**线程并发管理**

线程并发管理：

线程并发管理就是控制线程的执行，因为线程执行时不可控制的，实际应用中，如果不控制，就会出现先消费后生产的情况，都没有生产这件商品，但是却已经消费了。

以生产和消费为例：

第一步：

先得到一个生产的集合，存放生产的商品

ArrayList list = **new** ArrayList();

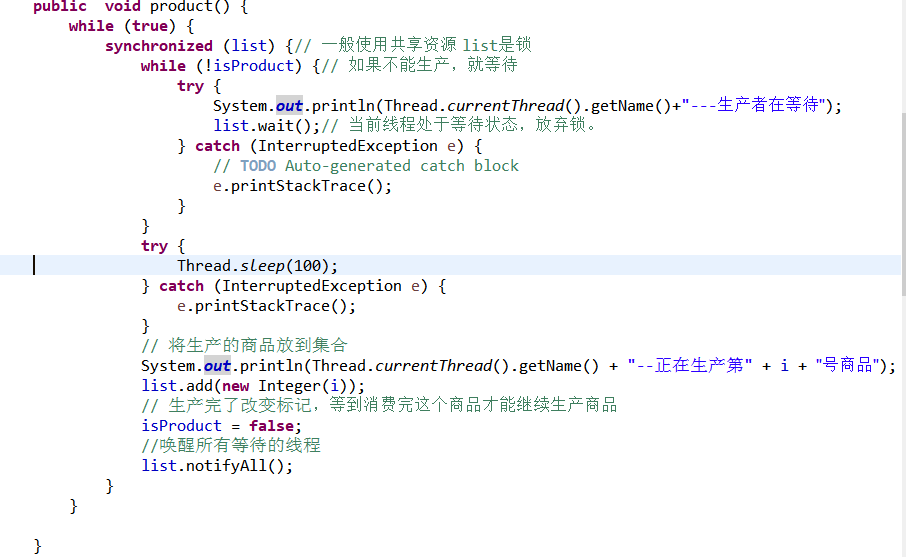
第二步：

定义生产的标记，因为实际中，不可能一直生产，只有需求时才会生产

**boolean** isProduct = **true**;

第三步：

定义生产的行为：



先给执行代码块加上一个同步块锁，商品集合就是锁，当标记为false时就是不能生产，此时就等待，然后使用wait（）放开锁，当标记为true时，就是能开始生产，然后将生产的商品放到集合中，然后给标记改为false，因为要卖完才能继续生产，然后使用notifyAll（）唤醒所有在等待的线程重新抢资源

定义消费的行为：



方法同上，当标记为true时就是不能消费，此时就等待，然后使用wait（）放开锁，当标记未false时，就是能开始消费，然后将集合中的商品消费掉，也就是remove，然后给标记改为true，

表示消费完可以继续生产了，再用notifyAll（）唤醒所有在等待的线程重新抢资源。

判断条件是while不能为if的原因时：当唤醒所有等待时，if条件不会重新判断条件，会直接执行下面的代码，所以会出现线程安全问题，但while就不会了，因为他是一个循环，唤醒之后会重新执行循环，再判断循环条件，如果不符合才会执行下面代码块