

一面

1、String 内容存放在哪儿？

答：运行时常量池，然后把 jvm 的内存结构讲了一下，堆、栈、程序计数器、方法区等，每块区域的作用之类的。

2、jvm 的回收机制是什么？

问的有点大，只好从分代回收算法说起，从 GC ROOTS 的引用链机制到新生代的小空间复制算法再到老生代的标记清除算法，最后就是新生代和老生代之间的关系以及跃迁等。

3、讲讲 jvm 的类加载机制吧，委托双亲机制吧。

反射机制是如何利用类加载机制的呢？

4、懂哪些设计模式？

答：单例，建造者，观察者，模板方法模式，迭代器模式等。

5、你在项目中用到过观察者模式吗，怎么用的？

答：做统一配置组件的时候使用到了 zookeeper 的发布订阅，创建了监听链表，完成应用系统的注册和同步。

6、写个单例模式吧，把你认为效率最高的单例实现写出来。

答：当然是通过 jvm 的内部类机制实现单例了，又快又安全

7、再写一个双检查的懒汉单例模式吧。

8、为什么要用 volatile 关键字？

答：在高并发时会出现并发模式异常，volatile 可以防止指令重排，创建对象操作并不是一个原子操作，分为三个步骤

（1）构建对象：根据 Person 类元信息确定对象的大小，向 JVM 堆中申请一块内存区域并构建对象的默认信息(加载 Person 对象成员变量信息并赋默认值如 int 类型为 0，引用类型为 null)。

（2）初始化对象：然后执行对象内部生成的 init 方法，初始化成员变量值，同时执行搜集到的{}代码块逻辑，最后执行对象构造方法。

（3）引用对象：对象实例化完毕后，再把栈中的 Person 对象引用地址指向 Person 对象在堆内存中的地址.....

9、能不能讲讲线程的内存模型？

10、谈谈你对 cas 的理解。

二面

1、假设有一个场景，系统需要某个特定时间内响应用户请求，比如说 100ms 内完成用户请求，但是在最高峰的时候每单位时间几百万的用户请求，也就是高并发，但我必须要实现系统响应及时，而且高可用(不宕机)，假如你是架构师，你该如何架构这个系统，聊聊你的方案。

2、然后向我介绍了一下他们部门是做什么业务的。

3、最后问我为什么离职和薪资待遇等。

