1)现在有 T1、T2、T3 三个线程，你怎样保证 T2 在 T1 执行完后执行，T3 在 T2 执行完后执行？

这个线程问题通常会在第一轮或电话面试阶段被问到，目的是检测你对”join”方法是否熟悉。这个多线程问题比较简单，可以用 join 方法实现。

2)在 Java 中 Lock 接口比 synchronized 块的优势是什么？你需要实现一个高效的缓存，它允许多个用户读，但只允许一个用户写，以此来保持它的完整性，你会怎样去实现它？

lock 接口在多线程和并发编程中最大的优势是它们为读和写分别提供了锁，它能满足你写像 ConcurrentHashMap 这样的高性能数据结构和有条件的阻塞。Java 线程面试的问题越来越会根据面试者的回答来提问。我强烈建议在你去参加多线程的面试之前认真读一下 Locks，因为当前其大量用于构建电子交易终统的客户端缓存和交易连接空间。

3)在 java 中 wait 和 sleep 方法的不同？

通常会在电话面试中经常被问到的 Java 线程面试问题。最大的不同是在等待时 wait 会释放锁，而 sleep 一直持有锁。Wait 通常被用于线程间交互，sleep 通常被用于暂停执行。

4）用 Java 实现阻塞队列。

这是一个相对艰难的多线程面试问题，它能达到很多的目的。第一，它可以检测侯选者是否能实际的用 Java 线程写程序；第二，可以检测侯选者对并发场景的理解，并且你可以根据这个问很多问题。如果他用 wait()和 notify()方法来实现阻塞队列，你可以要求他用最新的 Java 5 中的并发类来再写一次。

5）用 Java 写代码来解决生产者——消费者问题。

与上面的问题很类似，但这个问题更经典，有些时候面试都会问下面的问题。在 Java 中怎么解决生产者——消费者问题，当然有很多解决方法，我已经分享了一种用阻塞队列实现的方法。有些时候他们甚至会问怎么实现哲学家进餐问题。

6）用 Java 编程一个会导致死锁的程序，你将怎么解决？

这是我最喜欢的 Java 线程面试问题，因为即使死锁问题在写多线程并发程序时非常普遍，但是很多侯选者并不能写 deadlock free code（无死锁代码？），他们很挣扎。只要告诉他们，你有 N 个资源和 N 个线程，并且你需要所有的资源来完成一个操作。为了简单这里的 n 可以替换为 2，越大的数据会使问题看起来更复杂。通过避免 Java 中的死锁来得到关于死锁的更多信息。

7) 什么是原子操作，Java 中的原子操作是什么？

非常简单的 java 线程面试问题，接下来的问题是你需要同步一个原子操作。

8) Java 中的 volatile 关键是什么作用？怎样使用它？在 Java 中它跟 synchronized 方法有什么不同？

自从 Java 5 和 Java 内存模型改变以后，基于 volatile 关键字的线程问题越来越流行。应该准备好回答关于 volatile 变量怎样在并发环境中确保可见性。

9) 什么是竞争条件？你怎样发现和解决竞争？

这是一道出现在多线程面试的高级阶段的问题。大多数的面试官会问最近你遇到的竞争条件，以及你是怎么解决的。有些时间他们会写简单的代码，然后让你检测出代码的竞争条件。可以参考我之前发布的关于 Java 竞争条件的文章。在我看来这是最好的 java 线程面试问题之一，它可以确切的检测候选者解决竞争条件的经验，or writing code which is free of data race or anyother race condition。关于这方面最好的书是《Concurrency practices in Java》。

10) 你将如何使用 threaddump？你将如何分析 Thread dump？

在 UNIX 中你可以使用 kill -3，然后 thread dump 将会打印日志，在 windows 中你可以使用”CTRL+Break”。非常简单和专业的线程面试问题，但是如果他问你怎样分析它，就会很棘手。

11) 为什么我们调用 start()方法时会执行 run()方法，为什么我们不能直接调用 run()方法？

这是另一个非常经典的 java 多线程面试问题。这也是我刚开始写线程程序时候的困惑。现在这个问题通常在电话面试或者是在初中级 Java 面试的第一轮被问到。这个问题的回答应该是这样的，当你调用 start()方法时你将创建新的线程，并且执行在 run()方法里的代码。但是如果你直接调用 run()方法，它不会创建新的线程也不会执行调用线程的代码。阅读我之前写的《start 与 run 方法的区别》这篇文章来获得更多信息。

12) Java 中你怎样唤醒一个阻塞的线程？

这是个关于线程和阻塞的棘手的问题，它有很多解决方法。如果线程遇到了 IO 阻塞，我并且不认为有一种方法可以中止线程。如果线程因为调用 wait()、sleep()、或者 join()方法而导致的阻塞，你可以中断线程，并且通过抛出 InterruptedException 来唤醒它。我之前写的《How to deal with blocking methods in java》有很多关于处理线程阻塞的信息。

13)在 Java 中 CycliBarriar 和 CountdownLatch 有什么区别？

这个线程问题主要用来检测你是否熟悉 JDK5 中的并发包。这两个的区别是 CyclicBarrier 可以重复使用已经通过的障碍，而 CountdownLatch 不能重复使用。

14) 什么是不可变对象，它对写并发应用有什么帮助？

另一个多线程经典面试问题，并不直接跟线程有关，但间接帮助很多。这个 java 面试问题

可以变的非常棘手，如果他要求你写一个不可变对象，或者问你为什么 String 是不可变的。

15) 你在多线程环境中遇到的常见的问题是什么？你是怎么解决它的？

多线程和并发程序中常遇到的有 Memory-interface、竞争条件、死锁、活锁和饥饿。问题是没有止境的，如果你弄错了，将很难发现和调试。这是大多数基于面试的，而不是基于实际应用的 Java 线程问题。