

01 面向对象

→ 什么是面向对象

面向对象、面向过程

面向对象的三大基本特征和五大基本原则

→ 平台无关性

Java 如何实现的平台无关

JVM 还支持哪些语言（Kotlin、Groovy、JRuby、Jython、Scala）

→ 值传递

值传递、引用传递

为什么说 Java 中只有值传递

→ 封装、继承、多态

什么是多态、方法重写与重载

Java 的继承与实现

构造函数与默认构造函数

类变量、成员变量和局部变量

成员变量和方法作用域

02 Java 基础知识

→ 基本数据类型

8 种基本数据类型：整型、浮点型、布尔型、字符型

整型中 byte、short、int、long 的取值范围

什么是浮点型？什么是单精度和双精度？为什么不能用浮点型表示金额？

→ 自动拆装箱

什么是包装类型、什么是基本类型、什么是自动拆装箱

Integer 的缓存机制

→ String

字符串的不可变性

JDK 6 和 JDK 7 中 substring 的原理及区别、

replaceFirst、replaceAll、replace 区别、

String 对“+”的重载、字符串拼接的几种方式和区别

String.valueOf 和 Integer.toString 的区别、

switch 对 String 的支持

字符串池、常量池（运行时常量池、Class 常量池）、intern

→ 熟悉 Java 中各种关键字

transient、instanceof、final、static、volatile、synchronized、const 原理及用法

→ 集合类

常用集合类的使用、ArrayList 和 LinkedList 和 Vector 的区别、SynchronizedList 和 Vector 的区别、HashMap、HashTable、ConcurrentHashMap 区别、Set 和 List 区别？Set 如何保证元素不重复？Java 8 中 stream 相关用法、apache 集合处理工具类的使用、不同版本的 JDK 中 HashMap 的实现的区别以及原因 Collection 和 Collections 区别 Arrays.asList 获得的 List 使用时需要注意什么 Enumeration 和 Iterator 区别 fail-fast 和 fail-safe CopyOnWriteArrayList、ConcurrentSkipListMap

→ 枚举

枚举的用法、枚举的实现、枚举与单例、Enum 类 Java 枚举如何比较 switch 对枚举的支持 枚举的序列化如何实现 枚举的线程安全性问题

→ IO

字符流、字节流、输入流、输出流、同步、异步、阻塞、非阻塞、Linux 5 种 IO 模型 BIO、NIO 和 AIO 的区别、三种 IO 的用法与原理、netty

→ 反射

反射与工厂模式、反射有什么用 Class 类、java.lang.reflect.*

→ 动态代理

静态代理、动态代理 动态代理和反射的关系 动态代理的几种实现方式 AOP

→ 序列化

什么是序列化与反序列化、为什么序列化、序列化底层原理、序列化与单例模式、protobuf、为什么说序列化并不安全

→ 注解

元注解、自定义注解、Java 中常用注解使用、注解与反射的结合 Spring 常用注解

→ JMS

什么是 Java 消息服务、JMS 消息传送模型

→ JMX

java.lang.management.*、javax.management.*

→ 泛型

泛型与继承、类型擦除、泛型中 KTVE? object 等的含义、泛型各种用法 限定通配符和非限定通配符、上下界限定符 extends 和 super List<Object> 和原始类型 List 之间的区别？

List<?> 和 List<Object> 之间的区别是什么?

→ **单元测试**

junit、mock、mockito、内存数据库（h2）

→ **正则表达式**

java.lang.util.regex.*

→ **常用的 Java 工具库**

commons.lang、commons.*...、guava-libraries、netty

→ **API & SPI**

API、API 和 SPI 的关系和区别

如何定义 SPI、SPI 的实现原理

→ **异常**

异常类型、正确处理异常、自定义异常

Error 和 Exception

异常链、try-with-resources

finally 和 return 的执行顺序

→ **时间处理**

时区、冬令时和夏令时、时间戳、Java 中时间 API

格林威治时间、CET,UTC,GMT,CST 几种常见时间的含义和关系

SimpleDateFormat 的线程安全性问题

Java 8 中的时间处理

如何在东八区的计算机上获取美国时间

→ **编码方式**

Unicode、有了 Unicode 为啥还需要 UTF-8

GBK、GB2312、GB18030 之间的区别

UTF8、UTF16、UTF32 区别

URL 编解码、Big Endian 和 Little Endian

如何解决乱码问题

→ **语法糖**

Java 中语法糖原理、解语法糖

语法糖：switch 支持 String 与枚举、泛型、自动装箱与拆箱、方法变长参数、枚举、内部类、条件编译、断言、数值字面量、for-each、try-with-resource、Lambda 表达式

03 阅读源代码

String、Integer、Long、Enum、

BigDecimal、ThreadLocal、ClassLoader & URLClassLoader、

ArrayList & LinkedList、

HashMap & LinkedHashMap & TreeMap & CouncurrentHashMap、HashSet & LinkedHashSet & TreeSet

04 Java 并发编程

→ **并发与并行**

什么是并发、什么是并行

并发与并行的区别

→ 什么是线程，与进程的区别

线程的实现、线程的状态、优先级、线程调度、创建线程的多种方式、守护线程

线程与进程的区别

→ 线程池

自己设计线程池、submit() 和 execute()、线程池原理

为什么不允许使用 Executors 创建线程池

→ 线程安全

死锁、死锁如何排查、线程安全和内存模型的关系

→ 锁

CAS、乐观锁与悲观锁、数据库相关锁机制、分布式锁、偏向锁、轻量级锁、重量级锁、monitor、

锁优化、锁消除、锁粗化、自旋锁、可重入锁、阻塞锁、死锁

→ 死锁

什么是死锁

死锁如何解决

→ synchronized

synchronized 是如何实现的？

synchronized 和 lock 之间关系、不使用 synchronized 如何实现一个线程安全的单例

synchronized 和原子性、可见性和有序性之间的关系

→ volatile

happens-before、内存屏障、编译器指令重排和 CPU 指令重

volatile 的实现原理

volatile 和原子性、可见性和有序性之间的关系

有了 symchronized 为什么还需要 volatile

→ sleep 和 wait

→ wait 和 notify

→ notify 和 notifyAll

→ ThreadLocal

→ 写一个死锁的程序

→ 写代码来解决生产者消费者问题

→ 井方包

Thread 、 Runnable 、 Callable 、 ReentrantLock 、 ReentrantReadWriteLock 、 Atomic* 、 Semaphore 、 CountdownLatch 、 ConcurrentHashMap、 Executors

2

底层篇

01 JVM

→ JVM 内存结构

class 文件格式、运行时数据区：堆、栈、方法区、直接内存、运行时常量池、

堆和栈区别

Java 中的对象一定在堆上分配吗？

→ Java 内存模型

计算机内存模型、缓存一致性、MESI 协议
可见性、原子性、顺序性、happens-before、
内存屏障、synchronized、volatile、final、锁

→ 垃圾回收

GC 算法：标记清除、引用计数、复制、标记压缩、分代回收、增量式回收
GC 参数、对象存活的判定、垃圾收集器（CMS、G1、ZGC、Epsilon）

→ JVM 参数及调优

-Xmx、-Xmn、-Xms、Xss、-XX:SurvivorRatio、
-XX:PermSize、-XX:MaxPermSize、-XX:MaxTenuringThreshold

→ Java 对象模型

oop-klass、对象头

→ HotSpot

即时编译器、编译优化

→ 虚拟机性能监控与故障处理工具

jps, jstack, jmap, jstat, jconsole, jinfo, jhat, javap, btrace, TProfiler
Arthas

02 类加载机制

ClassLoader、类加载过程、双亲委派（破坏双亲委派）、模块化（jboss modules、osgi、jigsaw）

03 编译与反编译

什么是编译（前端编译、后端编译）、什么是反编译
JIT、JIT 优化（逃逸分析、栈上分配、标量替换、锁优化）
编译工具：javac
反编译工具：javap、jad、CRF

3 进阶篇

01 Java 底层知识

→ 字节码、class 文件格式

→ CPU 缓存，L1，L2，L3 和伪共享

→ 尾递归

→ 位运算

用位运算实现加、减、乘、除、取余

02 设计模式

设计模式的六大原则：

开闭原则（Open Close Principle）、里氏代换原则（Liskov Substitution Principle）、依赖倒转原则（Dependence Inversion Principle）

接口隔离原则（Interface Segregation Principle）、迪米特法则（最少知道原则）（Demeter Principle）、合成复用原则（Composite Reuse Principle）

→ **了解 23 种设计模式**

创建型模式：单例模式、抽象工厂模式、建造者模式、工厂模式、原型模式。

结构型模式：适配器模式、桥接模式、装饰模式、组合模式、外观模式、享元模式、代理模式。

行为型模式：模版方法模式、命令模式、迭代器模式、观察者模式、中介者模式、备忘录模式、解释器模式（Interpreter 模式）、状态模式、策略模式、职责链模式(责任链模式)、访问者模式。

→ **会使用常用设计模式**

单例的七种写法：懒汉——线程不安全、懒汉——线程安全、饿汉、饿汉——变种、静态内部类、枚举、双重校验锁

工厂模式、适配器模式、策略模式、模板方法模式、观察者模式、外观模式、代理模式等必会

→ **不用 synchronized 和 lock，实现线程安全的单例模式**

→ **实现 AOP**

→ **实现 IOC**

→ **nio 和 reactor 设计模式**

03 网络编程知识

→ **tcp、udp、http、https 等常用协议**

三次握手与四次关闭、流量控制和拥塞控制、OSI 七层模型、tcp 粘包与拆包

→ **http/1.0 http/1.1 http/2 之前的区别**

http 中 get 和 post 区别

常见的 web 请求返回的状态码

404、302、301、500分别代表什么

→ **http/3**

→ **Java RMI, Socket, HttpClient**

→ **cookie 与 session**

cookie 被禁用，如何实现 session

→ **用 Java 写一个简单的静态文件的 HTTP 服务器**

→ **了解 nginx 和 apache 服务器的特性并搭建一个对应的服务器**

→ **用 Java 实现 FTP、SMTP 协议**

→ **进程间通讯的方式**

→ **什么是 CDN? 如果实现?**

→ **DNS**

什么是 DNS、记录类型: A 记录、CNAME 记录、AAAA 记录等

域名解析、根域名服务器

DNS 污染、DNS 劫持、公共 DNS: 114 DNS、Google DNS、OpenDNS

→ **反向代理**

正向代理、反向代理

反向代理服务器

04 框架知识

→ Servlet

生命周期

线程安全问题

filter 和 listener

web.xml 中常用配置及作用

→ Hibernate

什么是 OR Mapping

Hibernate 的懒加载

Hibernate 的缓存机制

Hibernate / Ibatis / MyBatis 之间的区别

→ Spring

Bean 的初始化

AOP 原理

实现 Spring 的IOC

Spring 四种依赖注入方式

→ Spring MVC

什么是 MVC

Spring mvc 与 Struts mvc 的区别

→ Spring Boot

Spring Boot 2.0、起步依赖、自动配置、

Spring Boot 的 starter 原理，自己实现一个 starter

→ Spring Security

→ Spring Cloud

服务发现与注册：Eureka、Zookeeper、Consul

负载均衡：Feign、Spring Cloud Loadbalance

服务配置：Spring Cloud Config

服务限流与熔断：Hystrix

服务链路追踪：Dapper

服务网关、安全、消息

05 应用服务器知识

→ JBoss

→ tomcat

→ jetty

→ Weblogic

06 工具

→ git & svn

→ maven & gradle

→ IntelliJ IDEA

常用插件：Maven Helper 、FindBugs-IDEA、阿里巴巴代码规约检测、GsonFormat

Lombok plugin、.ignore、Mybatis plugin

01 新技术

→ Java 8

lambda 表达式、Stream API、时间 API

→ Java 9

Jigsaw、Jshell、Reactive Streams

→ Java 10

局部变量类型推断、G1 的并行 Full GC、ThreadLocal 握手机制

→ Java 11

ZGC、Epsilon、增强 var

→ Spring 5

响应式编程

→ Spring Boot 2.0

→ HTTP/2

→ HTTP/3

02 性能优化

使用单例、使用 Future 模式、使用线程池

选择就绪、减少上下文切换、减少锁粒度、数据压缩、结果缓存

03 线上问题分析

→ dump 获取

线程 Dump、内存 Dump、gc 情况

→ dump 分析

分析死锁、分析内存泄露

→ dump 分析及获取工具

jstack、jstat、jmap、jhat、Arthas

→ 自己编写各种 outofmemory, stackoverflow 程序

HeapOutOfMemory、Young OutOfMemory、

MethodArea OutOfMemory、ConstantPool OutOfMemory、

DirectMemory OutOfMemory、Stack OutOfMemory Stack OverFlow

→ Arthas

jvm 相关、class/classloader 相关、monitor/watch/trace 相关、

options、管道、后台异步任务

文档: <https://alibaba.github.io/arthas/advanced-use.html>

→ **常见问题解决思路**

内存溢出、线程死锁、类加载冲突

→ **使用工具尝试解决以下问题，并写下总结**

当一个 Java 程序响应很慢时如何查找问题

当一个 Java 程序频繁 FullGC 时如何解决问题

如何查看垃圾回收日志

当一个 Java 应用发生 OutOfMemory 时该如何解决

如何判断是否出现死锁

如何判断是否存在内存泄露

使用 Arthas 快速排查 Spring Boot 应用404/401问题

使用 Arthas 排查线上应用日志打满问题

利用 Arthas 排查 Spring Boot 应用 NoSuchMethodError

04 编译原理知识

→ **编译与反编译**

→ **Java 代码的编译与反编译**

→ **Java 的反编译工具**

javap、jad、CRF

→ **即时编译器**

→ **编译过程**

词法分析，语法分析（LL 算法，递归下降算法，LR 算法）

语义分析，运行时环境，中间代码，代码生成，代码优化

05 操作系统知识

→ **Linux 的常用命令**

→ **进程间通信**

→ **进程同步**

生产者消费者问题、哲学家就餐问题、读者写者问题

→ **缓冲区溢出**

→ **分段和分页**

→ **虚拟内存与主存**

→ **虚拟内存管理**

→ **换页算法**

06 数据库知识

→ **MySQL 执行引擎**

→ **MySQL 执行计划**

如何查看执行计划，如何根据执行计划进行 SQL 优化

→ 索引

Hash 索引、B 树索引（B+树、和B树、R树）

普通索引、唯一索引

覆盖索引、最左前缀原则、索引下推

→ SQL 优化

→ 数据库事务和隔离级别

事务的隔离级别、事务能不能实现锁的功能

→ 数据库锁

行锁、表锁、使用数据库锁实现乐观锁、

→ 连接

内连接，左连接，右连接

→ 数据库主备搭建

→ binlog

→ redolog

→ 内存数据库

h2

→ 分库分表

→ 读写分离

→ 常用的 NoSql 数据库

redis、memcached

→ 分别使用数据库锁、NoSql 实现分布式锁

→ 性能调优

→ 数据库连接池

07 数据结构与算法知识

→ 简单的数据结构

栈、队列、链表、数组、哈希表、

栈和队列的相同和不同之处

栈通常采用的两种存储结构

→ 树

二叉树、字典树、平衡树、排序树、

B 树、B+ 树、R 树、多路树、红黑树

→ 堆

大根堆、小根堆

→ 图

有向图、无向图、拓扑

→ 排序算法

稳定的排序：冒泡排序、插入排序、鸡尾酒排序、桶排序、计数排序、归并排序、原地归并排序、二叉排序树排序、鸽巢排序、基数排序、侏儒排序、图书馆排序、块排序

不稳定的排序：选择排序、希尔排序、Clover 排序算法、梳排序、堆排序、平滑排序、快速排序、内省排序、耐心排序

各种排序算法和时间复杂度

→ 两个栈实现队列，和两个队列实现栈

→ **深度优先和广度优先搜索**

→ **全排列、贪心算法、KMP 算法、hash 算法**

→ **海量数据处理**

分治，hash 映射，堆排序，双层桶划分，Bloom Filter，bitmap，数据库索引，mapreduce 等。

08 大数据知识

→ **Zookeeper**

基本概念、常见用法

→ **Solr, Lucene, ElasticSearch**

在 linux 上部署 solr，solrcloud，新增、删除、查询索引

→ **Storm, 流式计算, 了解 Spark, S4**

在 linux 上部署 storm，用 zookeeper 做协调，运行 storm hello world，local 和 remote 模式运行调试 storm topology。

→ **Hadoop, 离线计算**

HDFS、MapReduce

→ **分布式日志收集 flume, kafka, logstash**

→ **数据挖掘, mahout**

09 网络安全知识

→ **XSS**

XSS 的防御

→ **CSRF**

→ **注入攻击**

SQL 注入、XML 注入、CRLF 注入

→ **文件上传漏洞**

→ **加密与解密**

对称加密、非对称加密、哈希算法、加盐哈希算法

MD5，SHA1、DES、AES、RSA、DSA

彩虹表

→ **DDOS攻击**

DOS 攻击、DDOS 攻击

memcached 为什么可以导致 DDos 攻击、什么是反射型 DDoS

如何通过 Hash 碰撞进行 DOS 攻击

→ **SSL、TLS, HTTPS**

→ **用 openssl 签一个证书部署到 apache 或 nginx**

01 分布式

数据一致性、服务治理、服务降级

→ 分布式事务

2PC、3PC、CAP、BASE、可靠消息最终一致性、最大努力通知、TCC

→ Dubbo

服务注册、服务发现，服务治理

<http://dubbo.apache.org/zh-cn/>

→ 分布式数据库

怎样打造一个分布式数据库、什么时候需要分布式数据库、

mycat、otter、HBase

→ 分布式文件系统

mfs、fastdfs

→ 分布式缓存

缓存一致性、缓存命中率、缓存冗余

→ 限流降级

Hystrix、Sentinal

→ 算法

共识算法、Raft 协议、Paxos 算法与 Raft 算法、

拜占庭问题与算法、2PC、3PC

02 微服务

SOA、康威定律

→ ServiceMesh

sidecar

→ Docker & Kubernetes

→ Spring Boot

→ Spring Cloud

03 高并发

→ 分库分表

→ CDN 技术

→ 消息队列

ActiveMQ

04 监控

→ 监控什么

CPU、内存、磁盘 I/O、网络 I/O 等

→ **监控手段**

进程监控、语义监控、机器资源监控、数据波动

→ **监控数据采集**

日志、埋点

→ **Dapper**

05 负载均衡

tomcat 负载均衡、Nginx 负载均衡

四层负载均衡、七层负载均衡

06 DNS

DNS 原理、DNS 的设计

07 CDN

数据一致性

6

扩展篇

01 云计算

IaaS、SaaS、PaaS、虚拟化技术、openstack、Serverless

02 搜索引擎

Solr、Lucene、Nutch、Elasticsearch

03 权限管理

Shiro

04 区块链

哈希算法、Merkle 树、公钥密码算法、共识算法、
Raft 协议、Paxos 算法与 Raft 算法、拜占庭问题与算法、消息认证码与数字签名

→ **比特币**

挖矿、共识机制、闪电网络、侧链、热点问题、分叉

→ **以太坊**

→ **超级账本**

05 人工智能

数学基础、机器学习、人工神经网络、深度学习、应用场景。

→ **常用框架**

TensorFlow、DeepLearning4J

06 IoT

07 量子计算

08 AR & VR

09 其他语言

Groovy、Python、Go、NodeJs、Swift、Rust

7

推荐书籍

《深入理解 Java 虚拟机》

《Effective Java》

《深入分析 Java Web 技术内幕》

《大型网站技术架构》

《代码整洁之道》

《架构整洁之道》

《Head First 设计模式》

《maven 实战》

《区块链原理、设计与应用》

《Java 并发编程实战》

《鸟哥的 Linux 私房菜》

《从Paxos 到 Zookeeper》

《架构即未来》