多线程和并发问题已成为各种 Java 面试中必不可少的一部分。如果你准备参加投行的 Java 开发岗位面试，比如巴克莱银行（Barclays）、花旗银行（Citibank）、摩根史坦利投资公司（Morgan Stanley），你会遇到很多有关多线程的面试题。多线程和并发是投行面试的热门知识点，尤其是在面试有关电子交易开发工作时，他们喜欢用棘手的 Java 线程面试题轰炸面试者。他们希望确保面试者对 Java 多线程和并发有扎实的知识基础，因为他们大多数关注高性能带来的竞争优势。

举个例子，直接市场准入模式（Direct to Market，DMA）使用高容量低延迟的电子交易系统，通常来说是并发的。大多数时间他们致力于微秒级的延迟，所以掌握如何有效地降低延迟、提高吞吐量非常重要。

有一些 Java 线程面试题是我特别中意的。我并不会直接给你答案，而是尽可能给你指点。我会之后补充上详细答案，正如我在其他文章中那样。

JDK 1.5 中引入并发包之后，并发工具和并发集合备受欢迎，比如 ThreadLocal、 BlockingQueue、Counting Semaphore 和 ConcurrentHashMap，与这些工具相关的面试题也越来越多。

Java 8 和 Java 9 也是这种情况。围绕 lambda 表达式、并行流（parallel streams）、新的 Fork/Join 线程池、CompletableFuture 的问题在 2018 年不断涌现，2019 年还将持续。今后你也应该对这些知识点有所准备。

**15 个 Java 线程面试题和答案**

总之不要考虑那么多，下面是各种投行，比如巴克莱银行（Barclays）、花旗银行（Citibank）、摩根史坦利投资公司（Morgan Stanley）等等，面试 Java 开发者时常问的 Java 多线程和并发问题。

**1. 现在有线程 T1、T2 和 T3。你如何确保 T2 线程在 T1 之后执行，并且 T3 线程在 T2 之后执行？**

这个线程面试题通常在第一轮面试或电话面试时被问到，这道多线程问题为了测试面试者是否熟悉 join 方法的概念。答案也非常简单——可以用 Thread 类的 join 方法实现这一效果。

**2. Java 中新的 Lock 接口相对于同步代码块（synchronized block）有什么优势？如果让你实现一个高性能缓存，支持并发读取和单一写入，你如何保证数据完整性。**

多线程和并发编程中使用 lock 接口的最大优势是它为读和写提供两个单独的锁，可以让你构建高性能数据结构，比如 ConcurrentHashMap 和条件阻塞。  
这道 Java 线程面试题越来越多见，而且随后的面试题都基于面试者对这道题的回答。  
我强烈建议在任何 Java 多线程面试前都要多看看有关锁的知识，因为如今电子交易系统的客户端和数据交互中，锁被频繁使用来构建缓存。

**3. Java 中 wait 和 sleep 方法有什么区别？**

我们来看看另一个经常被问到的线程面试题。这道题常出现在电话面试中。两者主要的区别就是等待释放锁和监视器。sleep方法在等待时不会释放任何锁或监视器。wait 方法多用于线程间通信，而 sleep 只是在执行时暂停。可以看我另一篇有关[Java 中 wait 和 sleep](http://javarevisited.blogspot.sg/2011/12/difference-between-wait-sleep-yield.html)的文章。