

- 1.手撕代码。
- 2.JVM 内存查看与分析，编写内存泄露实例
- 3.线上 CPU100%，如何定位和排查问题
- 4.如何在运行时确定对象类型？
- 5.引用传递和值传递
- 6.类加载过程，如何获得当前对象的 ClassLoader
- 7.Git 操作，Maven 多项目聚合
- 8.10 万个 ip 段如何快速查找
- 9.Java 内存模型，volatile 和 i++的线程安全
- 10.线程池实现原理，Lock 机制的实现

(1) 其实 java 线程池的实现原理很简单，说白了就是一个线程集合 workerSet 和一个阻塞队列 workQueue。当用户向线程池提交一个任务(也就是线程)时，线程池会先将任务放入 workQueue 中。workerSet 中的线程会不断的从 workQueue 中获取线程然后执行。当 workQueue 中没有任务的时候，worker 就会阻塞，直到队列中有任务了就取出来继续执行。

- 11.JVM 对 final 关键字的编译优化
- 12.ConcurrentHashMap 深入分析
- 13.HashMap 和 HashSet 的使用
- 14.Synchronized 和 Lock 的使用
- 15.Dubbo 底层实现，Dubbo 相比 webservice 等方式的优势
- 16.设计模式，服务治理等
- 17.Spring 动态加载数据源
- 18.Spring boot 应用
- 19.Spring 中的设计模式

(1) 简单工厂模式：实质是由一个工厂类根据传入的参数，动态决定应该创建哪一个产品类。

(2) 工厂方法模式：通常由应用程序直接使用 new 创建新的对象，为了将对象的创建和使用相分离，采用工厂模式,即应用程序将对象的创建及初始化职责交给工厂对象。

(3) 单例模式：保证一个类仅有一个实例，并提供一个访问它的全局访问点。

(4) 适配器模式：

(5) 包装器模式：spring 中用到的包装器模式在类名上有两种表现：一种是类名中含有 Wrapper，另一种是类名中含有 Decorator。基本上都是动态地给一个对象添加一些额外的职责。

(6) 代理模式：为其他对象提供一种代理以控制对这个对象的访问。

(7) 观察者模式：定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

(8) 策略模式：定义一系列的算法，把它们一个个封装起来，并且使它们可相互替换。本模式使得算法可独立于使用它的客户而变化。

- 20.Linux 的常用命令，简单 shell 编写
- 21.Linux 系统监控命令
- 22.Redis 的持久化方式
- 23.缓存和数据库的一致性问题
- 24.Redis、MongoDB 和 Memcached 的区别

- 25.Solr 和 Lucene，中文分词技术
- 26.爬虫的深搜和广搜，队列和栈
- 27.数据库索引的实现，B+树的结构和操作，为什么不用红黑树
- 28.SQL 表设计 Join 操作
- 29.数据库范式，聚集索引与非聚集索引
- 30.数据库存储过程，SQL 注入