

Java线程通信简单调试方法介绍Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_Java_E7_BA_BF_E7_A8_8B_c104_645012.htm Java线程通信在我们使用的时候需要长期的学习，在学习的过程中我们就要详细的看看如何才能更好的掌握相关语言的使用问题。首先我们来看看具体的相关问题处理。使当前Java线程通信(即调用该方法的线程)暂停执行一段时间，让其他线程有机会继续执行，但它并不释放对象锁。也就是如果有Synchronized同步块，其他线程仍然不同访问共享数据。注意该方法要捕获异常 比如有两个线程同时执行(没有Synchronized)，一个线程优先级为MAX_PRIORITY，另一个为MIN_PRIORITY，如果没有Sleep()方法，只有高优先级的线程执行完成后，低优先级的线程才能执行；但当高优先级的线程sleep(5000)后，低优先级就有机会执行了。总之，sleep()可以使低优先级的Java线程通信得到执行的机会，当然也可以让同优先级、高优先级的线程有执行的机会。join() join()方法使调用该方法的Java线程通信在此之前执行完毕，也就是等待调用该方法的线程执行完毕后再往下继续执行。注意该方法也要捕获异常。yield()它与sleep()类似，只是不能由用户指定暂停多长时间，并且yield()方法只能让同优先级的线程有执行的机会。wait()和notify()、notifyAll() 这三个方法用于协调多个Java线程通信对共享数据的存取，所以必须在Synchronized语句块内使用这三个方法。前面说过 Synchronized这个关键字用于保护共享数据，阻止其他线程对共享数据的存取。但是这样程序的流程就很不灵活了，如何才能在当前线程还没退出 Synchronized数

据块时让其他线程也有机会访问共享数据呢？此时就用这三个方法来灵活控制。wait()方法使当前线程暂停执行并释放对象锁标志，让其他线程可以进入Synchronized数据块，当前线程被放入对象等待池中。当调用notify()方法后，将从对象的等待池中移走一个任意的线程并放到锁标志等待池中，只有锁标志等待池中的线程能够获取锁标志；如果锁标志等待池中没有任何Java线程通信，则notify()不起作用。notifyAll()则从对象等待池中移走所有等待那个对象的线程并放到锁标志等待池中。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