

〇〇 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，同时检查课堂所讲授的面向对象分析方法和设计原则。

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**网格区域**来模拟城市地图。所有的道路要么是**水平方向**，要么是**垂直方向**，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退格符），每个数字字符为 0 到 3 之间的整数，表示一个 80×80 的邻接矩阵 $A_{80 \times 80}$ 。输入文件中除数字 0, 1, 2, 3, 空格，退格和回车换行外，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。

输入第 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，若

- 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。

提示： (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。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。

- 3) 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，且不能存在点与图外的点有连接，例如 $A_{80,1}$ 不能是 2 或 3， $A_{80,80}$ 只能是 0。

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**随机分布**。
- 2) 出租车行驶一条格子边的时间为200ms。

- 3) 每辆出租车的状态有三种：停止运行、服务（在运行且车内有乘客）、等待服务（在运行但车内无乘客）。
- 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，停止运行1s，然后再次运行立即进入等待服务状态。
 - b) 出租车在抢到用户请求后，一旦到达用户等待位置，停止运行1s，然后再次运行立即进入服务状态。
 - c) 出租车完成当前服务（到达用户的目的地）后，停止运行 1s，然后再次运行立即进入等待服务状态。
 - d) 任何情况下，只要是停止运行，都处于停止运行状态。
- 5) 在停止运行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) 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，每抢单一次会使其信用度加 1，每成功服务顾客一次会使其信用度加 3。

3. 乘客请求及响应

- 1) 乘客在任意一点 C 向系统发出呼叫请求后，系统只把呼叫请求发送给在以 C 为中心的 4×4 区域（上下左右均延伸 2 个格子，但不超出 80×80 的大范围）里行驶的出租车。呼叫请求包括点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。
- 2) 系统一旦收到乘客请求，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，按照第5)条规则来选择相应出租车响应乘客请求。
- 3) 如果 3s 内无出租车对系统应答则视为无车响应，系统告知乘客无可用出租车，系统对该乘客请求的处理结束。
- 4) 处于等待服务状态的出租车只要在乘客请求的抢单时间窗口内，一旦进入以请求发出地为中心的 4×4 网格区域就会收到请求。出租车只要收到请求就

会抢单。

- 5) 如果有车响应，系统为乘客从当前时间窗口中抢单的出租车中自动进行选择，**在抢单时间窗口关闭时刻**选择**处于等待服务中**信用度最高的出租车；如果有多辆信用度相同的出租车，则选择当前距离用户请求出发地最近的；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) 同一时刻在同一地点发出的去同一目的地的请求视为相同请求，相同请求只处理一个和一次。

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，并提供需求分析文档。
- 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：请求发出时，处于以请求src为中心的 4×4 区域中的所有出租车状态、信用信息；在抢单时间窗内所有抢单的出租车；系统选择响应相应请求的出租车；出租车响应相应请求过程中的实际行驶路径。
- 5) 要求提供**测试接口来查询出指定出租车的状态**（这里的测试接口不是 JAVA 里面的接口），至少能够在**任意时刻**给出指定**出租车的状态信息**，信息包括但不限于**当前的时刻**，**出租车当前坐标**，以及在任意时刻返回处于指定状态

的出租车对象（如服务状态）。

- 6) 本次作业提供一个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界面上进行判断，便于进行测试判断。但是在发现bug时要求提供输出的文件内容作为证据。
（GUI的使用说明另行提供）
- 7) 要求提供测试者能够完成测试要求的相关接口。
- 8) 要求写清楚为测试者提供的测试接口功能说明。

2. 对测试者的要求

- 1) 对需求分析文档的详细程度给出综合评价：
 - a) 很好(+3)：使用了所介绍的方法，分析完整。
 - b) 好(+1)：基本使用了所介绍的方法，分析欠完整。
 - c) 一般(0)：其他。
- 2) 按照 SOLID 设计原则对实现进行检查，如果发现设计原则被违背，则记为一个 incomplete，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实验的功能要求和测试要求，如果有遗漏，则记为功能性缺失，但同一个原则不重复扣分。
- 4) 编写测试线程，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) 提交评价时，需要提供测试使用的地图文件和测试部分的代码。
- 6) 正式报告bug时需提交通过文件记录的乘客请求-响应过程，至少包括这样一组信息：
 - a) 乘客请求内容：时刻、请求位置、目的地位置
 - b) 4*4请求区域内的车辆信息：车辆编号、车辆位置、车辆状态、车辆信用
 - c) 响应请求的车辆运行信息：车辆ID、分派任务时的车辆位置坐标、时刻、达到乘客位置的时刻、位置坐标、到达目的的坐标、时刻、关键途经点（运行方向改变的位置点）的坐标和时刻。

3. 建议和说明

- 1) 由于100个车辆在6400网格内不停的变化，通过人工测试和检查是一个非常艰苦和耗时的工作，因此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2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（在运行且车内有乘客）；2-等待服务（在运行但车内无乘客）
- 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项且无需报告bug。