

## 1.Java 的 Object 类的 hashCode()方法的作用

hashCode()方法返回一个整形值，用来表示对象的 hash 码，有以下约定：

Java 代码在执行期间，对一个对象多次执行 hashCode()返回的哈希值相同，前提是将对象作为 equals 比较时的标尺消息未修改。Java 程序调用两次 main 方法，对象的 hashCode 方法返回值无须一致。

如果两个对象相等，则其调用 hashCode 方法的返回值一定相等。

但是，如果两个对象的 hashCode()返回相等，不能说明两个对象相等。

因此重写 equals 方法（equals 方法在 Object 类中使用==实现，如果自定义类要使用 equals 方法则必须重写该方法）时必须重写 hashCode 方法。而 hashCode 方法最主要的作用是增强哈希表的性能，如 hashMap，使用 hashCode 可以快速得出对象的哈希码，然后根据哈希算法得出对象的位置，直接判断此位置是否存在对象。

Object 类中的方法 :hashCode();equals();getClass();clone();wait();wait(long);wait(long,int);notify();notifyAll();toString();finalize();

## 2.Java 何时会使用到 wait 方法

wait () 必须必须在获取了对象的锁（monitor）才能执行该对象的 wait 方法（notify 方法同样如此）。当调用 wait 方法后，线程会释放对象的锁，与 Thread 类的 sleep 方法不同的是，sleep 方法只能使线程陷入休眠状态。

## 3.Java 执行 sleep 方\*\*\*不会释放资源

sleep ， 释放 cpu 资源，不释放锁资源，如果线程进入 sleep 的话，释放 cpu 资源，如果外层包有 Synchronize，那么此锁并没有释放掉。wait 用于锁机制，sleep 不是，这就是 sleep 不释放锁，wait 释放锁的原因，sleep 是线程的方法，跟锁没关系，wait，notify,notifyall 都是 Object 对象的方法，是一起使用的，用于锁机制。

## 4.Java 中有哪些锁

锁的分类有：公平锁/非公平锁、共享锁/排它锁、互斥锁/读写锁、乐观锁/悲观锁。可重入锁/不可重入锁、分段锁、轻量级锁/重量级锁/偏向锁、自旋锁。

我的回答是：synchronized，cas，lock，Volatile 等。

## 5.Volatile 关键字和锁有区别吗？

有区别，volatile 关键字只能保证变量的有序性和防止指令重排序，不能保证变量的原子性，所以他只能适用一些简单的数据结构，如 Java 中的基本类型，而不支持一些计算的操作。

ReentrantLock 默认是非公平锁；

## 6.hashMap 中的 hash 方法有什么作用？

异或其 hashCode 获得值的高 16 位的值，目的是提高返回值的随机性，减少 hash 冲突。

## 7.put 一个对象的过程

put 一个对象时，先定位对象在数组中的位置，如果该位置没有元素存在，就直接插入。如果该位置有元素存在，就先与要插入位置的 key 进行比较，如果 key 相同就直接把 value 进行覆盖，如果 key 不同就判断该位置是否是一个树节点如果是就把元素添加到树中，如果不是就把元素插入元素的尾部（如果此时 hashMap 底层数组长度大于等于 64，且链表节点的元素个数大于 8 就会对链表进行树化操作）

## 8.spring 的 IOC 和 AOP 使用了哪些 Java 底层的技术

AOP 中用到了动态代理和反射以及动态字节码技术（不写出 Java 文件通过代码直接生成 class 字节码文件）

## 9.JVM 对 synchronized 关键字的优化

1.锁消除（逃逸分析）：JVM 在编译时会对 synchronized 中的代码进行扫描，去除那些不可

能发生资源共享的锁，从而减少线程请求锁空间。

2.锁粗化：连续多个加锁解锁操作连在一起，扩展成一个范围更大的锁，将锁的粒度放大：如在 for 循环中的锁会被放在 for 循环外。

3.使用偏向锁和轻量级锁的锁升级概念优化 synchronized 获取锁和释放锁带来的性能消耗。

10.http 的请求的报文结构是什么

请求头、请求行和请求体组成。

请求行包涵请求的请求方法，请求地址和 http 的版本号

请求头包括请求首部字段，主要包括若干属\*\*\*端根据此获取客户端信息

请求体是请求的具体内容包含多个键值对组成。

11.http 的 keep-alive 有什么作用，及其与 tcpkeep-alive 的区别

http 的 keep-alive 的作用是开启 http 的长连接，减少 tcp 连接建立的次数，以提高性能和服务器的吞吐量。但是 keep-alive 也存在一些问题，长时间的 tcp 连接会导致系统资源的无效占用，故要正确设置 keep-alive timeout 时间（即服务器连接建立后如果 keep-alive timeout 的时间内客户端与服务端无请求那么就会断开连接）

Tcp 的 Keepalive：当超过一段时间之后，TCP 自动发送一个数据为空的报文（探测包）给对方，如果对方回应了这个报文，说明对方还在线，连接可以继续保持，如果对方没有报文返回，并且重试了多次之后则认为链接丢失，没有必要保持连接。

http keep-alive 与 tcp keep-alive，不是同一回事，意图不一样。http keep-alive 是为了让 tcp 活得更久一点，以便在同一个连接上传送多个 http，提高 socket 的效率。而 tcp keep-alive 是 TCP 的一种检测 TCP 连接状况的保鲜机制。

13.http 的四次挥手

客户端进程发出连接释放报文，并且停止发送数据。释放数据报文首部，FIN=1，其序列号为 seq=u（等于前面已经传送过来的数据的最后一个字节的序号加 1），此时，客户端进入 FIN-WAIT-1（终止等待 1）状态。TCP 规定，FIN 报文段即使不携带数据，也要消耗一个序号。

服务器收到连接释放报文，发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号 seq=v，此时，服务端就进入了 CLOSE-WAIT（关闭等待）状态。TCP 服务器通知高层的应用进程，客户端向服务器的方向就释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个 CLOSE-WAIT 状态持续的时间

客户端收到服务器的确认请求后，此时，客户端就进入 FIN-WAIT-2（终止等待 2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的的数据）。

服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，ack=u+1，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为 seq=w，此时，服务器就进入了 LAST-ACK（最后确认）状态，等待客户端的确认。

客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是 seq=u+1，此时，客户端就进入了 TIME-WAIT（时间等待）状态。注意此时 TCP 连接还没有释放，必须经过  $2 * MSL$ （最长报文段寿命）的时间后，当客户端撤销相应的 TCB 后，才进入 CLOSED 状态。

服务器只要收到了客户端发出的确认，立即进入 CLOSED 状态。同样，撤销 TCB 后，就结束了这次的 TCP 连接。可以看到，服务器结束 TCP 连接的时间要比客户端早一些。

为什么客户端最后还要等待 2MSL？

MSL（Maximum Segment Lifetime），TCP 允许不同的实现可以设置不同的 MSL 值。

第一，保证客户端发送的最后一个 ACK 报文能够到达服务器，因为这个 ACK 报文可能丢失，

站在服务器的角度来看，我已经发送了 FIN+ACK 报文请求断开了，客户端还没有给我回应，应该是我发送的请求断开报文它没有收到，于是服务器又会重新发送一次，而客户端就能在这个 2MSL 时间段内收到这个重传的报文，接着给出回应报文，并且会重启 2MSL 计时器。第二，防止类似与“三次握手”中提到了的“已经失效的连接请求报文段”出现在本连接中。客户端发送完最后一个确认报文后，在这个 2MSL 时间中，就可以使本连接持续的时间内所产生的所有报文段都从网络中消失。这样新的连接中不会出现旧连接的请求报文。

为什么建立连接是三次握手，关闭连接确是四次挥手呢？

建立连接的时候，服务器在 LISTEN 状态下，收到建立连接请求的 SYN 报文后，把 ACK 和 SYN 放在一个报文里发送给客户端。

而关闭连接时，服务器收到对方的 FIN 报文时，仅仅表示对方不再发送数据了但是还能接收数据，而自己也未必全部数据都发送给对方了，所以己方可以立即关闭，也可以发送一些数据给对方后，再发送 FIN 报文给对方来表示同意现在关闭连接，因此，己方 ACK 和 FIN 一般都会分开发送，从而导致多了一次。

关于四次挥手部分参考：<https://blog.csdn.net/qzcsu/article/details/72861891>

#### 14.Java 的垃圾回收机制

没答出来面试官题示说先简述 JVM 垃圾回收分那些区域，内存是怎么划分的，而各个区域用的垃圾回收。

#### 15.为什么要使用索引

通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。

可以大大加快 数据的检索速度（大大减少的检索的数据量），这也是创建索引的最主要的原因。

帮助服务器避免排序和临时表。

将随机 IO 变为顺序 IO

可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。

算法题：

1. 找一个链表的中间结点

双指针，快慢指针

2. 不使用迭代前序输出一个二叉树

栈

3. 理发师问题：一个理发师给  $n$  个人理发，每个人理发时间  $t[i]$ ，不能同时给两个相邻的人理发求最大理发时间。例：4 个人  $t=\{2, 5, 3, 9\}$  最大时间为  $5+9=14$

动态规划

面试官很 nice 我一些记不清楚的地方会引导我去回答，但奈何本人太菜，感觉一面必挂。就把面经发出来攒攒运气。