**第二次作业：生产者和消费者（2）**

**小组成员：王航 蒋馨 刘思言 马瑞轩 李海洋**

**分工如下：代码编写：蒋馨**

**文字编辑：刘思言 马瑞轩**

**运行：蒋馨 马瑞轩 刘思言 李海洋 王航（由于结果一样，只附一张图片）**

**生产者消费者模式是并发、多线程编程中经典的设计模式。就是一个类负责生产，一个类负责消费。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，火车票抢票等。**

**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--;**

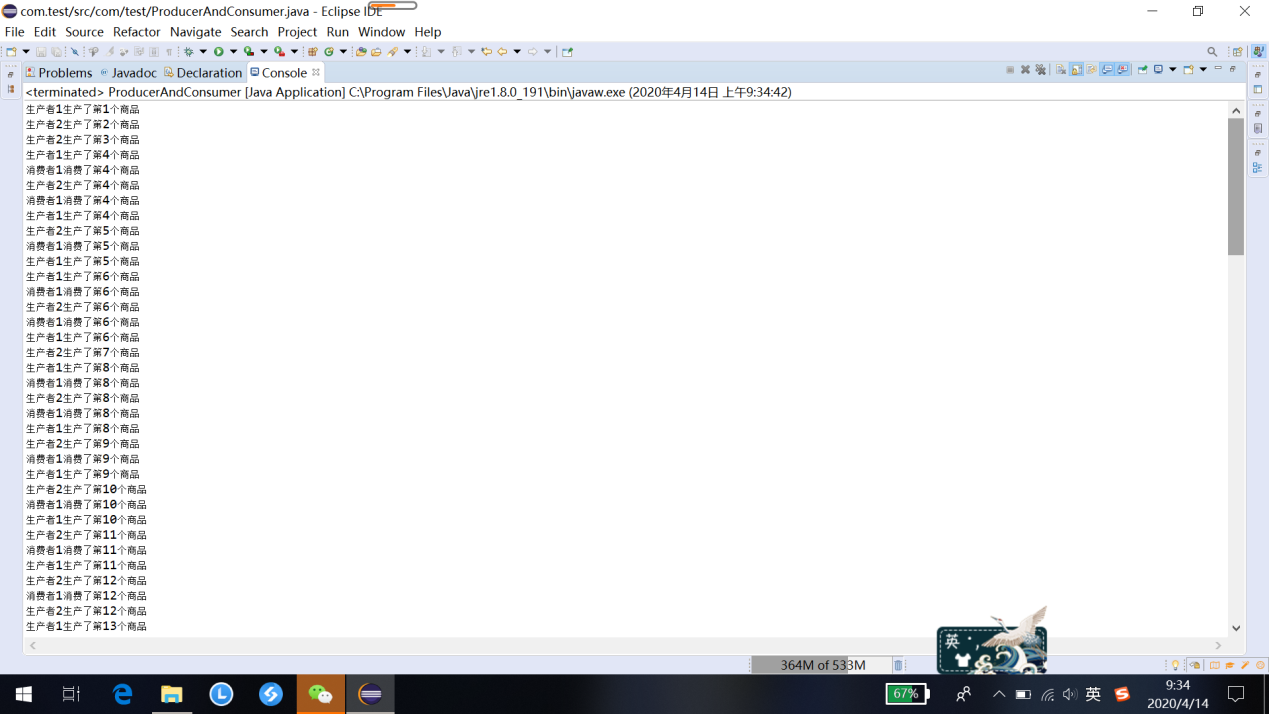
**notifyAll();**

**}**

**}**

**}**

**运行结果如下（由于很长，只截取了一页）：**

****