

一面：

- . `ArrayList` 和 `linkedlist` 区别。`ArrayList` 是否会越界。
- . `ArrayList` 和 `hashset` 有何区别。`hashset` 存的数是有序的么。
- . `volatile` 和 `synchronized` 区别
- . 多态的原理
- . 数据库引擎 `Innodb` 和 `myisam` 区别
- . `Redis` 的数据结构
- . `Redis` 是基于内存的么
- . `Redis` 的 `list` `zset` 的底层实现
- . `http` 和 `https` 的区别，`tcp` 握手过程
- . `jvm` 垃圾回收算法手写冒泡
- . 手写单例包括多线程下
- . `Java` 线程间怎么实现同步，`notify()` 与 `notifyAll()` 的区别
- . 数据库的悲观锁和乐观锁应用场景。
- . 排序算法的复杂度，快速排序非递归实现。
- . 海量数据过滤，黑名单过滤一个 `url`。

二面：

- . `list` `set` `map` 底层使用什么实现的有哪些典型实现
- . `hashmap` 扩容是怎么扩容的，为什么是 2 的幂
- . `concurrenthashmap` 为什么线程安全，采用了什么措施应对高并发
- . 线程池的参数有什么意义

- . *Springmvc* 请求流程
- . *Spring IOC, autowired* 如何实现
- . *Spring boot*
- . *SpringCloud* 的基本架构设计
- . *Dubbo* 和 *SpringCloud* 的区别在哪里，优劣势
- . 说说一致性 *Hash* 算法

三面：

- . 分布式架构设计哪方面比较熟悉
- . 讲讲你对 *CDN* 的了解，与分布式缓存和本地缓存的区别
- . 多线程和高并发有什么区别
- . 高并发下有哪些常用的技术解决方案，举三个高并发场景设计例子
- . 说一个你对 *JVM* 优化的实际案例，包括实际步骤和方法
- . *Docker* 有使用过和了解吗？*Docker* 和 *JVM* 的区别是什么？
- . *Docker* 的基本架构和使用场景？
- . 负载均衡有接触过哪些开源框架，优劣势是什么？
- . 数据库分库分表需要怎样来实现？
- . 数据库端的常用优化策略？
- . 如果让你来设计秒杀系统，你的设计思路是什么，为什么要这样设计？

面试总结：

- . *java* 的基础知识点，主要围绕在集合类和多线程等：*ArrayList*、*LinkedList*、*HashSet*、*HashMap* 的数据结果，以及如何扩容、以及 *ConcurrentHashMap* 相关的多线程安全等。
 - . *JVM* 的内存分配、几个常见的垃圾回收算法以及原理、还有对应的 *JVM* 优化参数需要牢记。
 - . 网络：*TCP* 的三次握手等网络都必问，重点掌握网络协议。
 - . *Redis*：作为分布式缓存的主力，基本也是 *BAT* 每次必考，重点是 *Redis* 的数据结构、内存、算法、持久化，以及与别的缓存 *memcached* 的优劣势。
 - . 多线程：状态流转、多线程的实现，以及与高并发的区别等。
 - . *Spring* 框架问得是最多的，*BAT* 非常喜欢问，重点掌握。
 - . 最后就是分布式架构设计
-
- 常用的分布式架构设计方案：单点登录、分布式缓存、存储、消息的选型，还有就是数据库端的优化方案(需要提前了解)。
 - 最好能提前了解深入一个类似秒杀这样的项目，如果面试官问到类似的项目，你能把设计思路讲出来，这对你的面试结果是很大的加分项。