# 实验7 多线程编程

### 实验目的

* 1. 掌握Java多线程机制；
  2. 掌握自定义线程；
  3. 掌握多线程同步与通信。

### 实验内容

采用多线程，模拟售票系统的售票和退票过程。编程要求如下：可售票的数量为一个固定值，当还有余票的时候可以售出票，如果没有余票了(票数为0)，则必须等待退票后才能售出票。售票线程正在售票的时候不能退票，反之亦然。使用synchronized实现同步控制，并使用wait()和notify()实现线程之间的通信。

1. 定义Tickets类（Tickets.java）， 表示所有的票，Tickets类至少有1个域。
   * 一个域：票的数量(size)
   * 两个方法：退票方法和售票方法

|  |
| --- |
| **//Ticket.java**  **public** **class** Ticket {  **int** rows; //票的总行数  **int** columns; //票的总列数  **int** total; //总票数=行数\*列数  **int** freeSeatNum; //余票数量  **boolean** isSelling; //true表示正在售票;不售票时为false，则表示可以退票    ……    //售票方法：售出num张票  **synchronized** **public** **void** sellTicket(**int** num){… }    //退票方法：退回num张票  **synchronized** **public** **void** returnTicket(**int** num){…… }  …… |

1. 定义一个线程SellTicketThread类，售票线程；
2. 定义一个线程ReturnTicketThread类，退票线程；
3. 定义一个测试线程的主类TicketManagementSystem.java，分别创建售票和退票的线程对象，模拟多次(至少3次)售票和退票的过程。观察线程采用同步控制和不采用同步控制两种情况之下程序的运行情况，并将运行结果输出到屏幕上。
4. 程序输出如下：

|  |
| --- |
| 请输入行数：  4  请输入列数：  4  一共有16张票  售票前余票：16; 想购票数：3  购买了3张票  售票后余票：13  售票前余票：13; 想购票数：3  退票中，SellTicketThread Waiting...  退了2张票  退票后余票：15  售票中，ReturnTicketThread Waiting...  购买了3张票  售票后余票：12  售票前余票：12; 想购票数：3  退票中，SellTicketThread Waiting...  退了2张票  退票后余票：14  售票中，ReturnTicketThread Waiting...  购买了3张票  售票后余票：11  退了2张票  退票后余票：13 |