

Java 后端社招两年半，拿了 10 多个 offer!

大家好，我是工作时长两年半的 Java 开发工程师，喜欢摸鱼、划水、code、开会，接下来我将介绍下我的社招求职心得和面经。

方法和知识点我就不多介绍了，Java 面试指南网站 (javaguide.cn) 里写的很全面，更多是想说一些我的感受和建议，分为以下两个方面：

- **面经**：白龙马云行科技、字节、淘宝、快手、百度、拼多多等上十家拿到 offer 或者谈薪阶段主动终止的公司的完整面经。
- **经验分享**：心态、简历、算法、技巧。

面经总结

面经部分都是拿到 offer 或者谈薪阶段主动终止的公司，其他小公司或创业公司都是为了练手，面试题没有普遍性的都没有列举出来，面试问题写的少的都是问项目、业务比较多的。算法题只在 leetcode 上找原题，没有贴出链接的就是 leetcode 上没有的。

白龙马云行科技

一面

主要按照项目职责描述一条一条问，考察设计思路：

- 1、项目中的难点；
- 2、介绍下 xxl-job 的实现原理；
- 3、disruptor 的使用场景；
- 4、生产者出现堆积的话怎么解决；
- 5、生产消费过程中如果服务出现异常，如何恢复；
- 6、介绍下 MySQL 的索引；
- 7、B+树的特征，为什么 MySQL 要使用 B+树；
- 8、如何做到监控线程池；
- 9、RPC 服务超时的情况如何止损；
- 10、如何实现异步多线程追踪；
- 11、Redis 超时问题，排查过程中考虑的角度，使用了哪些方法；
- 12、服务设计时怎么体现高可用性、易用性；
- 13、用户身份唯一性确认；
- 14、对我们公司有了解吗；
- 15、Integer a=200; Integer b=200, a 和 b equals 是否为 true;
- 16、介绍下缓存雪崩、击穿、穿透，如何避免；
- 17、工作中是否遇到过 CPU 过高、内存过高的情况，怎么排查问题原因的；
- 18、单链表和双链表的优缺点；
- 19、判断单链表是否有环，算法的时间复杂度；

20、ArrayList、HashMap、TreeSet 哪些属于有序的，哪些是稳定的；

21、线程池的工作原理；

22、线程池中关于时间的参数起什么作用；

23、项目中有没有用过并行流；

24、ForkJoin 线程池默认线程数；

25、新生代和老年代的特点，分别使用哪些算法；

26、软引用和弱引用的回收时机。

二面

1、离职原因；

2、现在公司的技术氛围；

3、下一份工作的偏好；

4、未来发展规划；

5、数组和链表遍历哪个更快（考虑存储）；

6、Integer 为什么要有-128~127 的常量；

7、Java 已有可达性分析和 GC，为什么还要设定软引用、弱引用；

8、ThreadLocal 为什么使用弱引用；

9、Redis 清理过期数据的策略；

10、一个 G 数字文件，100M 内存，排序。

瓴岳科技

一面

1、怎样参与 CR；

2、印象深的好的代码、不好的代码；

3、ThreadLocal 不 remove 会有什么问题；

4、Redis 7.0 的新特性；

5、介绍下 Disruptor，有哪些组件使用到了；

6、Disruptor 底层实现，为什么快；

7、Java 的锁是怎么实现的；

8、AQS 中线程并发时同步队列的放入情况；

9、Netty 如何使用 http 复用；

10、项目中线程池的用法、场景；

11、算法：找到二叉树指定节点中序遍历的下一个节点（类似剑指 Offer II 053. 二叉搜索树中的中序后继 <<https://leetcode.cn/problems/P5rCT8/>>）。

二面

1、为什么要引入生产者-消费者模式；

2、项目中 ES 的使用场景；

3、项目新老服务迁移的过程；

4、单元测试的情况；

5、工作以来最大的收获；

6、介绍下分布式事务；

7、项目中怎么保证分布式一致性；

8、介绍下 RocketMQ；

9、算法：155. 最小栈 <<https://leetcode.cn/problems/min-stack/>> ；

10、算法：爬楼梯，每次可走 3 步和 5 步（类似70. 爬楼梯 <<https://leetcode.cn/problems/climbing-stairs/>>，要求动态规划实现，五阶递推数列）。

三面

1、项目中可优化的点，对项目的思考；

2、生产者-消费者模式怎么保证消息不丢失；

3、微服务的概念；

4、设计题：设计一个消息队列；

5、算法：128. 最长连续序列 <<https://leetcode.cn/problems/longest-consecutive-sequence/>> 。

字节

一面

1、服务的可用性监控；

2、服务故障的保障性；

3、MQ 出现消费积压，如何处理；

4、线程池工作的机制；

5、核心线程可能会被销毁吗；

6、核心线程数和最大线程数的区别；

7、阻塞队列的特性，底层实现；

8、ThreadLocal 数据结构和工作原理；

9、弱引用的特性；

10、为什么 ThreadLocal 的 key 使用弱引用；

11、事务 ACID 的特性；

12、事务隔离可能出现的问题；

13、介绍下 Redis 的高可用；

14、Redis 的主从、哨兵模式；

15、算法：判断一个树是平衡二叉搜索树（110. 平衡二叉树 <<https://leetcode.cn/problems/balanced-binary-tree/>>、98. 验证二叉搜索树 <<https://leetcode.cn/problems/validate-binary-search-tree/>> 两道题的合并）。

二面

1、项目的具体模块设计；

2、RPC 中 tcp 协议和 http 协议的区别；

3、http 2.0 的优化；

4、介绍下多路复用；

5、MySQL 分表；

6、项目中用到缓存的场景；

7、Redis 的特性；

8、Redis 支持高并发的原理；

9、Redis 数据丢失后如何恢复；

10、Redis 的持久化策略；

11、执行 RDB 操作时如何对外提供服务；

12、Redis 淘汰策略；

13、算法：146. LRU 缓存 <<https://leetcode.cn/problems/lru-cache/>>。

三面

问题：

1、业务讨论；

2、系统的架构；

3、市面上流行的规则引擎有哪些；

4、为什么选择现在的规则引擎；

5、怎么控制规则变更过程中的风险；

6、项目可进一步优化点；

7、算法：22. 括号生成 <<https://leetcode.cn/problems/generate-parenttheses/>>。

一面

- 1、==和 equals;
- 2、重写 equals 的场景;
- 3、深拷贝、浅拷贝、零拷贝;
- 4、介绍下 Java 中的动态代理;
- 5、序列化, 有哪些方式, 选型;
- 6、maven 冲突, 怎么发现, 如何解决;
- 7、Linux 大文件查看, du -sh;
- 8、大文件按行去重。
- 9、算法: 2 种以上实现单例的方式;
- 10、算法: 0~n-1 选 m 个不同的随机数;
- 11、算法: 字符串转 long (类似8. 字符串转换整数 (atoi) <<https://leetcode.cn/problems/string-to-integer-atoi/>>)。

二面

- 1、ES 正排和倒排的区别;
- 2、ES 如何优化 query 查询速度;
- 3、ES 什么时候增加、减少节点;
- 4、ES 如何引入排序算法;
- 5、系统如何限制 TP999 时间;
- 6、hashmap 在多线程下可能出现的情况;
- 7、hash 冲突的意思;
- 8、cookie 和 session 的区别;
- 9、如果已经禁用了 cookie, 还能使用 session 吗;
- 10、如何实现单点登录;
- 11、安全性的保证;
- 12、同步、异步、阻塞、非阻塞;
- 13、线上 Java 服务有问题如何排查;
- 14、git merge 和 rebase 的区别;
- 15、一个没有源码的应用, 如何替换数据库地址;
- 16、算法: 剑指 Offer 62. 圆圈中最后剩下的数字 <<https://leetcode.cn/problems/yuan-quan-zhong-zui-hou-sheng-xia-de-shu-zi-lcof/>> (约瑟夫环)。

三面

- 1、为什么看新的工作机会；
- 2、在现在公司工作经历；
- 3、介绍下最近的项目；
- 4、项目中需要优化的点；
- 5、项目中策略的效果评估；
- 6、工作期间觉得成长比较快的事情；
- 7、做过哪些搜索引擎相关的工作。

菜鸟

一面

- 1、介绍几种 NIO 模式；
- 2、Redis 中使用的 NIO，与其他 NIO 对比的优势；
- 3、缓存击穿、穿透、雪崩，解决方法；
- 4、Redis 能否存储 null；
- 5、MySQL 的性能瓶颈；
- 6、介绍下分库、分表，常见操作。
- 7、编程题：小猫抢鱼，三个线程表示三只猫，抢一堆鱼，每次抢一半的鱼，直到为 0。

二面

面了两个小时，问了很多很多技术问题，挖的比较深入：

- 1、大量业务相关问题（考察知识边界）；
- 2、项目中的难点；
- 3、介绍下 kylin；
- 4、disruptor 不同等待策略；
- 5、为什么选择 disruptor；
- 6、介绍下 disruptor 的底层实现；
- 7、disruptor 的无锁体现在哪里；
- 8、disruptor 游标的技巧；
- 9、disruptor 如何解决 JDK 当时无法处理的伪共享；
- 10、CompletableFuture 和 Future 的区别；
- 11、CompletableFuture 提供的创新设计模式—promise 模式；

12. 关注的 CPU 使用率指标具体是哪个；
13. object 的 equals 和 hashCode 方法，能不能只改 equals；
14. hashmap 的数据结构，扩容时机；
15. concurrentHashMap 和 hashmap 的区别，为什么前者的 key 和 value 都不能为 null；
16. concurrentHashMap 怎么保证的线程安全；
17. 怎么启动一个线程，怎么让线程停下来；
18. 怎么让变量线程安全；
19. volatile 怎么做到可见性；
20. CPU 的真实存储结构；
21. 平时的学习途径；
22. 对哪方面研究的比较深；
23. 了解哪些设计模式；
24. 使用 arthas 的场景；
25. arthas 排查线程占比过高的实现原理；
26. 编程题：实现一个方法，判断两个数组是否相等，元素类型为泛型。

快手

一面

- 1、ConcurrentHashMap 如何实现线程安全，如何保证安全扩容；
- 2、synchronized 锁升级的过程；
- 3、Java 中有哪些乐观锁；
- 4、读写锁和 synchronized 有哪些区别；
- 5、synchronized 为什么可以做到异常自动释放锁；
- 6、介绍下 ReentrantLock；
- 7、AQS 同步队列中的线程什么情况下知道可以激活；
- 8、Redis String 类型底层数据结构；
- 9、Redis 持久化方式；区别、优缺点；根据业务判断下选择哪种方式比较好；
- 10、OOM 的排查；
- 11、JDK 版本与 GC；
- 12、G1 的优势；
- 13、算法：2. 两数相加 <<https://leetcode.cn/problems/add-two-numbers/>> 。

二面

- 1、项目中 kylin 的使用场景；
- 2、项目中 CompletableFuture 的使用场景；
- 3、生产并发过高的问题，如何解决的；
- 4、Spring 中的 BeanFactory 和 FactoryBean 的区别；
- 5、线程池的核心参数，任务执行过程中的变化；
- 6、MySQL 索引的底层数据结构；
- 7、介绍 disruptor 框架的特点；介绍下使用场景；
- 8、Spring AOP 的实现原理；
- 9、动态代理和静态代理的区别；
- 10、介绍下 RPC 中调用方请求服务方接口中间的过程；
- 11、服务方轮询的地方，如何确定具体提供服务的 IP；
- 12、多个 PRC 服务如何保证一致性；
- 13、设计题：班级、学生、课程，设计 MySQL 表；
- 14、设计题：实现快手关注中心功能，发布者发布的内容能被看到，观看者能看到所有关注的人发布的内容且形成信息流列表；
- 15、平时的学习渠道和方法，举一些例子；
- 16、换工作的原因，对下一份工作的期望。

三面

- 1、规则引擎的选型与实现；
- 2、Redis 分片和复制的机制；
- 3、排查 Java 服务 CPU 占用率百分之百；
- 4、GC 识别垃圾对象的方法；
- 5、常用的 GC；

百度

一面

问题：

- 1、如何保证分布式数据一致性；
- 2、MQ 消息发送失败、消费失败如何恢复；
- 3、Redis 集群模式下数据在不同机器上的影响；
- 4、Redis 的 NIO 模型；

5、百度上亿的广告物料，考虑到检索效率、更新，如何设计比较好；

6、介绍下 JMM 模型，作用，场景；

7、synchronized 升级的内容，锁升级过程；

8、介绍下 RocketMQ；

9、如何解决重复写、重复读、消息丢失的问题；

10、算法：445. 两数相加 II <<https://leetcode.cn/problems/add-two-numbers-ii/>> 。

二面

除了项目，全部都是数据库相关的问题：

1、分页查询大量数据，如何设计（避免出现页码过深，耗时很长的问题）；

2、limit 为什么会出现页码越大，耗时越长的的问题；

3、>、<条件会用到索引吗，处理过程；

4、那些条件会用到索引；

5、between and 会走索引吗？什么情况下不走索引；

6、between and 和 limit 的区别；

7、设计一个适用消费者模式的消费表，对外提供消费接口，分布式并行消费，保证不漏消费、不重复消费；

8、如何通过 MySQL 保证行的原子性；

9、Redis 如何解决原子性；

10、MySQL 如何做到 where 条件匹配多条但只更新一条（用 update limit）；

11、MySQL 表同时收到多条更新操作，如何处理；

12、update 时，行级锁是在什么时间点加上的。

三面

三面技术文的较少，闲聊偏多：

1、最近在看哪方面的知识；

2、介绍下对 Java 并发编程的理解；

3、单线程一定会比多线程慢吗？分别适用于什么场景；

4、实际工作中使用多线程的场景；

5、业务模式讨论；

6、觉得自己项目中做的比较出彩的点；

7、介绍下熟悉的设计模式；

8、下一份工作的目标。

拼多多

一面

- 1、描述下对 Spring IOC 的理解，与传统 new 的方式相比优势有哪些；
- 2、SpringBoot 如何实现每次 Autowired 都创建新的实例；
- 3、Mybatis 的 Transactional 注解实现事务的原理；
- 4、动态代理原理实现的注解有哪些注意事项；
- 5、MySQL 的几种 log 的作用；
- 6、聚簇索引和二级索引的区别；
- 7、varchar 和 text 类型的字段存储形式有什么区别；
- 8、text 类型的字段可以创建索引吗；
- 9、介绍下 Redis 管道通信；
- 10、算法：150. 逆波兰表达式求值 <<https://leetcode.cn/problems/evaluate-reverse-polish-notation/>> ；
- 11、算法：计算四个数字数组和为 0 的组数， $O(n^2)$ 时间复杂度。

二面

- 1、ES 的优化方式；
- 2、Redis 的哨兵机制；
- 3、分库分表；
- 4、MySQL 的事务隔离级别；
- 5、介绍下 InnoDB 的 mvcc；
- 6、算法：剑指 Offer 29. 顺时针打印矩阵 <<https://leetcode.cn/problems/shun-shi-zhen-da-yin-ju-zhen-lcof/>> 。

三面

主要是一些闲聊，问到了加班问题：

- 1、为什么不考虑读研；
- 2、校招时为什么要选择现在的公司；
- 3、在每个阶段的工作人员组成情况；
- 4、目前的状态，为什么想出来看机会；
- 5、对于工作地点是怎样的考虑。
- 6、对于拼多多的工作时间接受吗；

一面

- 1、git merge 原理;
- 2、jvm GC，垃圾回收器，为什么不用 G1;
- 3、SpringBoot 启动 装载 starter 过程;
- 4、concurrentHashMap 扩容过程;
- 5、slfj 选择 log 的机制;
- 6、clone()方法，为什么没人用;
- 7、hashCode，生成方式;
- 8、算法：判断整数奇偶;
- 9、算法：合并 map，求平均值，带嵌套;
- 10、算法：链表相加，实现 this.add(ListNode b)。

二面

- 1、几乎全部为业务、项目相关问题;
- 2、任务中断如何处理;
- 3、项目中用到的设计模式，解决的问题;
- 4、未来的规划;
- 5、对下一份工作的期望。

三面

现场面试:

- 1、项目流程，架构画图;
- 2、项目可优化的点;
- 3、项目效果;
- 4、算法：判断一个数是否能被每一位上的数整除(美丽数);
- 5、算法：56. 合并区间 <<https://leetcode.cn/problems/merge-intervals/>>。

小红书

一面

- 1、介绍下项目;
- 2、介绍下 hashmap 的原理;
- 3、为什么引入红黑树;

- 4、如果不用树，引入其他数据结构解决链表过长查询效率低；
- 5、CompletableFuture 的使用场景；
- 6、线程池的执行流程；
- 7、为什么线程池执行的机制设计成这样；
- 8、并发包下的工具类；
- 9、RPC 服务，A B C D 四个调用，A B C 之间没有依赖关系，D 需要等到 A B C 执行完之后再执行，有哪些实现方式；
- 10、MySQL 中(a,b,c)建立了联合索引，几种可以命中索引的场景；
- 11、如何减少回表；
- 12、select * 为什么会导回表；
- 13、服务中使用的 GC；
- 14、meta space 中存储的数据；
- 15、服务启动后，mate space 的空间不断增加的原因；
- 16、查看 mate space 的存储情况；
- 17、SpringBoot 动态代理的实现方式；默认使用方式的变化；
- 18、动态代理方式的性能区别；
- 19、Redis 查询突然变慢的原因；
- 20、项目中出现 Redis 连接超时的原因；
- 21、消息队列，rocketMQ 的实现原理；
- 22、disruptor 的使用，实现原理；
- 23、设计题：直播间大红包分发，10 亿红包，10 万观众，每个红包额度 1-100 元；
- 14、算法：剑指 Offer II 069. 山峰数组的顶部 <<https://leetcode.cn/problems/811id/>>。

二面

- 1、大量业务、项目问题，根据简历上的职责描述一条一条的问；
- 2、项目中解决问题；
- 3、介绍下 MySQL 分库分表，垂直、水平、服务端、客户端分表；
- 4、实现的方式有哪些；
- 5、ES 如何实现深度分页；
- 6、在团队中的角色；
- 7、离职的原因，未来的规划；
- 8、有了解竞品产品吗，了解到什么程度；

9、过往绩效情况。

三面

基本都是项目相关的问题：

- 1、项目中规则引擎的实现；
- 2、本地缓存存在的问题；
- 3、介绍下 disruptor；
- 4、学习途径，看过哪些书；
- 5、对下一份工作的期待。

滴滴

一面

- 1、针对项目不断延展，给的压力很大；
- 2、列举项目中的三个难点；
- 3、SQL 中为什么 in 比 between and 慢；
- 4、工作中觉得做的不足的地方；
- 5、如何提升业务理解能力；
- 6、设计题：设计广告索引引擎，几百万的广告数据，如何做到几十毫秒左右检索、返回广告；
- 7、离职的原因；
- 8、对下一份工作的预期；
- 9、算法：53. 最大子数组和 <<https://leetcode.cn/problems/maximum-subarray/>> ；
- 10、算法：字符串相减 (415. 字符串相加 <<https://leetcode.cn/problems/add-strings/>> 入参相同) 。

二面

倾向于方法论相关的问题：

- 1、工作期间的成长；
- 2、服务可用性的监控有哪些，监控体系是怎么考虑的；
- 3、负责的服务在整个工程中的角色，重要性如何体现；
- 4、针对服务出问题的情况，值班机制是什么样的；
- 5、系统哪部分查的问题最多，解决方案是什么；
- 6、出现问题多的原因是什么；
- 7、未来的职业规划；
- 8、沟通能力的体现；

9、算法：54. 螺旋矩阵 <<https://leetcode.cn/problems/spiral-matrix/>> 。

参考答案

你可以在下面两份参考资料上面大部分问题的参考答案：

- JavaGuide 在线阅读版： <https://javaguide.cn/> <<https://javaguide.cn/>> 。
- 《Java 面试指北》<https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247533385&idx=1&sn=105fdab847e4cf93d2603e57461658be&chksm=cea10c82f9d685944dc09010e5354566f8a8a40adce7cdcd3049444f7f83d4d084c0ef072542&token=766741944&lang=zh_CN&scene=21#wechat_redirect> 。

经验分享

心态

求职过程中最重要的就是心态，包括准备阶段和面试过程。我在去年三月份的时候被字节 hr 热情约面，于是想着多投几家试试水，结果发现最多到二面就结束了，很多问题都不会。现在想想很正常，因为当时 JavaGuide 的面试指南都没有看完，leetcode 刷了不到 50 题，但是当时觉得很郁闷，总有自己太菜的想法冒出来。知道要学哪些知识，但是知识太多了，加上看到别人的面经，投的公司一样，人家都拿到 offer 了，就觉得差距太大了，导致心态总是很崩溃。当时的化解方法是先去旅游放松了一下，然后制订了近三个月的学习计划，中间穿插着看放松心态的书，比如《蛤蟆先生去看心理医生》，看书加上实际的学习，让心态真的平静下来。

然后是求职过程中的心态，我都是使用 BOSS 直聘投递简历的，和中小公司 hr 沟通的过程中，想到最多的一句话就是“夏虫不可语冰”，主要体现在问期望薪资，他们要么说太高了、要么说现在的行情不可能给到这么高，还有说我现在薪资就挺高的，建议多待一段时间 😊。后面遇到大厂的 hr、猎头、主管，都是直言期望薪资正常，所以在和 hr 沟通的过程中也要稳住心态，坚信自己的想法。

最后是面试过程中，中小厂的一些面试官喜欢你简短介绍项目，等你介绍完了，又不断的暗示或明说你的项目简单，这个时候也不要觉得自己项目真的很差，因为只是一些复杂的细节你没有介绍出来，或者面试官故意制造的压力测试，再或者就是面试官的素质、水平原因，因为我确实没有遇到大厂面试官说过这种话。

最关键的结果面试没有通过，千万不要气馁，总结下面试经验，接着投下一家就好了，只要坚持找，一定能找到工作。

简历

不管校招还是社招，简历都是很重要的敲门砖，像我在试水阶段就被面试官说过看我的简历不知道我是什么岗位。后面我看了很多简历指导课，跟着视频优化简历，最后发给 Guide 和前同事，根据反馈再修改。最后，都是 hr 主动要简历，并且没有简历筛选不过的。

算法

现在面试过程中算法的重要性不言而喻，在试水后的几个月，我都坚持在 leetcode 上刷题，一共刷了 600 多道题。不过不要盲目的刷，最好是一组题型一刷，从认识技巧的简单题到认清技巧变种的中等题，做到刷完记住这个技巧。最好是选个大佬，跟着题单刷，我选择的是左程云--左神，跟着他的《程序员代码面试指南：IT 名企算法与数据结构题目最优解》这本书。让我很意外的一点是，我在看书的过程中遇到了问题，在知乎上私信他后，得到了详细的回答，就凭这一点，我就大力推荐左神！如果觉得看书思路比较难以理解，可以看看左神的视频，有些弹幕会问“为什么使用这个技巧”类似的问题，左神会说说他思考推进的过程，对于理解算法很有意义。

技巧

一是拿小公司练手，太久没有面试，表达能力和描述项目可以通过真实面试过程来精进，这时就可以拿中小公司或者没有意愿的公司来面试练手。

二是面试录音，所有面试全程录音，下来复盘问题，听听自己的回答是否流程、重点有没有说出，以及不会的问题再学习寻找答案。我觉得面试就像是戴着耳机唱歌，只有唱完听录音才自己唱的怎么样。