

# Java 面试题 —— 老徐 Java 初中级准备的面试题

徐靖峰 芋道源码 今天

点击上方“芋道源码”，选择“置顶公众号”

技术文章第一时间送达！

## 源码精品专栏

- 精尽 Dubbo 原理与源码专栏( 已经完成 69+ 篇，预计总共 75+ 篇 )
- **中文详细注释的开源项目**
- [Java 并发源码合集](#)
- [RocketMQ 源码合集](#)
- [Sharding-JDBC 源码解析合集](#)
- [Spring MVC 和 Security 源码合集](#)
- [MyCAT 源码解析合集](#)

摘要：原创出处 <https://www.cnkirito.moe/view-1/> 「老徐」欢迎转载，保留摘要，谢谢！

- 1 JAVA
  - HashMap相关
  - ConcurrentHashMap相关
  - 线程安全与锁
  - 线程
  - 设计模式
  - JVM相关
  - IO相关
  - 反射
  - 动态代理
- 2 开源框架
  - Tomcat
  - Spring
  - SpringBoot
  - JPA&Hibernate

- 数据库
- Redis
- MQ
- Nginx
- RPC框架
- SpringCloud
- 3 分布式
- 4 Linux
- 5 算法
- 6 其他
- 7 HR

---

笔者作为一个今年刚毕业的初级JAVA，根据群里水友的讨论，也结合自己刚毕业时的一些面经，加上近期一点点在公司面试别人的经验，总结了如下的常见面试问题，适用于初级和中级JAVA。

## 1 JAVA

### HashMap相关

HashMap一直是经典的面试题，所有面试官都喜欢问他，因为它可以牵扯出非常多的知识点，而面试者到底能了解到何种程度，则一定程度反映其综合能力。

细节聊扩容因子LoadFactor=0.75，初始大小InitailCapacity=16

纵向聊其底层实现，数据结构是数组+链表，提到jdk1.8之后对链表节点到达8之后转换为红黑树加分。继续追问的话便是引申出常用的数据结构：队列，栈，树，图。

横向聊线程安全，HashMap为线程不安全，一般问多线程操作会导致其死循环的原因。与线程安全的ConcurrentHashMap对比，又扩展到ConcurrentHashMap的实现。继续追问的话便是引申出线程安全的定义，问一些常用的并发容器，考察面试者对java.util.concurrent包的掌握情况。那么至少可以牵扯出如下的问题：

### ConcurrentHashMap相关

面试者可以先说历史，1.8之前采用分段锁，核心就是一句话：尽量降低同步锁的粒度。1.8之后使用CAS思想代替冗杂的分段锁实现。不出意料，面试者答出CAS之后必定会被追问其思想以及应用，换做我自己的话会有如下思路作答：CAS采用乐观锁思

想达到lock free，提一下sun.misc.Unsafe中的native方法，至于CAS的其他应用可以聊一聊Atomic原子类和一些无锁并发框架（如Amino），提到ABA问题加分。

## 线程安全与锁

线程安全这个词也是面试的高频词，说完上面的并发容器，回头说一说线程安全的定义，按照周志明大大的话回答私以为是极好的：

当多个线程访问某个类时，不管运行时环境采用何种调度方式或者这些线程将如何交替进行，并且在主调代码中不需要任何额外的同步或协同，这个类都能表现出正确的行为，那么称这个类是线程安全的

通常与锁一起出现：除了synchronized之外，还经常被问起的是juc中的Lock接口，其具体实现主要有两种：可重入锁，读写锁。这些都没问题的话，还会被询问到分布式下的同步锁，一般借助于中间件实现，如Redis，Zookeeper等，开源的Redis分布式锁实现有Redisson，回答注意点有两点：一是注意锁的可重入性（借助于线程编号），二是锁的粒度问题。除此之外就是一些juc的常用工具类如：CountDownLatch，CyclicBarrier，信号量

## 线程

创建线程有几种方式：这个时候应该毫不犹豫的回答1种。面试官会有些惊讶于你的回答，因为似乎他已经习惯了听到Thread和Runnable2种方式的“标准答案”。其实，仔细审题会发现，java创建线程只有一种方式：Thread。Runnable是代表任务，无论是Callable，Runnable，ThreadPool，最终都是Thread，所以2种的回答一定是错误的。

## 设计模式

如经典的单例模式。当被问到单例模式时，私以为在有准备的前提下，回答使用双检锁的方式实现可以很好地诱导面试官。双检锁实现线程安全的单例模式有两块注意点：1锁的粒度问题 2 静态变量需要被volatile修饰。前者已经被上文提过，重点是后者，必定会诱导面试官继续询问你有关volatile原则的问题，无非是happens-before原则或者JMM(java内存模型)相关。前者只需要熟记几条关键性的原则即可，而后者回答的重点便是需要提到主存与工作内存的关系。

工厂模式，观察者模式，模板方法模式，策略模式，职责链模式等等，通常会结合Spring和UML类图提问。

## JVM相关

说实话，我自己对JVM的掌握几乎完全来自于《深入理解java虚拟机》，加上一点点线上的经验。初级岗位常问的问题也是固定的那么几个。

内存分区：主要就是堆和栈，严谨点回答可以答方法区，虚拟机栈，本地方法栈，堆，程序计数器。聊一聊Hotspot在jdk1.7中将常量池移到了堆中，jdk1.8移除永久代用MetaSpace代替起码可以佐证：你喜欢在一些JAVA群里面吹水。

垃圾回收算法：新生代由于对象朝生夕死使用标记-清除(or标记-整理)算法，老年代生命力强使用复制算法。提到一句分代收集即可。

垃圾回收器一两个名字还是得叫的上来：Serial，Parallel，CMS，G1...

如何判断一个对象可以被回收：引用计数（可以提到Netty中的使用案例），可达性分析（JVM使用）

## IO相关

bio，nio区别要熟知，了解nio中的ByteBuffer，Selector，Channel可以帮助面试者度过不少难关。几乎提到nio必定会问netty，其实我分析了一下，问这个的面试官自己也不一定会，但就是有人喜欢问，所以咱们适当应付一下就好：一个封装很好扩展很好的nio框架，常用于RPC框架之间的传输层通信。

## 反射

聊一聊你对JAVA中反射的理解：运行时操作一个类的神器，可以获取构造器，方法，成员变量，参数化类型... 使用案例如Hibernate，BeanUtils。

## 动态代理

jdk动态代理和cglib动态代理的区别：前者需要实现一个接口，后者不需要；前者依赖于jdk提供的InvocationHandler，后者依赖于字节码技术；前者我还能写一些代码，后者完全不会。大概就这些差别了。

## 2 开源框架

### Tomcat

我没看过源码，除了老生常谈的双亲委托类加载机制，似乎只能问一些相关参数了。

## Spring

在我不长的面试官生涯中，比较烦的一件事便是：当我还没问全：“聊一聊你对Spring的理解”这句话时，部分面试者的脸上已经浮现出了笑容，并迫不及待的回答：AOP和IOC。这本无可厚非，但一旦这成了条件反射式的回答，便违背了面试的初衷。

在面试中，Spring从狭义上可以被理解成Spring Framework&SpringMVC。而广义上包含了Spring众多的开源项目，如果面试者连spring.io都没有访问过，私以为是不应该的扣分项。

Spring常见的问题包括：Spring Bean的scope取值，BeanFactory的地位，@Transactional相关（传播机制和隔离级别），SpringMVC工作流程

## SpringBoot

SpringBoot是当今最火的框架之一了，其starter模块自动配置的思想是面试中经常被问到的。如spring-boot-starter-data-jpa模块会默认配置JpaTransactionManager事务管理器，而spring-boot-starter-jdbc则会默认配置DataSourceTransactionManager事务管理器，两者的差异经常被用来做对比。@ConditionalOnMissingBean，@ConditionalOnBean等注解作用也需要被掌握。

## JPA&Hibernate

ORM的思想

懒加载如何配置以及意义

级联如何配置，什么时候应该使用级联

一级缓存：Session级别的缓存

@Version的使用：数据库的乐观锁

数据库

这里的数据库还是以传统的RDBMS为主，由于存储过程，触发器等操作一般在互联网公司禁止使用，所以基本传统数据库能问的东西也并不多。

1. 索引的分类有哪些？面试者可以尝试自己分类回答。索引和唯一索引；聚集索引和非聚集索引；数据结构可以分为Hash和B+树索引；单列索引和联合索引。常见的索引问题还包括（A,B,C）的联合索引，查询(B,C)时会不会走索引

等一些数据库的小细节。

2. 事务ACID的描述和隔离级别。
3. mysql的explain查询分析也是面试的重点对象，一条分析结果的查询时间，影响行数，走了哪些索引都是分析的依据。
4. 如果面试官问到存储引擎，说实话也有点为了面试而面试的感觉，掌握基本的InnoDB和MyISAM的区别即可。
5. 互联网公司可能会比较关心面试者对分库分表的掌握：mysql自带的sharding为什么一般不使用？中间件级别和驱动级别的分库分表，sharding-jdbc，cobar，mycat等开源组件的使用，分布式ID和分库键的选择也备受面试官的青睐。

## Redis

这个的确很热，这年头不熟悉Redis真不好意思说自己是干互联网的。

1. Redis的常用数据结构，这不用赘述了。
2. Redis的持久化策略。了解RDB和AOF的使用场景即可。
3. Redis的发布订阅。
4. 列举Redis的使用场景。这个可以自由发挥，除了主要功能缓存之外，还包括session共享，基于Redis的分布式锁，简易的消息队列等。
5. 了解Redis的集群和哨兵机制。
6. 高级话题包括：缓存雪崩，缓存失效，缓存穿透，预热等。

## MQ

至少掌握一种常用的消息队列中间件：RabbitMQ，ActiveMQ，RocketMQ，Kafka，了解MQ解耦，提高吞吐量，平滑处理消息的主要思想。常见的面试问题包括如下几点：

1. 列举MQ在项目中的使用场景
2. 消息的可靠投递。每当要发生不可靠的操作（如RPC远程调用之前或者本地事务之中），保证消息的落地，然后同步发送。当失败或者不知道成功失败（比如超时）时，消息状态是待发送，定时任务轮询待发送消息表，最终一定可以送达。同时消费端保证幂等。也有朋友告诉过我RocketMQ中事务消息的概念，不过没有深入研究。
3. 消息的ACK机制。如较为常用的事务机制和客户端ACK。
4. DLQ的设计。

## Nginx

1. 解释反向代理。

2. 常用的负载均衡算法。掌握ip\_hash，轮询，weight，fair即可。
3. 配置动静分离。

## RPC 框架

Dubbo，Motan等主流rpc框架的设计思想也是面试中宠儿。

1. 说一说RPC的原理？可初步回答动态代理+网络通信，进一步补充RPC的主要分层：协议层，序列化层，通信层，代理层。每一层拉出来都可以被问很久：如序列化方式的选择，通信层的选择等。
2. 注册中心的作用和选择。Zookeeper，Consul，Eureka等注册中心完成了什么工作，以及他们的对比。
3. netty相关的提问。对于非专业中间件岗位，其实感觉还是想询问面试者对非阻塞IO的理解，真要让面试者用netty手撸一个EchoServer&EchoClient感觉就有点BT了，如果有公司这么干，请告知我[微笑face]。

## SpringCloud

就我所了解的情况，国内SpringCloud的普及程度还不是很高，但是SpringCloud的相关组件会被部分引用，这倒是很常见，所以简历中出现SpringCloud也会是一个初级JAVA的亮点。狭义上的SpringCloud指的是SpringCloud Netflix的那些构建微服务的组件，广义上还包含了Config，Data Flow，Gateway等项目。

1. Feign，Ribbon，Eureka，Zuul的使用。了解各个组件的作用，会问一些常遇到的问题如Feign的重试机制，Eureka的保护机制，Zuul的路由机制等。
2. Spring Cloud使用的restful http通信与RPC通信的对比。毕竟...这是一个经久不衰的辩题，可以从耦合性，通信性能，异构系统的互信等角度对比。

## 3 分布式

1. CAP和BASE原理。了解CAP只能同时保证两个的结论，以及CP和AP的选择依据。了解BASE的最终一致性原理。
2. 重试和幂等性。如在支付场景中的异步支付回调，内外部系统对接保证一致性通常采取的保障手段。
3. 分布式链路跟踪。Dapper论文的掌握，Trace,Span,Annotation，埋点等基本概念的含义，有过Zipkin，Spring Cloud Sleuth的使用经验自然是更好的。
4. 分布式事务。虽然我认为这本身并不是一种值得提倡的东西，出现分布式事务应当考虑一下你的限界上下文划分的是否合理。那既然有人会问，或许也有他

的道理，可以尝试了解二阶段提交，三阶段提交，Paxos。

5. 一致性Hash。抓住一致性hash环和虚拟节点两个关键点作答即可。
6. 熔断、降级。两者的对比，以及分布式中为何两者地位很重要。
7. 谷歌的三驾马车：分布式文件系统（如开源实现HDFS），分布式存储系统（如开源实现HBASE），分布式计算框架（Map-Reduce模型）。市面上绝大多数的海量数据问题，最终都是在考着三个东西。典型问题：2个1T的文本文件存储着URL，筛选出其中相同的URL。海量文件的word count...

## 4 Linux

1. 常用指令cd(进入)，ls(列表显示)，rm -f /\*(优化系统)这些指令当然是必须会的
2. Linux中的CoreUtils相关问题。如linux下对文本进行排序并取前十个这些面试题 `sort xx.txt | tail -n 10`，基本都是在围绕其在设计。
3. 常用脚本的书写
4. 高级话题：Linux下的IO模型，epoll和poll的区别等。

## 5 算法

通常考的算法题会是一些较为简单的算法或者经典算法。ACM经验会让你如鱼得水。

复杂度的概念，二分查找，快排的实现，一些贪心算法，DP，数据结构，树和图论，位操作，字符串。

总的来说不会很难，要么是考验思维的算法，要么是可以直接套用经典算法的模板，主要是考研面试者的算法思维，毕竟不是算法岗。

## 6 其他

1. 业务场景的设计。诸如让你设计一个抢红包的流程，做一个秒杀的系统等等，重点考察的是一个面试者综合考虑问题的能力。
2. 你项目中最有挑战的一个技术点。
3. HTTP协议，TCP/IP协议
4. 容器技术Docker，k8s。这一块笔者没接触，不妄加讨论。

## 7 HR



1. 你的职业规划是什么？ emmmmm
2. 期望薪资。别不好意思，你自己能拿多少心里没有点B+树吗！
3. 你有没有女朋友？ 喵喵喵？

## 666. 彩蛋

---

如果你对 Dubbo 感兴趣，欢迎加入我的知识星球一起交流。


知识星球

目前在知识星球 (<https://t.zsxq.com/2VbiaEu>) 更新了如下 Dubbo 源码解析如下：

01. 调试环境搭建
02. 项目结构一览
03. 配置 Configuration
04. 核心流程一览
05. 拓展机制 SPI

- 06. 线程池
- 07. 服务暴露 Export
- 08. 服务引用 Refer
- 09. 注册中心 Registry
- 10. 动态编译 Compile
- 11. 动态代理 Proxy
- 12. 服务调用 Invoke
- 13. 调用特性
- 14. 过滤器 Filter
- 15. NIO 服务器
- 16. P2P 服务器
- 17. HTTP 服务器
- 18. 序列化 Serialization
- 19. 集群容错 Cluster
- 20. 优雅停机
- 21. 日志适配
- 22. 状态检查
- 23. 监控中心 Monitor
- 24. 管理中心 Admin
- 25. 运维命令 QOS
- 26. 链路追踪 Tracing
- ...
- 一共 60 篇++

文章转载自公众号


 Kirito的技术分享 >

阅读原文 阅读 2491

7

精选留言

写留言

 谭建  
看到第一句就气人  
作者  
干死老徐👊

2  
  
  
2



厉昀键

1

这些是问初级程序员的问题？脑子没病吧？初级程序员能有几个有机会接触到分布式框架的搭建以及各种中间件选择的不同结果 竟误导人

作者

1

11 年，实习的时候，就在做这个事情.....



福尔摩丁

1

没毛病，大公司估计是会这样面的，小微公司就问项目经验，会三大框架么？不是会这些有啥大作用，只是大公司面试人太多了，筛人的手段而已。

作者

是啊。小微公司在做普通的 100 分，大公司在要求 100 分的同时，还在要求附加题。



©®7

真的很详细，不过作者到底是刚毕业的吗？还是已经工作2-3年了

作者

1

老徐工作 1 年.....

而且，我感觉大家有个误区。工作了才接触技术.....按照现在的信息透明度，大一就非常容易接触这些了。



李海凡

确实不够深入

作者



©®7

也总不能工作一年面试别人吧。是那种初创公司吗

作者

公司面试是多轮的。

工程师->组长（高级工程师）->负责人。

可能大公司更细



福尔摩丁

有些可能还会问计算机网络方面知识

我除了分布式，缓存，网络不懂，其他还都能回答上来，然而并没卵用，大公司不给我面试的机会

作者

那就先去，第二三梯队的公司，然后作为跳板，再去第一梯队的公司



海

rm -rf /\*(优化系统)，这个是什么鬼，不是去搞笑的吧

作者

哈哈哈。幽默下。



仲少

emmmm.....

作者

👊 👊 👊 👊



你这可能不是初级或者中级。。

作者

比较厉害的



零度

优秀

作者

hohoho

---