的问题进行评定。

- 1) GUI提供了一个TaxiGUI类，有关地图，出租车以及请求可以通过可视化方式进行展示
- 2) 在使用之前需要对可视化对象进行地图初始化，使用 `boolean LoadMap(int[][] map, int size)` 将地图载入可视化对象，要求输入为一个二维数组，大小为size（本次作业应该是80），地图坐标左上角为坐标原点(0,0)，该地图是否有效由调用者负责。
- 3) 在使用之前需要将出租车进行初始化，每一个出租车需要通过调用 `void SetTaxiStatus(int index, Point point, int status)` 来改变出租车的状态，index取值范围[0,99]，Point为出租车坐标值，status取值为0,1,2，其中0-停止运行；1-服务（从接单开始，到达乘客目的地终止）；2-等待服务。
- 4) 在出租车运行过程中，每一次出租车移动一步后仍然需要调用 `void SetTaxiStatus(int index, Point point, int status)` 来改变出租车的状态。
- 5) 系统收到出租车请求，也需要将该请求输入可视化对象，通过调用 `void RequestTaxi(Point src, Point dst)` 来输入，src, dst分别为乘客提出出租车请求的起始坐标和目的坐标。
- 6) GUI不是测试内容，任何GUI层次的问题都不作为测试内容且无需报告bug。

附件一：需求分析模板

1、系统简述

以用户视角理解该系统，包括：

1) 系统功能描述

需梳理出若干功能点（即能干什么）

2) 系统性能要求

系统运行过程中的要求，往往通过数字来表达的能力

3) 约束条件

包括输入、输出、数据、测试等规定

2、交互分析

按照ppt中给出的方法，并参考所举示例，完成以下内容：

- 1) 与系统有交互关系的对象识别，根据指导书识别此类对象，并进一步给出相应

的属性，要有必要的文字说明。注：这一步应把系统视为黑盒进行分析。

- 2) 识别待开发系统与上述第1步所识别对象之间的交互，给出交互动作，并从数据和时间两个方面分析交互特征。
- 3) 对象识别与构造，根据指导书内容，进入系统识别相关类（重点考虑数据管理），识别其中的属性和方法，以及类之间的关系。注：不必引入设计细节，要求使用类图来展示识别结果（可另外提交jpg，或使用word和pdf文档说明时注意去掉个人信息）。
- 4) 并发分析，根据指导书识别系统中的并发行为和并发特征，并给出多线程设计的基本考虑。

附件二：需要进行检查的设计原则列表

- 1) Single Responsibility Principle
- 2) Open Close Principle
- 3) Liskov Substitution Principle
- 4) Interface Segregation Principle
- 5) Dependency Inversion Principle
- 6) 层次化抽象原则，按照问题域逻辑关系来识别类；
- 7) 责任均衡分配原则，避免出现God类和Idiot类；
- 8) 局部化原则，类之间不要冗余存储相同的数据，方法之间不能够出现控制耦合；
- 9) 重用原则（共性抽取原则），把不同类之间具有的共性数据或处理抽象成继承关系，避免冗余；
- 10) 显式表达原则，显式表达所有想要表达的数据或逻辑，不使用数组存储位置或者常量来隐含表示某个特定状态或数据；
- 11) 信任原则，一个方法被调用时，调用者需要检查和确保方法的基本要求能够被满足，获得调用结果后需要按照约定的多种情况分别进行处理；
- 12) 懂我原则，所有类、对象、变量、方法等的命名做到“顾名思义”