2.7.2 队列：银行叫号系统的实现

1. 需求描述

银行取号机（叫号机）到处可见，也就是在生活中经常看到的排队机，它给用户带来了很大的方便。

使用取号排队系统，一方面可消除用户长时间“站队”的辛苦、对“站错队”、“插队”的抱怨，避免发生排错队和混乱嘈杂的现象，减少许多不必要的纠纷，全面改善服务质量和企业形象，另一方面更可以依据统计数据调整业务分配、挖掘潜力、合理安排窗口服务，减少群众的等候时间，提高办事效率。同时，排队系统支持多种形式的排队，可依照业务的种类或用户种类进行排队，支持对特殊对象（会员）的优先服务。

编写一个程序模拟银行取号系统，该程序只是模拟了取号排队的功能，至于完全的模拟需要硬件与软件的结合。本模拟系统的最终目标是：实现叫号机的基本功能，即用户到达后可以叫号，工作人员登录进入系统后可以对用户进行办理业务，有如下规则：

1）银行客户有多个优先级普通业务用户，VIP 业务用户，对公业务用户。

2）银行有多个窗口，分为普通窗口，VIP 窗口和对公窗口，窗口数任意，三种业务窗口的比例为 3∶1∶1。

3）各类型客户在其对应窗口按顺序依次办理业务。

4）VIP 窗口和对公业务窗口，在空闲的时候可以处理普通业务窗口，而一旦有对应的客户等待办理业务的时候，则优先处理对应客户的业务。

5）随机生成各种类型的客户到达银行取票的过程，各类型业务用户数量可以预先设定。

6）自行设定业务办理时间，以及业务操作，可设置为三种用户设置相同的服务时间。

7）办理业务的过程通过一个时间变量来模拟。无需图形界面，只需要控制台以命令行形式模拟，窗口叫号过程，输出用户的类型还有该类型的编号，如果窗口由用户在接受服务，则输出窗口类型和编号，并且输出用户类型及编号。

2. 问题分析

本题目是队列的应用，主要需要模拟的是，用户入队列，办理业务用户出队列，优先级比较选择队列。

模拟用户入队列，也就是模拟不同类型的用户取号的过程，为简化代码做出如下分析。用户类型分为三种类型，一种普通用户数量较多，两种特殊用户类型较少，故可以将这三种用户设置成为三个队列。根据用户比例，可以简化程序构造，预先设置这三类用户分别由 n1 位，n2 位，n3 位，预先存储于数组之中。取号时间即入队时间亦可预先设定，但是需要出现条件中预设的特殊情况，例如特殊队列空闲，普通用户队列有人的情况。

另外两种情况的模拟，也就是模拟个窗口在处理用户业务之后的叫号过程。同样做一下简化，银行窗口的处理用户业务的时间设定为固定的时间 t 秒，窗口处理业务完成之后，立刻从对应的队列中取得下个客户，并且在窗口中显示窗口号以及其叫的用户号。

这里假设普通用户窗口一直有人办理业务，不会出现空闲情况。VIP 和对公窗口客户较少，如果出现空闲情况，可以从普通用户队列用户中取得下一个用户，如果处理业务的过程中，一旦有对应的客户等待办理业务的时候，则处理完之后优先处理对应客户的业务。最后根据三种窗口的比例，不妨假设三种类型的窗口个数分别为普通用户窗口 3 个，VIP 和对公窗口各一个。

综上所述，需要三个用户队列来模拟存储不同类型的用户，需要三种对象来模拟存储不同类型的银行窗口，还需要编写一个入队模拟器来模拟用户的到达情况以及窗口叫

号的过程，最重要的还需要预设好用户到达的场景以达到简化程序，覆盖所有题目要求的条件。

3. 概要设计

首先根据上一部分的分析，先将程序的逻辑结构分析如下：整个模拟系统可以分为三大类：用户类，银行窗口类和模拟程序类。其中用户类为基类下面可以分为普通用户，VIP 用户，还有对公用户。

银行窗口类按照三种用户类型可以抽取公有成员作为基类，然后针对不同的用户类型分为普通窗口，VIP 窗口，还有对公窗口。模拟程序类，负责三种情况的模拟，首先模拟程序主结构也就是时间的变化，模拟用户来到银行入队列的过程，和银行窗口服务和叫号的过程，整个程序的类图如图 2-41 所示。

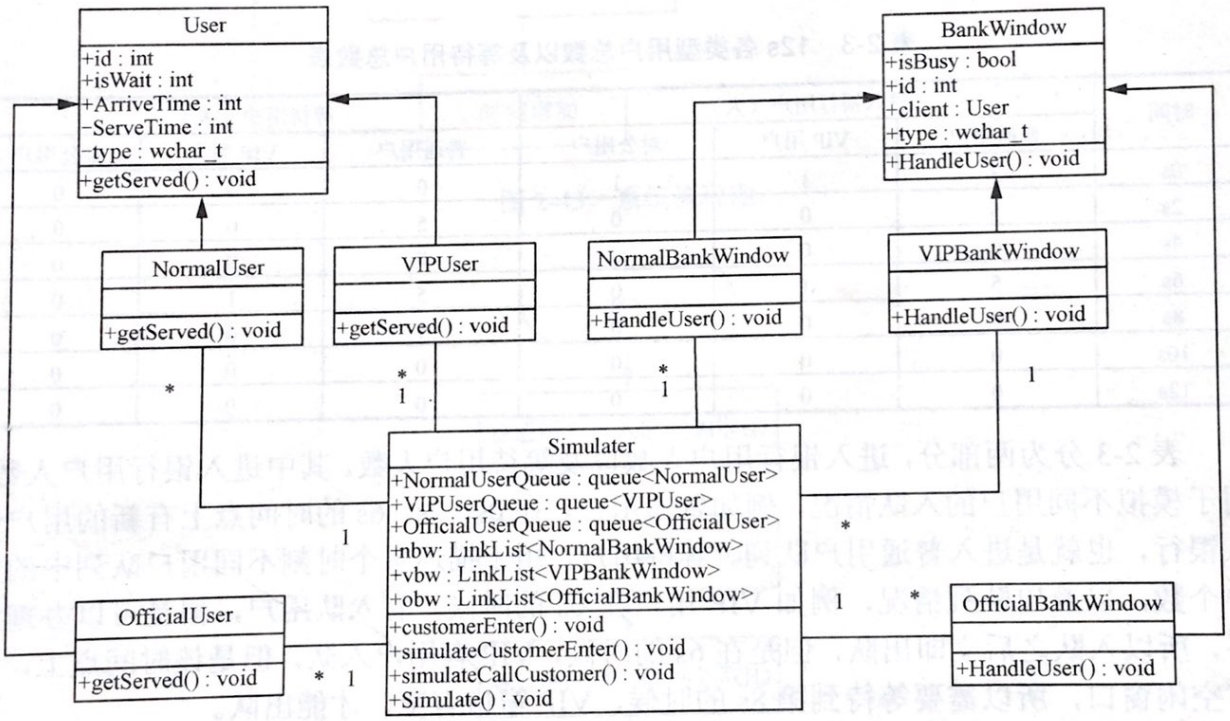

图 2-41 银行叫号系统类图

程序主体通过调用 Simulater 模拟器类的 Simulate 方法，来模拟整个叫号过程。该类是整个程序的核心类，数据成员包括三个用户队列来标识三类用户，还有三个银行窗口链表用来模拟存储银行窗口，方法包括模拟用户入队，银行处理用户队列，以及呼叫用户，程序结构如图 2-42 所示。

然后设置用户到达的场景。业务的处理时间是 t 秒，那么可以假设用户入队的时间间隔是 t1 秒，显然需要 t1<t，才会出现等待用户，故不妨设 t=4s，t1=2s。然后假设初始用户就可以将窗口占满，也就是初始情况下有 3 位普通用户，VIP 和对公用户各一位，然后模拟一个 12s 的叫号过程，会在 0s，4s 和 8s 的时候会有叫号的过程，12s 的时候没

有用户，没有叫号，程序结束，如表 2-3 所示。

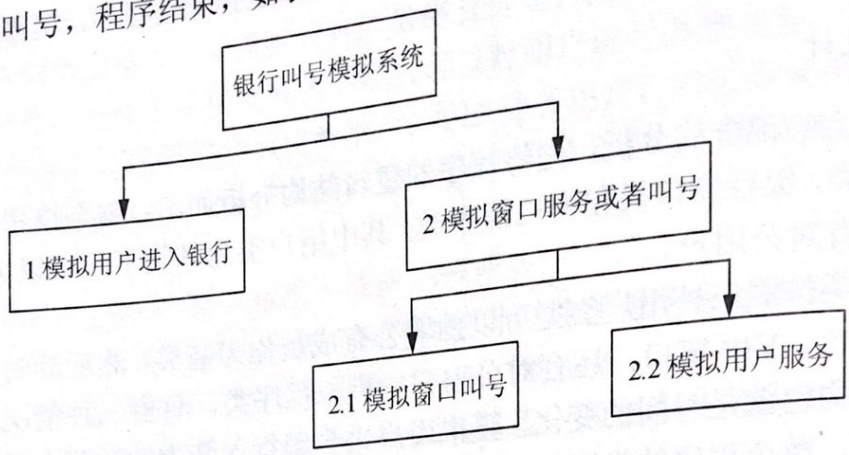

图 2-42　程序结构图

表 2-3　12s 各类型用户总数以及等待用户总数表

时间	进入银行用户（人）			等待用户（人）		
	普通用户	VIP 用户	对公用户	普通用户	VIP 用户	对公用户
0s	3	1	1	0	0	0
2s	5	0	0	5	0	0
4s	0	0	0	0	0	0
6s	5	1	0	5	1	0
8s	0	0	0	1	0	0
10s	0	0	0	0	0	0
12s	0	0	0	0	0	0

表 2-3 分为两部分，进入银行用户人数以及等待用户人数。其中进入银行用户人数，用于模拟不同用户的入队情况，例如普通用户，在 0s，2s，6s 的时间点上有新的用户进入银行，也就是进入普通用户队列。等待用户，用于描述各个时刻不同用户队列中的用户个数，以及出队列情况，例如 VIP 用户在 0s 的时候，有入队用户，但是可以办理业务，所以入队之后立即出队，但是在 6s 的时候，VIP 有用户入队，但是该时间点上，没有空闲窗口，所以需要等待到第 8s 的时候，VIP 窗口有空，才能出队。

4. 详细设计

首先需要预先设置一下系统参数，包括时间，窗口类型，用户类型，以及用于表示用户到达场景的矩阵。

有了这些预定义参数之后，就可以直接定义 Simulater 中的主要模拟程序，模拟整个时间过程中的用户进入银行，接受服务，叫号的过程，流程如图 2-43 所示。

在模拟用户进入银行的过程中，将表格中的数据抽象成是一个二维数组定义为 CustomerEnerMatrix。根据 CustomerEnerMatrix 每一行的三个元素表示在时间 t 的时间点上，三类用户进入银行的数量，将不同用户进入银行的过程抽象为三种用户在同一个时间点上依次进入各自的队列的过程，流程图如图 2-44 所示。

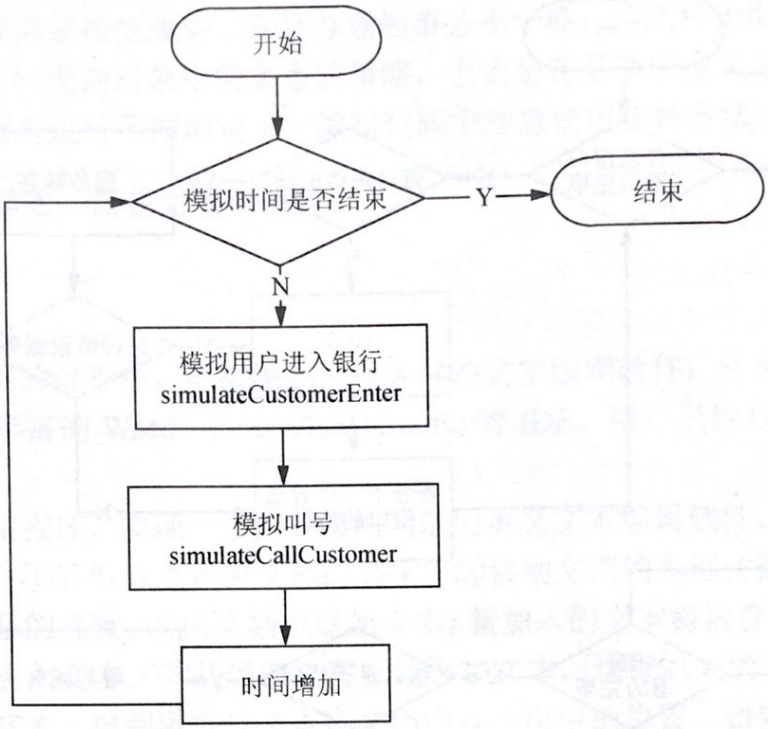

图 2-43 系统流程图

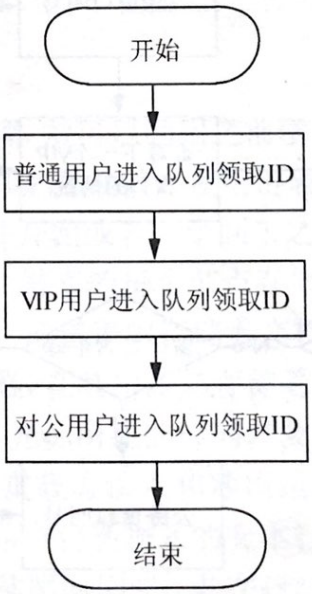

图 2-44 流程图

接下来介绍模拟叫号过程，假设一个用户被叫号之后，立刻开始接受窗口服务，处理事务，并且窗口一旦处理完用户立刻呼叫下一位用户，中间没有时间间隔。和模拟进入的结构类似，分别遍历银行里面的三类窗口，调用函数来处理用户队列，如果窗口上有正在接受服务的顾客，则显示窗口信息以及接受服务的客户信息，如果窗口上顾客结束服务后立即叫号。其流程如图 2-45 所示。

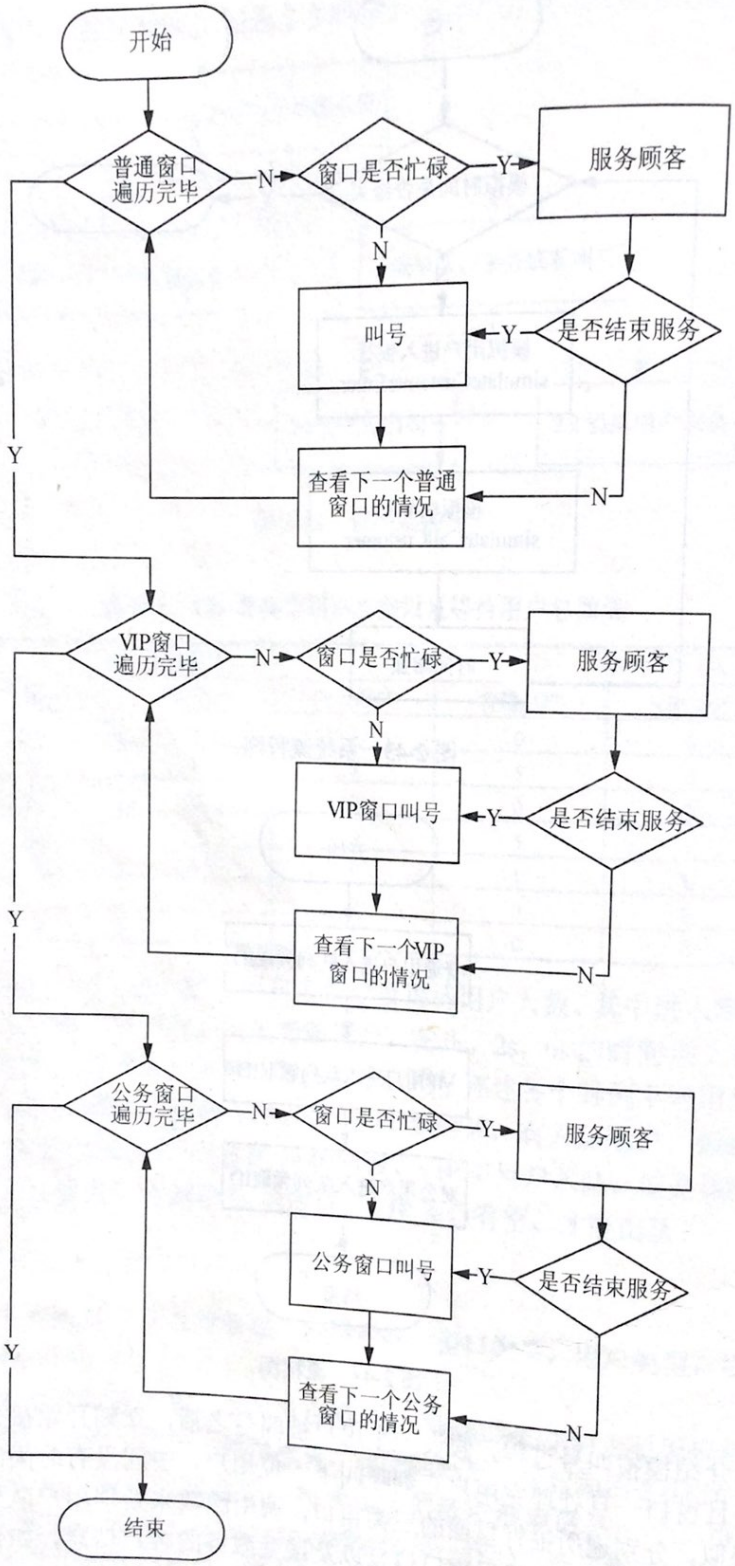
开始
普通窗口遍历完毕
N
窗口是否忙碌
Y
服务顾客
N
叫号
Y
是否结束服务
Y
查看下一个普通窗口的情况
N
Y
VIP窗口遍历完毕
N
窗口是否忙碌
Y
服务顾客
N
VIP窗口叫号
Y
是否结束服务
Y
查看下一个VIP窗口的情况
N
Y
公务窗口遍历完毕
N
窗口是否忙碌
Y
服务顾客
N
公务窗口叫号
Y
是否结束服务
Y
查看下一个公务窗口的情况
N
结束

图 2-45　流程图

不同的用户都需要接受服务，只是办理的事务不一样，因此对于用户接受服务这个事件，需要用到C++面向对象中的多态的策略，也就是在基类中定义通用方法，在各子类中，根据自身情况进行不同地定义，编写代码中注意使用这种方法。