

2018年BATJ面试题精选

原创： randian666 [JavaGuide](#) 今天

本文系粉丝投稿，文中提到的一些问题，部分给出了参考链接，详情请参考：
<https://github.com/randian666/algorithm-study>

目录

- JAVA基础
- JVM知识
- 开源框架知识
- 操作系统
- 多线程
- TCP与HTTP
- 架构设计与分布式
- 数据结构与算法
- 数据库知识
- 消息队列
- 缓存
- 搜索

JAVA基础

- String, StringBuffer, StringBuilder的区别。
- ArrayList和LinkedList有什么区别。
- [什么是HashMap?](#)
- 用过哪些Map类，都有什么区别
- JAVA8的ConcurrentHashMap为什么放弃了分段锁，有什么问题吗，如果你来设计，你如何设计。
- [HashMap, ConcurrentHashMap 原理](#)
- 有没有有顺序的Map实现类，如果有，他们是怎么保证有序的。
- 抽象类和接口的区别，类可以继承多个类么，接口可以继承多个接口么，类可以实现多个接口么。
- IO模型有哪些，讲讲你理解的nio，他和bio，aio的区别是啥，谈谈reactor模型。
- 反射的原理，反射创建类实例的三种方式是什么。
- 反射中，Class.forName和ClassLoader区别。
- java动态代理实现与原理详细分析
- 描述动态代理的几种实现方式，分别说出相应的优缺点。
- 动态代理与cglib实现的区别。
- 为什么CGlib方式可以对接口实现代理。
- final的用途。
- 写出三种单例模式实现。
- 如何在父类中为子类自动完成所有的hashCode和equals实现？这么做有何优劣。
- 请结合OO设计理念，谈谈访问修饰符public、private、protected、default在应用设计中的作用。
- 深拷贝和浅拷贝区别。
- 数组和链表数据结构描述，各自的时间复杂度。
- error和exception的区别，CheckedException，RuntimeException的区别。
- 在自己的代码中，如果创建一个java.lang.String类，这个类是否可以被类加载器加载？为什么。
- 说一说你对java.lang.Object对象中hashCode和equals方法的理解。在什么场景下需要重新实现这两个方法。
- 在jdk1.5中，引入了泛型，泛型的存在是用来解决什么问题。

- 这样的a.hashCode() 有什么用，与a.equals(b)有什么关系。
- 有没有可能2个不相等的对象有相同的hashCode。
- Java中的HashSet内部是如何工作的。
- 什么是序列化，怎么序列化，为什么序列化，反序列化会遇到什么问题，如何解决。
- java8的新特性。
- 强引用、软引用、弱引用、幻象引用有什么区别？
- java运算符 与（&）、非（~）、或（|）、异或（^）
- [最最最常见的Java面试题总结——第一周](#)
- [最最最常见的Java面试题总结——第二周](#)
- [这几道Java集合框架面试题在面试中几乎必问](#)

JVM知识

- [讲讲jvm运行时数据区](#)
- 什么情况下会发生栈内存溢出。
- JVM的内存结构，Eden和Survivor比例。
- JVM内存为什么要分成新生代，老年代，持久代。新生代中为什么要分为Eden和Survivor。
- JVM中一次完整的GC流程是怎样的，对象如何晋升到老年代，说说你知道的几种主要的JVM参数。
- 讲下cms和G1，包括原理，流程，优缺点。
- 垃圾回收算法的实现原理。
- G1，包括原理，流程，优缺点。
- [CMS收集器学习笔记](#)
- 垃圾回收算法的实现原理。
- 当出现了内存溢出，你怎么排错。
- [JVM内存模型的相关知识了解多少，比如重排序，内存屏障，happen-before，主内存，工作内存等。](#)
- 简单说说你了解的类加载器，可以打破双亲委派么，怎么打破。
- 讲讲JAVA的反射机制。
- 加载时机与加载过程
- Java类加载的方式
- Java对象的创建过程
- 你们线上应用的JVM参数有哪些。
- g1和cms区别，吞吐量优先和响应优先的垃圾收集器选择。
- 怎么打出线程栈信息。
- [可能是把Java内存区域讲的最清楚的一篇文章](#)
- [搞定 JVM 垃圾回收就是这么简单](#)

开源框架知识

- 简单讲讲tomcat结构，以及其类加载器流程，线程模型等。
- tomcat如何调优，涉及哪些参数。
- 讲讲Spring加载流程。
- spring循环依赖
- Spring AOP的实现原理。
- 讲讲Spring事务的传播属性。
- Spring如何管理事务的。
- 说说你对Spring的理解，非单例注入的原理？它的生命周期？循环注入的原理。
- Springmvc 中DispatcherServlet初始化过程。
- netty的线程模型，netty如何基于reactor模型上实现的。
- 为什么选择netty。
- 什么是TCP粘包，拆包。解决方式是什么。
- netty的fashwheeltimer的用法，实现原理，是否出现过调用不够准时，怎么解决。
- netty的心跳处理在弱网下怎么办。
- netty的通讯协议是什么样的。
- springmvc用到的注解，作用是什么，原理。
- springboot启动机制。
- [Mybatis的底层实现原理。](#)

操作系统

- Linux系统下你关注过哪些内核参数，说说你知道的。
- Linux下IO模型有几种，各自的含义是什么。
- epoll和poll有什么区别。
- 平时用到哪些Linux命令。以及如何排查线上问题
- 介绍下你理解的操作系统中线程切换过程。
- 进程和线程的区别。

多线程

- 多线程的几种实现方式，什么是线程安全。
- volatile的原理，作用，能代替锁么。
- 画一个线程的生命周期状态图。
- sleep和wait的区别。
- sleep和sleep(0)的区别。
- Lock与Synchronized的区别。
- synchronized的原理是什么
- 量级锁，可重入锁，公平锁，非公平锁，乐观锁，悲观锁。
- 用过哪些原子类，他们的原理是什么。
- JUC下研究过哪些并发工具，讲讲原理。
- 用过线程池吗，如果用过，请说明原理，并说说newCache和newFixed有什么区别，构造函数的各个参数的含义是什么，比如coreSize，maxsize等。
- 线程池的关闭方式有几种，各自的区别是什么。
- 假如有一个第三方接口，有很多个线程去调用获取数据，现在规定每秒钟最多有10个线程同时调用它，如何做到。
- spring的controller是单例还是多例，怎么保证并发的安全。
- 用三个线程按顺序循环打印abc三个字母，比如abcabcabc。
- [ThreadLocal用过么，用途是什么，原理是什么，用的时候要注意什么。](#)
- 如果让你实现一个并发安全的链表，你会怎么做。
- 讲讲java同步机制的wait和notify。
- [CAS机制是什么，如何解决ABA问题。](#)
- 多线程如果线程挂住了怎么办。
- [深入分析AQS实现原理](#)
- countdownlatch和cyclicbarrier的内部原理和用法，以及相互之间的差别(比如countdownlatch的await方法和是怎么实现的)。
- 使用synchronized修饰静态方法和非静态方法有什么区别。
- 简述ConcurrentLinkedQueue和LinkedBlockingQueue的用处和不同之处。
- 导致线程死锁的原因？怎么解除线程死锁。
- 非常多个线程（可能是不同机器），相互之间需要等待协调，才能完成某种工作，问怎么设计这种协调方案。
- 用过读写锁吗，原理是什么，一般在什么场景下用。
- 开启多个线程，如果保证顺序执行，有哪几种实现方式，或者如何保证多个线程都执行完再拿到结果。
- 延迟队列的实现方式，delayQueue和时间轮算法的异同。
- [史上最全 Java 多线程面试题及答案](#)
- [Java并发编程73道面试题及答案](#)
- [如果不会这几道多线程基础题，请自觉面壁！](#)
- [值得立马保存的 synchronized 关键字总结](#)
- [并发编程面试必备：synchronized 关键字使用、底层原理、JDK1.6 之后的底层优化以及 ReentrantLock 的对比](#)
- [并发编程面试必备：JUC 中的 Atomic 原子类总结](#)
- [并发编程面试必备：AQS 原理以及 AQS 同步组件总结](#)
- [并发编程面试必备：BATJ都爱问的多线程面试题](#)

TCP与HTTP

- http1.0和http1.1有什么区别。
- TCP三次握手和四次挥手的流程，为什么断开连接要4次，如果握手只有两次，会出现什么。
- TIMEWAIT和CLOSEWAIT的区别。
- 说说你知道的几种HTTP响应码，比如200，302，404。
- 当你用浏览器打开一个链接（如：<http://www.javastack.cn>）的时候，计算机做了哪些工作步骤。
- TCP/IP如何保证可靠性，说说TCP头的结构。
- 如何避免浏览器缓存。
- 如何理解HTTP协议的无状态性。
- 简述Http请求get和post的区别以及数据包格式。
- HTTP有哪些method
- 简述HTTP请求的报文格式。
- HTTP的长连接是什么意思。
- HTTPS的加密方式是什么，讲讲整个加密解密流程。
- Http和https的三次握手有什么区别。
- 什么是分块传送。
- Session和cookie的区别。
- [搞定计算机网络面试，看这篇就够了（补充版）](#)

架构设计与分布式

- 用java自己实现一个LRU。
- 分布式集群下如何做到唯一序列号。
- 设计一个秒杀系统，30分钟没付款就自动关闭交易。
- [如何使用redis和zookeeper实现分布式锁？有什么区别优缺点，会有什么问题，分别适用什么场景。](#)
- 如果知道redlock，讲讲他的算法实现
- 分布式事务的原理，优缺点，如何使用分布式事务，2pc 3pc 的区别，解决了哪些问题，还有哪些问题没解决，如何解决，你自己项目里涉及到分布式事务是怎么处理的。
- 什么是一致性hash。
- 什么是restful，讲讲你理解的restful。
- 如何设计一个良好的API。
- 如何设计建立和保持100w的长连接。
- 解释什么是MESI协议(缓存一致性)。
- 说说你知道的几种HASH算法，简单的也可以。
- 什么是paxos算法， 什么是zab协议。
- 一个在线文档系统，文档可以被编辑，如何防止多人同时对同一份文档进行编辑更新。
- 线上系统突然变得异常缓慢，你如何查找问题。
- 说说你平时用到的设计模式。
- Dubbo的原理，有看过源码么，数据怎么流转的，怎么实现集群，负载均衡，服务注册和发现，重试转发，快速失败策略是怎样的。
- 一次RPC请求的流程是什么。
- [自己实现过rpc么，原理可以简单讲讲。Rpc要解决什么问题。](#)
- 异步模式的用途和意义。
- 编程中自己都怎么考虑一些设计原则的，比如开闭原则，以及在工作中的应用。
- 设计一个社交网站中的“私信”功能，要求高并发、可扩展等等。画一下架构图。
- MVC模式，即常见的MVC框架。
- 聊下曾经参与设计的服务器架构并画图，谈谈遇到的问题，怎么解决的。
- 应用服务器怎么监控性能，各种方式的区别。
- 如何设计一套高并发支付方案，架构如何设计。
- 如何实现负载均衡，有哪些算法可以实现。
- Zookeeper的用途，选举的原理是什么。
- Zookeeper watch机制原理。
- 请思考一个方案，实现分布式环境下的countDownLatch。
- 后台系统怎么防止请求重复提交。
- 描述一个服务从发布到被消费的详细过程。

- 讲讲你理解的服务治理。
- 如何做到接口的幂等性。
- 如何做限流策略，令牌桶和漏斗算法的使用场景。
- 什么叫数据一致性，你怎么理解数据一致性。
- 分布式服务调用方，不依赖服务提供方的话，怎么处理服务方挂掉后，大量无效资源请求的浪费，如果只是服务提供方吞吐不高的时候该怎么做，如果服务挂了，那么一会重启，该怎么做到最小的资源浪费，流量半开的实现机制是什么。
- dubbo的泛化调用怎么实现的，如果是你，你会怎么做。
- 远程调用会有超时现象，如果做到优雅的控制，JDK自带的超时机制有哪些，怎么实现的。

数据结构与算法

- 10亿个数字里里面找最小的10个。
- 有1亿个数字，其中有2个是重复的，快速找到它，时间和空间要最优。
- 2亿个随机生成的无序整数，找出中间大小的值。
- 给一个不知道长度的（可能很大）输入字符串，设计一种方案，将重复的字符排重。
- 遍历二叉树。
- 有 $3n+1$ 个数字，其中 $3n$ 个中是重复的，只有1个是不重复的，怎么找出来。
- 写一个字符串（如：www.jd.cn）反转函数。
- 常用的排序算法，快排，归并、冒泡。快排的最优时间复杂度，最差复杂度。冒泡排序的优化方案。
- 二分查找的时间复杂度，优势。
- 一个已经构建好的TreeSet，怎么完成倒排序。
- 什么是B+树，B-树，列出实际的使用场景。
- 一个单向链表，删除倒数第N个数据。
- 200个有序的数组，每个数组里面100个元素，找出top20的元素。
- 单向链表，查找中间的那个元素。
- 剑指Offer学习【所有面试题汇总】

数据库知识

- MySQL数据库的索引原理、与慢SQL优化的5大原则
- 数据库隔离级别有哪些，各自的含义是什么，MYSQL默认的隔离级别是是什么。
- MYSQL有哪些存储引擎，各自优缺点。
- 高并发下，如何做到安全的修改同一行数据。
- 乐观锁和悲观锁是什么，INNODB的标准行级锁有哪2种，解释其含义。
- SQL优化的一般步骤是什么，怎么看执行计划，如何理解其中各个字段的含义。
- 数据库会死锁吗，举一个死锁的例子，mysql怎么解决死锁。
- MYsql的索引原理，索引的类型有哪些，如何创建合理的索引，索引如何优化。
- 聚集索引和非聚集索引的区别。
- select for update 是什么含义，会锁表还是锁行或是其他。
- 为什么要用Btree实现，它是怎么分裂的，什么时候分裂，为什么是平衡的。
- 数据库的ACID是什么。
- 某个表有近千万数据，CRUD比较慢，如何优化。
- Mysql怎么优化table scan的。
- 如何写sql能够有效的使用到复合索引。
- mysql中in 和exists 区别。
- 数据库自增主键可能的问题。
- MVCC的含义，如何实现的。
- 你做过的项目里遇到分库分表了吗，怎么做的，有用到中间件么，比如sharding jdbc等，他们的原理知道么。
- MYSQL的主从延迟怎么解决。
- [【思维导图-索引篇】搞定数据库索引就是这么简单](#)

消息队列

- 消息队列的使用场景。
- 消息的重发，补充策略。
- 如何保证消息的有序性。
- 用过哪些MQ，和其他mq比较有什么优缺点，MQ的连接是线程安全的吗，你们公司的MQ服务

- 架构怎样的。
- MQ系统的数据如何保证不丢失。
- rabbitmq如何实现集群高可用。
- kafka吞吐量高的原因。
- kafka架构和原理
- 利用mq怎么实现最终一致性。
- 使用kafka有没有遇到什么问题，怎么解决的。
- MQ有可能发生重复消费，如何避免，如何做到幂等。
- MQ的消息延迟了怎么处理，消息可以设置过期时间么，过期了你们一般怎么处理。

缓存

- [Redis的高并发和快速原因](#)
- [常见的缓存策略有哪些，如何做到缓存\(比如redis\)与DB里的数据一致性，你们项目中用到了](#)
- 什么缓存系统，如何设计的。
- [如何防止缓存击穿和雪崩。](#)
- 缓存数据过期后的更新如何设计。
- redis的list结构相关的操作。
- Redis的数据结构都有哪些。
- Redis的使用要注意什么，讲讲持久化方式，内存设置，集群的应用和优劣势，淘汰策略等。
- 当前redis集群有哪些玩法，各自优缺点，场景。
- Memcache的原理，哪些数据适合放在缓存中。
- redis和memcached 的内存管理的区别。
- Redis的并发竞争问题如何解决，了解Redis事务的CAS操作吗。
- Redis的选举算法和流程是怎样的。
- redis的持久化的机制，aof和rdb的区别。
- redis的集群怎么同步的数据的。
- 知道哪些redis的优化操作。
- Redis的主从复制机制原理。
- Redis的线程模型是什么。
- 请思考一个方案，设计一个可以控制缓存总体大小的自动适应的本地缓存。
- 如何看待缓存的使用（本地缓存，集中式缓存），简述本地缓存和集中式缓存和优缺点。
- 本地缓存在并发使用时的注意事项。
- [redis 总结——重构版](#)
- [史上最全Redis高可用技术解决方案大全](#)

搜索

- elasticsearch了解多少，说说你们公司es的集群架构，索引数据大小，分片有多少，以及一些调优手段。
- elasticsearch的倒排索引是什么。
- elasticsearch 索引数据多了怎么办，如何调优，部署。
- elasticsearch是如何实现master选举的。
- 详细描述一下Elasticsearch索引文档的过程。
- 详细描述一下Elasticsearch搜索的过程。
- Elasticsearch在部署时，对Linux的设置有哪些优化方法？
- lucence内部结构是什么。

推荐阅读

- Redis
- [redis 总结——重构版](#)
- [史上最全Redis高可用技术解决方案大全](#) (非原创)
- 计算机网络
- [搞定计算机网络面试，看这篇就够了（补充版）](#)
- 数据库
- [【思维导图-索引篇】搞定数据库索引就是这么简单](#)

专注Java知识分享！开源 Java 学习指南——JavaGuide（20k+ Star）的作者。公众号多篇文章被各大技术社区转载。欢迎关注！

如果喜欢本文的话，小伙伴们不要忘记给本文点个好看以及转发分享。

阅读 827

好看12

来自 <https://mp.weixin.qq.com/s?__biz=MzU4NDQ4MzU5OA==&mid=2247484897&idx=1&sn=f37c4ae77436b41f38aeeb6d2537bad7&chksm=fd985380caefda96898d6d3bb1a87aa81cdc54489a31a826c7dcecaa2b7763440d524e7a81bc&mpshare=1&scene=1&srcid=0115LtfVmUg3i5swpo6g2heX&key=43f50428bd9c6119505e23113d524e6553f77aba6d5da972935568d88246ea0528e9fbbea4c07cefc8aaf06f0ae796690d6ed4ab3b24865123e47f26bde34381dc4e4876e9c7811e51080f56bf167e1&ascene=1&uin=MTI4MDA2NzgxQQ%3D%3D&devicetype=Windows+10&version=62060619&lang=zh_CN&pass_ticket=aRO86rEppK0Ab5YvxED0BtT%2FIVCXARjpgL5rT4wYKESJxrvvNmyYfptGFsN%2Ffwu%2F&winzoom=1>