

二级Java考试辅导教程：6.2多线程的互斥与同步[2] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022__E4_BA_8C_E7_BA_A7Java_c97_138764.htm synchronized 除了象上面讲的放在对象前面限制一段代码的执行外，还可以放在方法声明中

，表示整个方法为同步方法。 `public synchronized void push(char c){ ... }` 如果synchronized用在类声明中，则表明该类中的所有方法都是synchronized的。 6.2.2多线程的同步 本节将讨论如何控制互相交互的线程之间的运行进度,即多线程之间的同步问题,下面我们将通过多线程同步的模型:生产者-消费者问题来说明怎样实现多线程的同步。 我们把系统中使用某类资源的线程称为消费者，产生或释放同类资源的线程称为生产者。 在下面的Java的应用程序中，生产者线程向文件中写数据，消费者从文件中读数据，这样，在这个程序中同时运行的两个线程共享同一个文件资源。通过这个例子我们来了解怎样使它们同步。 类Producer是生产者模型，其中的run()方法中定义了生产者线程所做的操作，循环调用push()方法,将生产的20个字母送入堆栈中，每次执行完push操作后，调用sleep()方法睡眠一段随机时间，以给其他线程执行的机会。 类Consumer是消费者模型，循环调用pop()方法,从堆栈中取出一个数据，一共取20次，每次执行完pop操作后，调用sleep()方法睡眠一段随机时间，以给其他线程执行的机会。 程序执行结果 Produced:V Consumed:V Produced:E

Consumed:E Produced:P Produced:L ... Consumed:L Consumed:P 在上述的例子中，通过运用wait()和notify()方法来实现线程的同步，在同步中还会用到notifyAll()方法，一般来说，每个共

享对象的互斥锁存在两个队列，一个是锁等待队列，另一个是锁申请队列，锁申请队列中的第一个线程可以对该共享对象进行操作，而锁等待队列中的线程在某些情况下将移入到锁申请队列。下面比较一下wait()、notify()和notifyAll()方法：

(1) wait, notify, notifyAll必须在已经持有锁的情况下执行,所以它们只能出现在synchronized作用的范围内，也就是出现在用synchronized修饰的方法或类中。

(2) wait的作用:释放已持有的锁,进入等待队列. 来源：www.examda.com

(3) notify的作用:唤醒wait队列中的第一个线程并把它移入锁申请队列.

(4) notifyAll的作用:唤醒wait队列中的所有的线程并把它们移入锁申请队列.

注意：1) suspend()和resume() 在JDK1.2中不再使用suspend()和resume(),其相应功能由wait()和notify()来实现。

2) stop() 在JDK1.2中不再使用stop(),而是通过标志位来使程序正常执行完毕。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com