第二次作业：生产者与消费者问题

小组成员：修玥 石雨宁 王超 周学士

分工如下： 代码编写：石雨宁

文字编辑：修玥

运行：修玥 石雨宁 王超 周学士（由于运行结果一样 只附一张图片）

生产者消费者问题具体来讲，就是在一个系统中，存在生产者和消费者两种角色，生产者生产消费者需要的资料，消费者把资料做成产品。举例来说，一个变量，生产者不断增加这个变量，消费者不断减少这个变量。例如：火车票选票等。

代码如下：

package com.test; public class ProducerAndConsumer { public static void main(String[] args) { Warehouse warehouse = new Warehouse(); Producer p1 = new Producer(warehouse); Consumer c1 = new Consumer(warehouse); Thread pt1 = new Thread(p1); pt1.setName("生产者1"); Thread pt2 = new Thread(p1); pt2.setName("生产者2"); Thread ct1 = new Thread(c1); ct1.setName("消费者1"); pt1.start(); pt2.start(); ct1.start(); }} class Producer implements Runnable{ Warehouse warehouse; public Producer(Warehouse warehouse) { this.warehouse = warehouse; } @Override public void run() { while(true) { try { Thread.sleep(100); } catch (InterruptedException e) { e.printStackTrace(); } warehouse.add(); } }}class Consumer implements Runnable{ Warehouse warehouse; public Consumer(Warehouse warehouse) { this.warehouse = warehouse; } @Override public void run() { while(true) { try { Thread.sleep(100); } catch (InterruptedException e) { e.printStackTrace(); } warehouse.reduce(); } }}class Warehouse{ private int product; public synchronized void add() { if(product >= 20) { try { wait(); } catch (InterruptedException e) { e.printStackTrace(); } }else { product++; System.out.println(Thread.currentThread().getName()+ "生产了第" + product + "个商品"); notifyAll(); } } public synchronized void reduce(){ if(product <= 0) { try { wait(); } catch (InterruptedException e) { e.printStackTrace(); } }else { System.out.println(Thread.currentThread().getName()+ "消费了第" + product + "个商品"); product--; notifyAll(); } }}

运行结果如图：

