

JD Java 工程师面试题

一, Java 基础

- ✓ Static 方法是否能被覆盖？
- ✓ 集合不用泛型会造成什么问题？比如 `List list = new ArrayList();`
- ✓ Java 泛型中 `extends` 和 `super` 的用法？
- ✓ IO 字节流和字符流区别？
- ✓ 系列化
 - ◆ `Serializable` 和 `Externalizable` 区别？
 - ◆ `serialVersionUID` 不定义有什么问题？
- ✓ NIO？
- ✓ `sleep()`和 `wait()`方法区别？
- ✓ 为什么等待和通知是 `Object` 类而不是 `Thread` 中声明的？
- ✓ 为什么 Java 中 `wait` 方法需要在 `synchronized` 的方法中调用？(`wait` 和 `notify` 方法之间的任何潜在竞争条件)
- ✓ 为什么 Java 中不支持多重继承？(钻石问题)
- ✓ 为什么 Java 不支持运算符重载？(清晰、避免混乱、JVM 复杂性)
- ✓ 为什么 `String` 在 Java 中是不可变的？(对象池、Map 键、安全性)
- ✓ 为什么 `char` 数组比 Java 中的 `String` 更适合存储密码？(`String` 不变)
- ✓ 如何使用双重检查锁定在 Java 中创建线程安全的单例？
- ✓ `synchronized` 和 `volatile`？
- ✓ 同步方法和同步代码块区别？
- ✓ `Lock` 和 `synchronized` 区别？
- ✓ Java 中偏向锁，自旋锁，轻量级锁，和重量级锁？
- ✓ 无锁化编程的途径有哪些？
- ✓ Java 线程池 `corePoolSize`, `maximuPoolSize`, `workQueue` 的含义？
- ✓ 如何结合业务场景和机器设置合适的线程池大小？
- ✓ 同步器：`Semaphore`，`CyclicBarrier`，`CountDownLatch`，`Exchanger`？
- ✓ Java 显示锁和隐式锁区别？
- ✓ 什么是线程上下文切换？(存储和恢复 CPU 状态的过程，它使得线程执行能够从中断点恢复执行)
- ✓ 多线程最佳实践？
 - 给线程取有意义的名字
 - 避免锁定和缩小同步范围
 - 多用同步类少用 `wait` 和 `notify`
 - 多用并发集合少用同步集合
- ✓ 强引用，软 引用，弱引用？
 - 强引用：主动放，才回收
 - 软引用：内存缺，可回收，Get 可能为 null

- 弱引用：GC 被回收，Get 可能为 null
- ✓ HashMap 实现原理？在非线程安全的情况使用 HashMap 会造成什么问题？
- ✓ ConcurrentHashMap 原理？
- ✓ ThreadLocal？
- ✓ 遇到过什么 OOM 或 SOF (StackOverflow) ？
- ✓ ClassLoader 继承关系和过程？
- ✓ JVM 模型？
- ✓ final 的变量不变的是引用还是对象？
- ✓ Java 实现多态的机制？
- ✓ 两个对象 equals 为 true，却拥有不同的 hash code？
- ✓ 深克隆和浅克隆区别？
- ✓ 数据库乐观锁和悲观锁？
- ✓ SSL？
- ✓ Java 程序性能优化？
- ✓ jstack，jstat，jmap 用法？怎么使用 jstack 检测机器 CPU 飙升的问题？
- ✓ 开发会遵循那些编码规范？

二，框架

- ✓ Spring Aop？
- ✓ 如何在系统启动后执行一个 bean 的方法？
- ✓ 如何定制 Spring 中 bean 初始化的过程？
- ✓ 如何自定义 Spring 标签？
- ✓ 自定义的实例化、初始化、依赖装配、依赖检查等流程？
- ✓ Netty 原理？和高性能原理？
 - 逻辑架构？
 - 零拷贝
 - 高效的并发编程？
- ✓ 消息队列的使用场景？
- ✓ Redis 数据结构，持久化，集群和事务？
- ✓ Redis 单线程为什么并发能力那么强？
- ✓ ZooKeeper 使用场景？选举算法？

三，设计模式和原则

- ✓ 单一职责，里氏替换，依赖倒置，接口隔离，迪米特，开闭原则？
- ✓ 设计模式：模板模式，适配器，策略模式等？
- ✓ 继承和组合区别？

四，分布式

- ✓ 一致性哈希？
- ✓ Dubbo？框架如何扩展（微核+插件），API 和 SPI 区别？
- ✓ BASE 和 CAP？
- ✓ 分布式事务？
- ✓ 负载均衡技术？
- ✓ 如何构建一个高并发，高性能，高扩展性的系统？
- ✓ 如何应对类似秒杀抢红包瞬间高并发的场景？

五，数据库

- ✓ MySQL 中 myisam 和 innodb 区别？
- ✓ MySQL 的数据如何恢复到任意时间点？（全量和增量 binlog，先恢复在回放）
- ✓ MySQL 数据库设计和使用准则？
 - 尽量不在 DB 做运算，复杂运算移到程序中
 - 主键设计（INT 主键自增，和 B+树叶子节点顺序一致，效率最高）
 - 注释，禁止存储过程，视图，触发器等
 - 禁止存储大文件
 - 定义为 NOT NULL 并提供默认值
 - WHERE 条件的属性上使用函数或者表达式
 - 禁止负向查询，以及%开头的模糊查询
 - 不要使用 IS NULL，IS NOT NULL
 - UNION ALL 优于 UNION
 - 考虑使用 limit N，少用 limit M,N
- ✓ MySQL explain 和 profile 用法？
- ✓ MySQL 复制原理和流程？（dump，IO，SQL 线程）
- ✓ MySQL 中 varchar 与 char 的区别以及 varchar(50)中的 50 代表的涵义？
- ✓ MySQL 索引？B+树
- ✓ MySQL 锁？
 - gap 锁，next-key 锁，以及 innodb 的行锁是怎么实现的，以及 myisam 的锁是怎么实现
 - Innodb 的锁的策略为 next-key 锁，即 record lock+gap lock。是通过在 index 上加 lock 实现的，index 为 unique index，降级为 record lock，如普通 index，则为 next-key lock，如没有 index，则直接锁住全表。myisam 直接使用全表扫描
- ✓ 如何应对数据库的瓶颈？
- ✓ 如何水平，垂直拆分？
- ✓ 数据库优化？（设计和查询优化）
- ✓ 分布式中间件？

六，代码题目

- ✓ 给你一个日期，得出日期在当年的第几天？
- ✓ 给定一个数，判断是不是回文数？
- ✓ 用 Java 实现快速排序？

- ✓ 用 Java 实现线程安全的任务生产者和消费者模式队列？
- ✓ 画出模板方法，策略模式，装饰者模式类图？
- ✓ 下面输出什么？

```
int a = 1;
Integer b = new Integer(1);
System.out.println(a == b);
```

- ✓ 执行 SubClass 中 main 方法，输出什么？（父类静态代码块->子类静态代码块->父类普通代码块->父类构造方法->子类代码块->子类构造方法）

```
public class Parent {
    public Parent() {
        System.out.println("Parent A");
    }

    {
        System.out.println("Parent B");
    }

    static {
        System.out.println("Parent C");
    }
}
```

```
public class SubClass extends Parent {

    public SubClass() {
        System.out.println("SubClass A");
    }

    {
        System.out.println("SubClass B");
    }

    static {
        System.out.println("SubClass C");
    }

    public static void main(String[] args) {
        new SubClass();
    }
}
```

- ✓ 最终 userList 包含哪些元素？

```
List<String> userList = Lists.newArrayList("tom", "jack", "billy");  
for (String user : userList) {  
    If (user.startsWith("t")) {  
        userList.remove(user);  
    }  
}
```

- ✓ String username = new String("tom");产生几个对象？
- ✓ 用 Java 写一个死锁代码？如何避免死锁

七，开放性题目

- ✓ 我们行业未来 3 到 5 年会有什么挑战和机遇？
- ✓ 你的学习能力如何？最近在学习什么新东西？如何学习？看过什么书？
- ✓ 做过最有成就感的事情是什么？遇到最大的挫折和问题是什么？你是怎么应对和解决挫折和问题的？
- ✓ 最近看过的书有哪些？得到哪些启发呢？
- ✓ 如何判断一个数是否存在 40 亿个整数中？
- ✓ 一共有 25 匹马，有一个赛场，赛场有 5 个赛道，最多同时可以有 5 匹马一起比赛。假设每匹马速度是稳定的，通过马与马之间的比赛，问最少比多少场才能知道跑得最快的 5 匹马？
- ✓ 你有无限量的水，还有两个罐子，容量分别是 5 升和 3 升。请精确的称量出 4 升水。
 - 先把 5 升的罐子装满，然后用罐子里的水来倒满 3 升的罐子，此时 5 升罐子中还剩 $5-3=2$ 升水；倒掉 3 升罐子里的水，然后把 5 升罐子里剩下的 2 升水倒入 3 升罐子，再次把 5 升罐子装满水，并用它往 3 升罐子倒水，由于把 3 升罐子装满还需要 1 升水，因此 5 升罐子里的水量最终变成了 $5-1=4$ 升水