**《Java程序设计》实验报告**

**学生姓名： 耿立博 班级： 19计算机3班 学号： 2019207320132**

**实验日期： 2021.6.8 指导教师： 胡继礼、谷宗运、李芳芳**

**实验环境：win7+Jdk1.8**

**一、实验目的**

**掌握** 1.多进程和多线程的概念；

2.创建线程的常用方法；

3.线程同步方法synchronized和ReentrantLock可重入锁；

**熟悉** 1. 线程的常用方法：join、sleep()、currentThread()、getName()、

**二、实验内容**

**1.定义一个类TestRunThread类要求如下：**

（1）创建线程t1（使用继承Thread方式），run方法打印“Thread 1 start”和当前线程名称；

（2）创建线程t2（使用实现Runnable接口方式），run方法打印“Thread 2 start”和当前线程名称；

（3）创建线程t3（使用java8 lambda方式），run方法打印“Thread 3 start”和当前线程名称；

（4）启动三个线程；

（5）主线程main打印“main end”；

（6）分别使用join方法和sleep方法确保主线程main打印的信息“main end”在最后输出；

（7）包名：cn.edu.ahtcm.thread

**2.从合肥-北京的动车G270复兴号动车目前还有1000张余票，目前火车站有三个窗口同时售票，请编写多线程TestSaleTicketThread2类程序，实现三个窗口同时售票，要求：**

（1）使用多线程进行同步售票，每个窗口售票后，打印售票窗口信息（窗口1、窗口2、窗口3）和票号信息（假设编号为1，2，3…1000）和窗口信息；

（2）不能出现超卖现象；

（3）不得将一张票同时售卖给两个人；

（4）使用synchronized关键字对一个对象进行加锁，实现同步售票；

（4）包名cn.edu.ahtcm.thread2;

**3.从合肥-北京的动车G270复兴号动车目前还有1000张余票，目前火车站有三个窗口同时售票，请编写多线程TestSaleTicketThread3类程序，实现三个窗口同时售票：要求：**

（1）使用多线程进行同步售票，每个窗口售票后，打印售票窗口信息（窗口1、窗口2、窗口3）和票号信息（假设编号为1，2，3…1000）和窗口信息；

（2）不能出现超卖现象；

（3）不得将一张票同时售卖给两个人；

（4）使用synchronized关键字对方法加锁，实现同步售票；

（4）包名cn.edu.ahtcm.thread3;

**4.从合肥-北京的动车G270复兴号动车目前还有1000张余票，目前火车站有三个窗口同时售票，请编写多线程TestSaleTicketThread4类程序，实现三个窗口同时售票：要求：**

（1）使用多线程进行同步售票，每个窗口售票后，打印售票窗口信息（窗口1、窗口2、窗口3）和票号信息（假设编号为1，2，3…1000）和窗口信息；

（2）不能出现超卖现象；

（3）不得将一张票同时售卖给两个人；

（4）使用ReentrantLock对卖票方法进行加锁和解锁，以实现同步售票；

（4）包名cn.edu.ahtcm.thread4;

**三、实验过程（按上述实验内容要求写出源代码及运行结果）**

**1.实验-1**

**【源码】**

**package** cn.edu.ahtcm.thread;

/\*（1）创建线程t1（使用继承Thread方式），run方法打印“Thread 1 start”和当前线程名称；

（2）创建线程t2（使用实现Runnable接口方式），run方法打印“Thread 2 start”和当前线程名称；

（3）创建线程t3（使用java8 lambda方式），run方法打印“Thread 3 start”和当前线程名称；

（4）启动三个线程；

（5）主线程main打印“main end”；

（6）分别使用join方法和sleep方法确保主线程main打印的信息“main end”在最后输出；

（7）包名：cn.edu.ahtcm.thread

\*/

**public** **class** TestRunThread {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Thread t1 = **new** MyThread();

Thread t2 = **new** Thread(**new** Thread2());

Thread t3 = **new** Thread(()->{

System.***out***.println("Thread 3 start");

System.***out***.println("当前线程名称为线程3");

});

t1.start();

t2.start();

t3.start();

**try** {

t1.join();

t2.join();

t3.join();

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println("main end");

}

}

**class** MyThread **extends** Thread{

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**super**.run();

System.***out***.println("Thread 1 start");

System.***out***.println("当前线程名称为线程1");

}

}

//实现Runnable接口

**class** Thread2 **implements** Runnable{

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

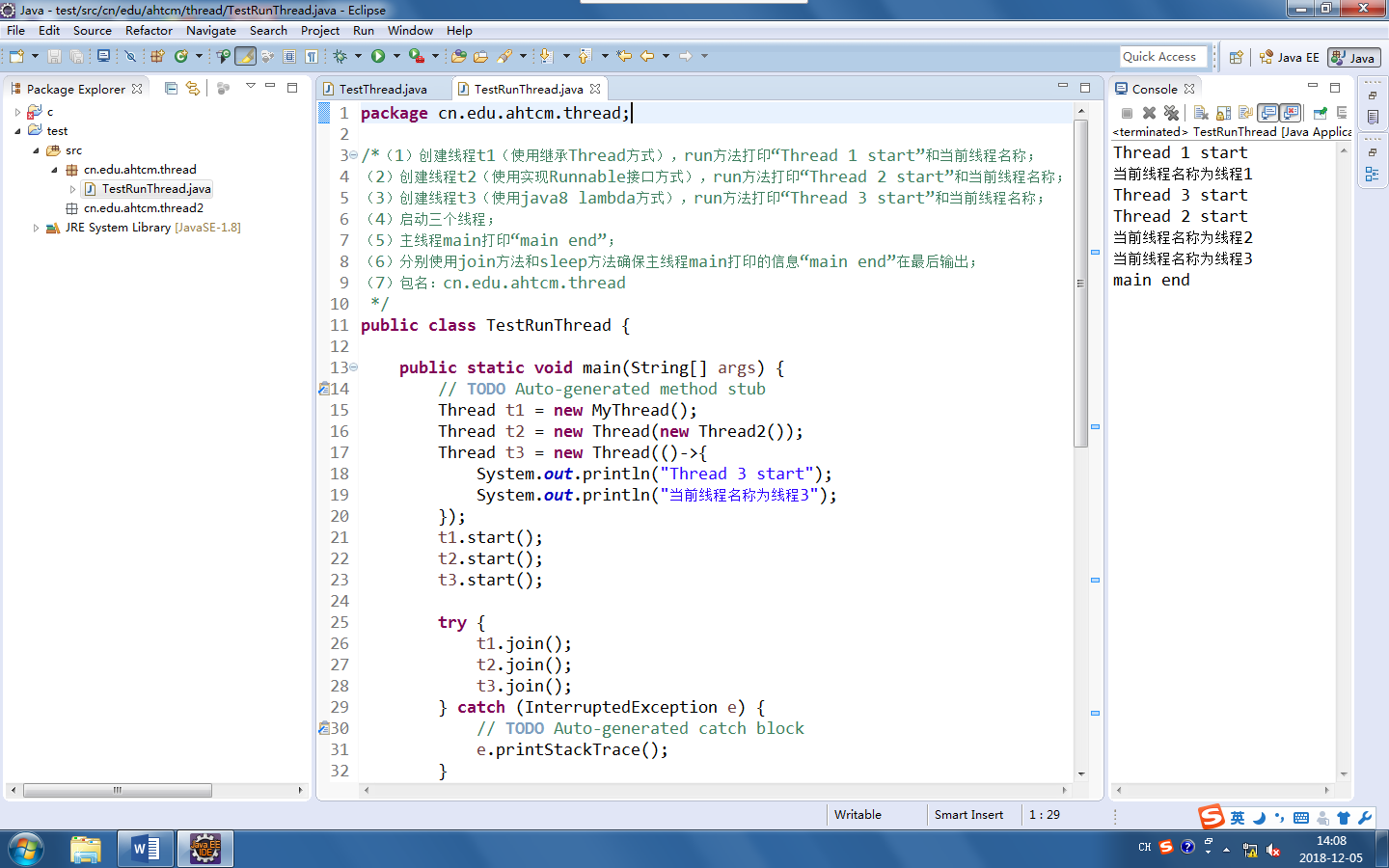
System.***out***.println("Thread 2 start");

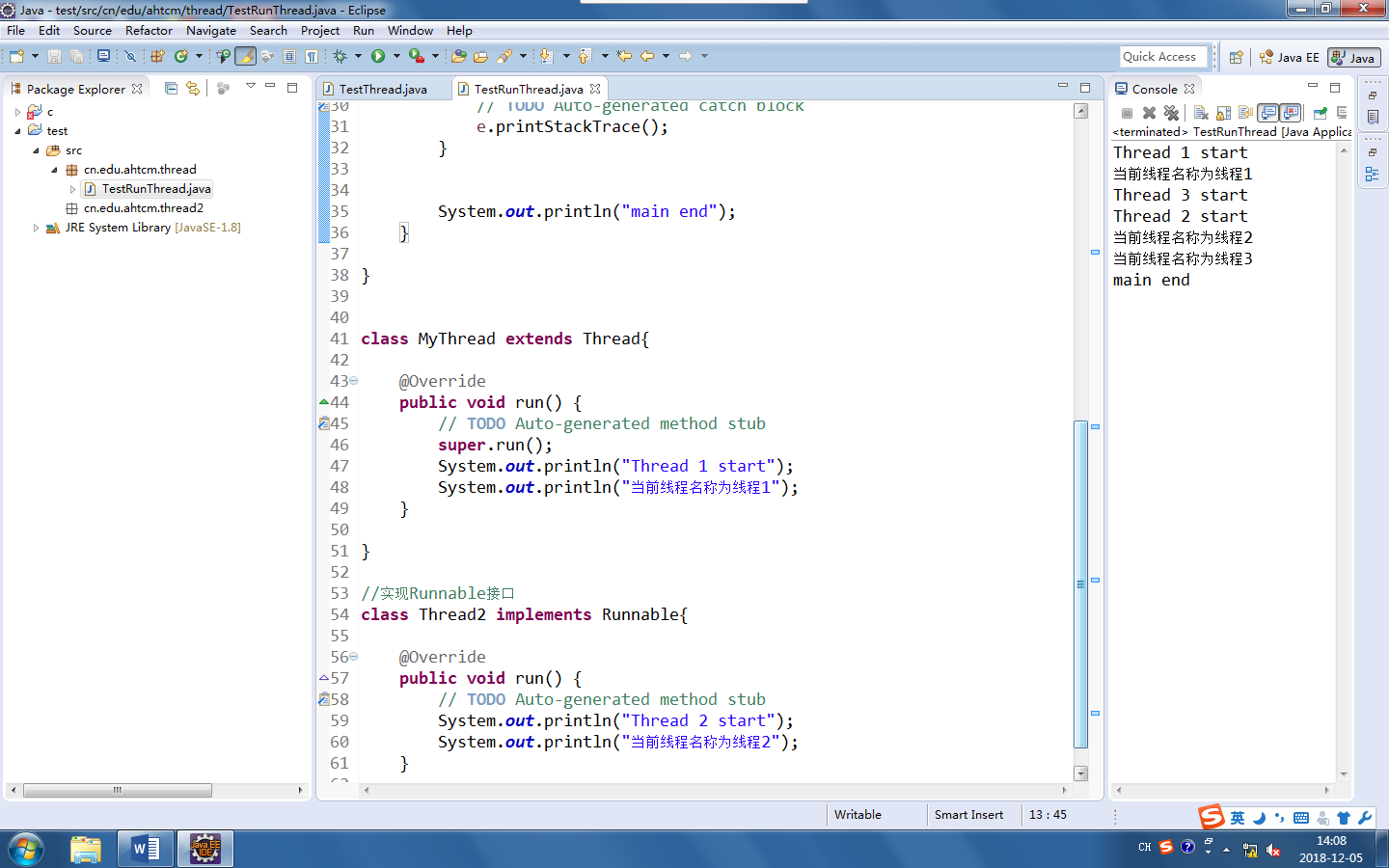
System.***out***.println("当前线程名称为线程2");

}

}

**【运行结果】**





**2.实验-2：**

**【源码】**

**package** cn.edu.ahtcm.thread2;

**public** **class** TestSaleTicketThread2 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Thread t1 = **new** Thread(**new** TicketWindow());

Thread t2 = **new** Thread(**new** TicketWindow());

Thread t3 = **new** Thread(**new** TicketWindow());

t1.setName("窗口1");

t2.setName("窗口2");

t3.setName("窗口3");

t1.start();

t2.start();

t3.start();

}

}

**class** Counter{

**public** **static** **int** *count* = 1000;

//定义锁对象

**public** **final** **static** Object ***lock*** = **new** Counter();

}

**class** TicketWindow **implements** Runnable{

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**while**(**true**){

//

**synchronized**(Counter.***lock***){

**if**(Counter.*count*>0){

System.***out***.println(Thread.*currentThread*().getName()+"票号:"+Counter.*count*);

Counter.*count*--;

}**else**{

**break**;

}

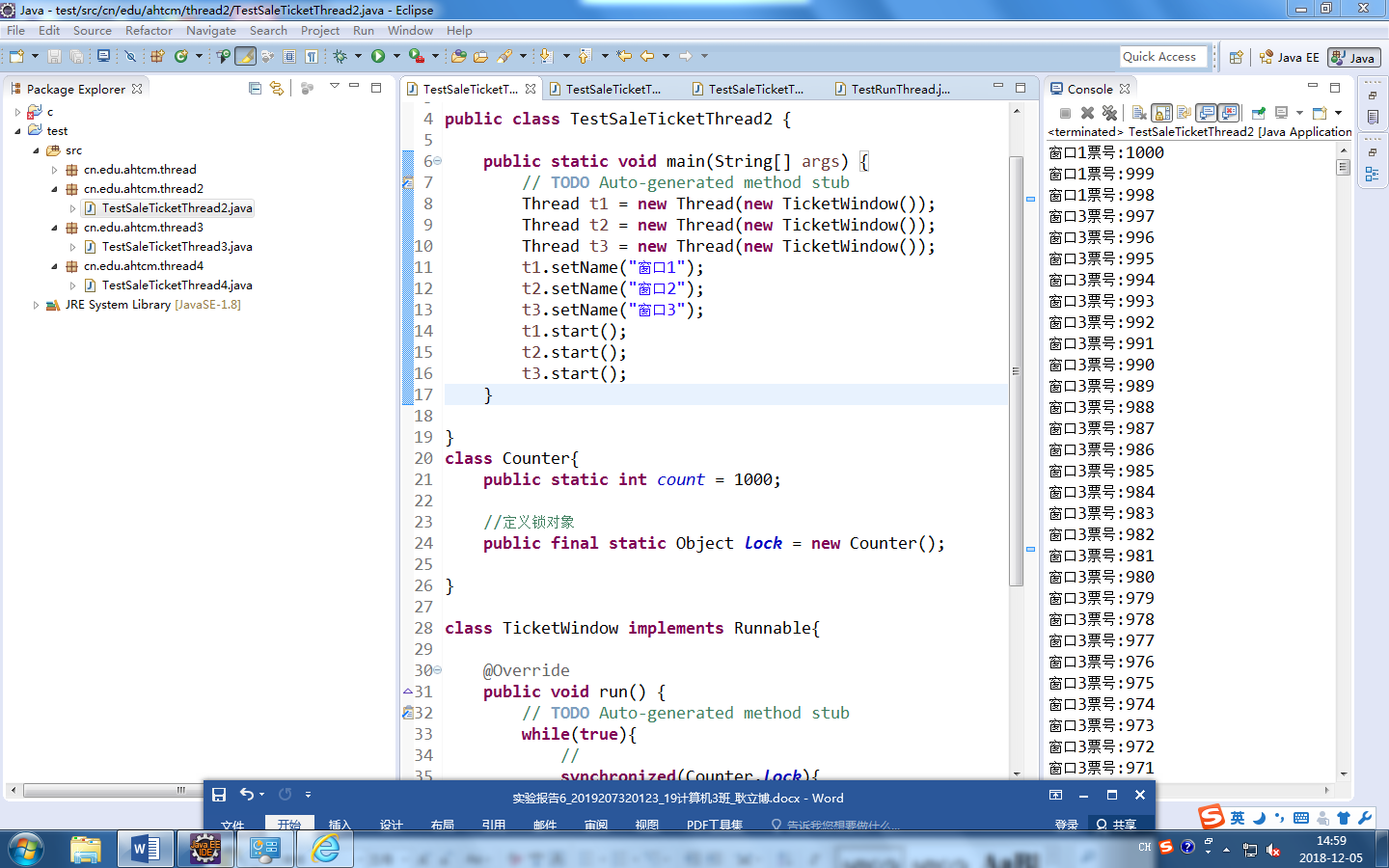
}

}

}

}

**【运行结果】**



**3.实验-3：**

**【源码】**

**package** cn.edu.ahtcm.thread3;

**import** cn.edu.ahtcm.thread3.Counter;

**public** **class** TestSaleTicketThread3 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Thread t1 = **new** Thread(**new** TicketWindow());

Thread t2 = **new** Thread(**new** TicketWindow());

Thread t3 = **new** Thread(**new** TicketWindow());

t1.setName("窗口1");

t2.setName("窗口2");

t3.setName("窗口3");

t1.start();

t2.start();

t3.start();

}

}

**class** Counter{

**public** **static** **int** *count* = 1000;

//锁定方法

**static** **synchronized** **void** sale(){

**while**(**true**){

**if**(Counter.*count*>0){

System.***out***.println(Thread.*currentThread*().getName()+"票号:"+Counter.*count*);

Counter.*count*--;

}**else**{

**break**;

}

}

}

}

**class** TicketWindow **implements** Runnable{

@Override

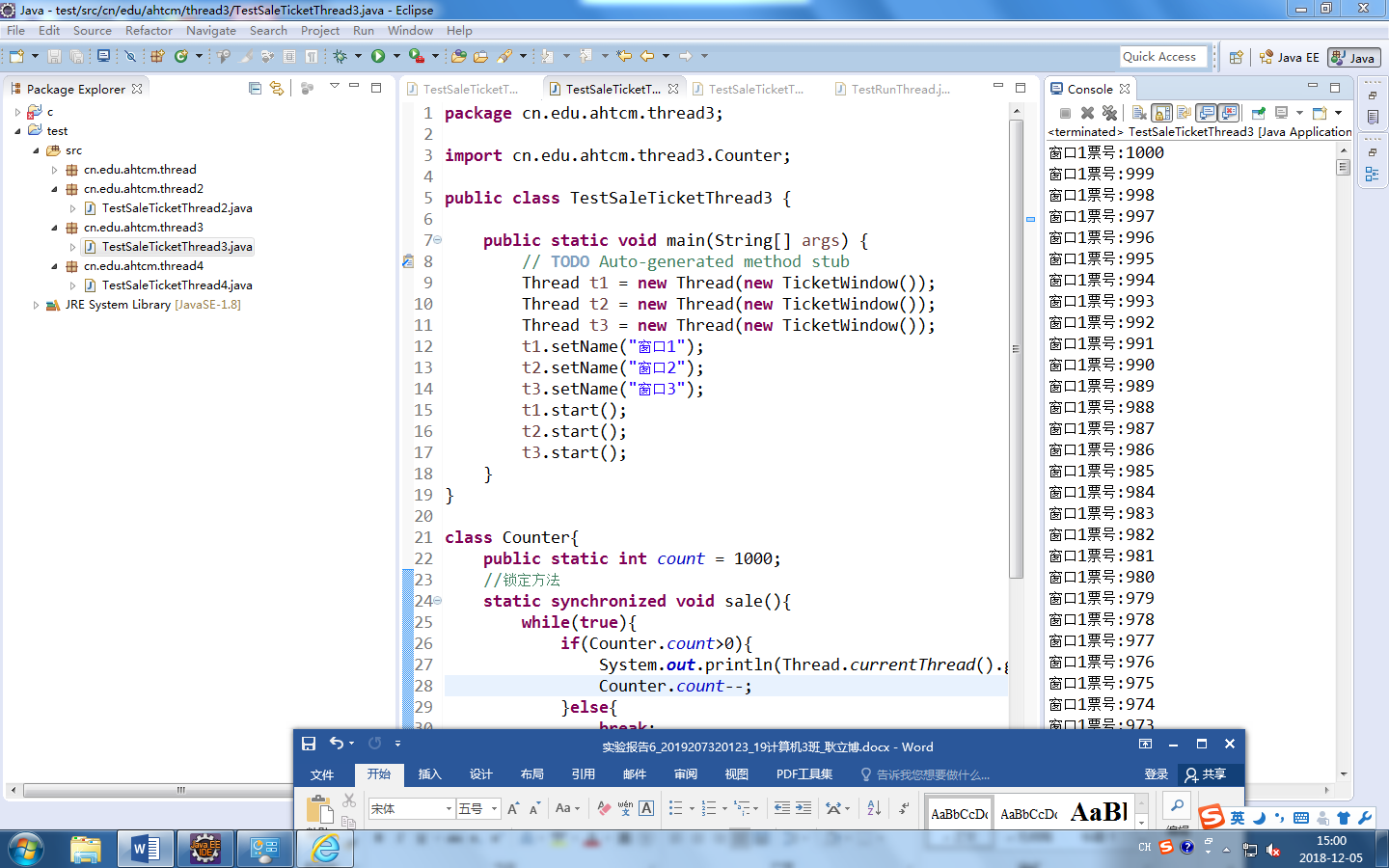
**public** **void** run() {

Counter.*sale*();

}

}

**【运行结果】**



**4.实验-4：**

**【源码】**

**package** cn.edu.ahtcm.thread4;

**import** java.util.Locale;

**import** java.util.concurrent.TimeUnit;

**import** java.util.concurrent.locks.Lock;

**import** java.util.concurrent.locks.ReentrantLock;

**import** cn.edu.ahtcm.thread4.Counter;

**public** **class** TestSaleTicketThread4 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Thread t1 = **new** Thread(**new** TicketWindow());

Thread t2 = **new** Thread(**new** TicketWindow());

Thread t3 = **new** Thread(**new** TicketWindow());

t1.setName("窗口1");

t2.setName("窗口2");

t3.setName("窗口3");

t1.start();

t2.start();

t3.start();

}

}

**class** Counter{

**public** **static** **int** *count* = 1000;

}

**class** TicketWindow **implements** Runnable{

Lock lock1 = **new** ReentrantLock();

@Override

**public** **void** run() {

**while**(**true**){

lock1.lock();

**if** (Counter.*count*>0){

**try** {

Thread.*sleep*(100);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

System.***out***.println(Thread.*currentThread*().getName()+':'+Counter.*count*);

Counter.*count*--;

}

lock1.unlock();

}

}

}

**【运行结果】**

