

基础篇

参考这里的面试题：面试题写在后面了

能回答上百分之七十，基础的广度就算 OK 了。如果达不到，那么缺什么就赶紧补什么。广度达到了，还需要对个别热点问题有深度。每个人的精力都有限，可以适当挑选两个热点问题进行深入挖掘。比如 HashMap，要不仅仅能够说上来源码是什么去实现的，还要了解 JDK1.7 版本到 1.8 版本都做了哪些改进，为什么这么改进，ConcurrentHashMap 怎么实现的，针对并发场景都做了哪些优化措施，JDK1.7 和 1.8 的实现有什么差异。了解这些可能需要花费很大精力。但是，面试的时候一旦问到了这个你非常精通的知识点，就基本进入你的主场了。这个能起到一锤定音的作用，给面试官留下极强的印象，甚至直接决定了此次面试的成败。因此，至少准备那么一两个非常精通的知识点，一定不要多，但要比面试官和其他求职者都要精通。这个赌的就是运气，只要在一个问题上能给面试官上一课，就能让对方产生“你很厉害”的错觉。

所有基础知识都要做好笔记，面试前多回顾下，在配合刷下题，问题就不大了。

项目篇

要回答好项目问题，核心就是要提前做好充足的准备。首先，仔细回顾自己做过的项目。然后挑选最有价值的部分，写一篇关于这个项目

的总结性文档，要讲清楚这个项目的架构，设计思路。其次是努力回想当时遇到的难点，踩过的坑，以及自己怎么思考和应对的，还有对项目后续改进的思考。这个非常重要，一定写下来。文档写出来后，还要反复提炼和补充。面试的时候就以此文档为纲领，方能做到侃侃而谈，从容应对。

情商篇

面试的时候，可能会穿插一些考察性格的问题。比如，问你为什么离职/想换工作，对技术和业务怎么看，对我们公司有什么了解，对我们的业务有什么了解，你有什么想问我的，你觉得自己有什么优点/缺点。做技术的同学往往在情商方面有所欠缺，如果没有准备的话，面试时一不小心说错一句话就可能导致前功尽弃。尤其在阿里，HR是有一票否决权的。那么，应对此类问题的第一点就是提前预防，可以提前拿张纸，猜想对方可能会问的问题，把标准答案写出来，反复推敲，然后背下来即可。比如，“你觉得自己有什么优点”这个问题，我当时的标准答案就是，“我觉得人最难了解的就是自己，所以我想借用我前领导对我的评价，XXX是个.....的人”。我的这个标准答案是基于“第三方的评价更有可信度”这个原理来设计的。第二点，回答问题的时候表现的谦虚一点即可，而且千万不要有负能量。比如“为什么离职/想换工作”，这个问题，不要去吐槽公司或者同事，或者表达情绪上的不满。要尽量展现出自己好学上进的一面，这一点很重要。

面试官会把对你的印象记录在案的，包括你的优点和缺点。因此即便技术达不到要求，也尽量展现性格上的优点。

阿里面试流程

面试一般是四到五面，以电话面试为主。最后一轮面试时 HR 面试，所以只要挺过前面的技术面试一般就 OK 了。第一轮是考察基础，第二轮开始考察项目，沟通表达能力。第三四轮会可能有编程，要求现场写代码，或者线下完成一个指定的项目。面试官的级别在 p6-p9 之间，一般随着面试流程的推进而升高。技术面试通过后就是 HR 面了，这一轮主要考察价值观。所有面试结束后，就是一个漫长的等待了，期间会有体检，背景调查，然后发 offer。这个流程会有一个月的时间，比较久。需要注意的是，面试过程中一定要保证诚实，不能有半点弄虚作假。因为一旦背景调查的时候发现说谎了，不但 offer 会取消，还会记录到人才库里。再有一点，约好了的面试不能随意爽约，这个也会永久记录。你的每次面试的过程都会详细的记录在人才库里的，因此尽量不要留下污点，以免影响以后的面试。

面试题

本人是做 [java 开发](#) 的，这是我参加 58，搜狐，搜狗，新浪微博，百度，腾讯文学，网易以及其他一些小的创业型公司的面试常被问的问题，当然有重复，弄清楚这些，相信面试会轻松许多。

1. junit 用法，before, beforeClass, after, afterClass 的执行顺序

2. 分布式锁
3. nginx 的请求转发算法，如何配置根据权重转发
4. 用 hashmap 实现 redis 有什么问题（死锁，死循环，可用 ConcurrentHashmap）
5. 线程的状态
5. 线程的阻塞的方式
6. sleep 和 wait 的区别
7. hashmap 的底层实现
8. 一万个人抢 100 个红包，如何实现（不用队列），如何保证 2 个人不能抢到同一个红包，可用分布式锁
9. java 内存模型，垃圾回收机制，不可达算法
10. 两个 Integer 的引用对象传给一个 swap 方法在方法内部交换引用，返回后，两个引用的值是否会发现变化
11. aop 的底层实现，动态代理是如何动态，假如有 100 个对象，如何动态的为这 100 个对象代理
12. 是否用过 maven install。 maven test。 git（make install 是安装本地 jar 包）
13. tomcat 的各种配置，如何配置 docBase
14. spring 的 bean 配置的几种方式
15. web.xml 的配置
16. spring 的监听器。
17. zookeeper 的实现机制，有缓存，如何存储注册服务的

18. IO 会阻塞吗? `readLine` 是不是阻塞的
19. 用过 `spring` 的线程池还是 `java` 的线程池?
20. 字符串的格式化方法 (20, 21 这两个问题问的太低级了)
21. 时间的格式化方法
22. 定时器用什么做的
23. 线程如何退出结束
24. `java` 有哪些锁? 乐观锁 悲观锁 `synchronized` 可重入锁 读写锁, 用过 `reentrantlock` 吗? `reentrantlock` 与 `synmchronized` 的区别
25. `ThreadLocal` 的使用场景
26. `java` 的内存模型, 垃圾回收机制
27. 为什么线程执行要调用 `start` 而不是直接 `run` (直接 `run`, 跟普通方法没什么区别, 先调 `start`, `run` 才会作为一个线程方法运行)
28. `qmq` 消息的实现机制(`qmq` 是去哪儿网自己封装的消息队列)
29. 遍历 `hashmap` 的三种方式
30. `jvm` 的一些命令
31. `memcache` 和 `redis` 的区别
32. `mysql` 的行级锁加在哪个位置
33. `ConcurrentHashMap` 的锁是如何加的? 是不是分段越多越好
34. `myisam` 和 `innodb` 的区别 (`innodb` 是行级锁, `myisam` 是表级锁)
35. `mysql` 其他的性能优化方式
36. `linux` 系统日志在哪里看
37. 如何查看网络进程

38. 统计一个整数的二进制表示中 bit 为 1 的个数
39. jvm 内存模型, java 内存模型
40. 如何把 java 内存的数据全部 dump 出来
41. 如何手动触发全量回收垃圾, 如何立即触发垃圾回收
42. hashmap 如果只有一个写其他全读会出什么问题
43. git rebase
44. mongodb 和 hbase 的区别
45. 如何解决并发问题
46. volatile 的用途
47. java 线程池 (好像之前我的理解有问题)
48. mysql 的 binlog
49. 代理模式
50. mysql 是如何实现事务的
51. 读写分离何时强制要读主库, 读哪个从库是通过什么方式决定的, 从库的同步 mysql 用的什么方式
52. mysql 的存储引擎
53. mysql 的默认隔离级别, 其他隔离级别
54. 将一个链表反转 (用三个指针, 但是每次只发转一个)
55. spring Aop 的实现原理, 具体说说
56. 何时会内存泄漏, 内存泄漏会抛哪些异常
57. 是否用过 Autowire 注解
58. spring 的注入 bean 的方式

59. sql 语句各种条件的执行顺序, 如 select, where, order by, group by
60. select xx from xx where xx and xx order by xx limit xx; 如何优化这个 (看 explain)
61. 四则元算写代码
62. 统计 100G 的 ip 文件中出现 ip 次数最多的 100 个 ip
63. zookeeper 的事物, 结点, 服务提供方挂了如何告知消费方
64. 5 台服务器如何选出 leader(选举算法)
65. 适配器和代理模式的区别
66. 读写锁
67. static 加锁
68. 事务隔离级别
69. 门面模式, 类图(外观模式)
70. mybatis 如何映射表结构
71. 二叉树遍历
72. 主从复制
73. mysql 引擎区别
74. 静态内部类加载到了哪个区? 方法区
75. class 文件编译后加载到了哪
76. web 的 http 请求如何整体响应时间变长导致处理的请求数变少, 该如何处理? 用队列, 当处理不了那么多 http 请求时将请求放到队列中慢慢处理, web 如何实现队列

77. 线程安全的单例模式
78. 快速排序性能考虑
79. volatile 关键字用法
80. 求表的 size，或做数据统计可用什么存储引擎
81. 读多写少可用什么引擎
82. 假如要统计多个表应该用什么引擎
83. concurrenhashmap 求 size 是如何加锁的，如果刚求完一段后这段发生了变化该如何处理
84. 1000 个苹果放 10 个篮子，怎么放，能让我拿到所有可能的个数
85. 可重入的读写锁，可重入是如何实现的？
86. 是否用过 NIO
87. java 的 concurrent 包用过没
88. sting s=new string("abc")分别在堆栈上新建了哪些对象
89. java 虚拟机的区域分配，各区分别存什么
90. 分布式事务（JTA）
91. threadlocal 使用时注意的问题（ThreadLocal 和 Synchronized 都用于解决多线程并发访问。但是 ThreadLocal 与 synchronized 有本质的区别。synchronized 是利用锁的机制，使变量或代码块在某一时该只能被一个线程访问。而 ThreadLocal 为每一个线程都提供了变量的副本，使得每个线程在某一时间访问到的并不是同一个对象，这样就隔离了多个线程对数据的数据共享。而 Synchronized 却正好相反，它用于在多个线程间通信时能够获得数据共享）

92. java 有哪些容器(集合，tomcat 也是一种容器)
93. 二分查找算法
94. myisam 的优点，和 innodb 的区别
95. redis 能存哪些类型
96. http 协议格式，get 和 post 的区别
97. 可重入锁中对应的 wait 和 notify
98. redis 能把内存空间交换进磁盘中吗(这个应该是可以的，但是那个面试官非跟我说不可以)
99. java 线程池中基于缓存和基于定长的两种线程池，当请求太多时分别是如何处理的？定长的事用的队列，如果队列也满了呢？交换进磁盘？基于缓存的线程池解决方法呢？
100. synchronized 加在方法上用的什么锁
101. 可重入锁中的 lock 和 trylock 的区别
102. innodb 对一行数据的读会枷锁吗？不枷锁，读实际读的是副本
103. redis 做缓存是分布式存的？不同的服务器上存的数据是否重复？guava cache 呢？是否重复？不同的机器存的数据不同
104. 用 awk 统计一个 ip 文件中 top10
105. 对表做统计时可直接看 schema info 信息，即查看表的系统信息
106. mysql 目前用的版本
107. 公司经验丰富的人给了什么帮助？(一般 boss 面会问这些)
108. 自己相对于一样的应届生有什么优势

109. 自己的好的总结习惯给自己今后的工作带了什么帮助，举例为证

110. 原子类，线程安全的对象，异常的处理方式

111. 4 亿个 int 数，如何找出重复的数（用 hash 方法，建一个 2 的 32 次方个 bit 的 hash 数组，每取一个 int 数，可 hash 下 2 的 32 次方找到它在 hash 数组中的位置，然后将 bit 置 1 表示已存在）

112. 4 亿个 url，找出其中重复的（考虑内存不够，通过 hash 算法，将 url 分配到 1000 个文件中，不同的文件间肯定就不会重复了，再分别找出重复的）

有 1 万个数组，每个数组有 1000 个整数，每个数组都是降序的，从中找出最大的 N 个数， $N < 1000$

113. LinkedHashMap 的底层实现

114. 类序列化时类的版本号的用途，如果没有指定一个版本号，系统是怎么处理的？如果加了字段会怎么样？

115. Override 和 Overload 的区别，分别用在什么场景

116. java 的反射是如何实现的