**广州大学学生实验报告**

**开课实验室：**电子楼416B **2019年9月30日**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 计算机科学与网络工程学院 | | **年级、专业、班** | 软件171 | **姓名** | 谢金宏 | **学号** | 1706300001 |
| **实验课程名称** | | Java语言实验 | | | | | **成绩** |  |
| **实验项目名称** | | 实验3 | | | | | **指导老师** | 王宇 |
| **一、实验目的**  **熟悉Java多线程应用程序开发方法。**  **二、实验任务**  **编写一个Java多线程应用程序，完成3个售票窗口同时出售20张票，如图所示。**    **要求：**   1. **票数使用同一个静态值。** 2. **为保证不会出现卖出同一张票，要求使用Java多线程同步锁。**   **三、实验内容**  public class Cinema {      public static void main(String... args) {          new Cinema(20, 3);  // 创建一个20张票待售，3个窗口售票的Cinema。      }      Station[] stations;         // 售票窗口      int totalTicketCount;       // 全部票的数量      int soldTicketCount = 0;    // 已卖出票的数量      Cinema(int ticketNumber, int stationNumber) {          this.totalTicketCount = ticketNumber;          prepareStations(stationNumber);  // 创建stationNumber个售票窗口。          openStations();  // 各个售票窗口开始售票。          waitUntilAllStationsAreClosed();  // 等待所有窗口结束售票。          System.out.println("票卖完了！");      }      // 创建指定数量的售票窗口。      private void prepareStations(int number) {          stations = new Station[number];          for (int i = 0; i < number; ++i) {              stations[i] = new Station(this, "窗口" + (i+1));          }      }      // 各个窗口开始售票。      private void openStations() {          for (Station station : stations) {              station.start();          }      }      // 阻塞直到所有的售票站关门。      private void waitUntilAllStationsAreClosed() {          for (Station station : stations) {              try {                  station.join();              } catch (InterruptedException e) {                  e.printStackTrace();              }          }      }      // 返回未卖出的一张票的序号；若票已卖完则返回0。      public synchronized int getTicket() {          if (soldTicketCount < totalTicketCount) {              ++soldTicketCount;              return soldTicketCount;          }          return 0;      }  }  class Station extends Thread {      Cinema cinema;      Station(Cinema cinema, String stationName) {          super(stationName);          this.cinema = cinema;      }      public void run() {          while (true) {              // 模拟等待顾客到来的时间。              try {                  sleep((int)(Math.random() \* 2000));              } catch (InterruptedException e) {                  e.printStackTrace();              }              // 尝试向顾客售票。              int ticket = cinema.getTicket();  // 从Cinema取票。              if (ticket != 0) {  // 若成功取票，则卖出这张票。                  System.out.println(getName() + "卖出了第" + ticket + "张票。");              }              else {                  return; // 余票不足时关闭窗口。              }          }      }  }  **四、实验结果记录（程序运行结果截图）** | | | | | | | | |
|  | | | | | | | | |