

1、手撕代码。

2、String 类的常用方法都有哪些？

3、简述线程，程序、进程的基本概念。以及他们之间关系是什么？

(1) 线程与进程相似，但线程是一个比进程更小的执行单位。一个进程在其执行的过程中可以产生多个线程。与进程不同的是同类的多个线程共享同一块内存空间和一组系统资源，所以系统在产生一个线程，或是在各个线程之间作切换工作时，负担要比进程小得多，也正因为如此，线程也被称为轻量级进程。

(2) 程序是含有指令和数据文件，被存储在磁盘或其他的数据存储设备中，也就是说程序是静态的代码。

(3) 进程是程序的一次执行过程，是系统运行程序的基本单位，因此进程是动态的。系统运行一个程序即是一个进程从创建，运行到消亡的过程。简单来说，一个进程就是一个执行中的程序，它在计算机中一个指令接着一个指令地执行着，同时，每个进程还占有某些系统资源如 CPU 时间，内存空间，文件，文件，输入输出设备的使用权等等。换句话说，当程序在执行时，将会被操作系统载入内存中。线程是进程划分成的更小的运行单位。线程和进程最大的不同在于基本上各进程是独立的，而各线程则不一定，因为同一进程中的线程极有可能会相互影响。从另一角度来说，进程属于操作系统的范畴，主要是同一段时间内，可以同时执行一个以上的程序，而线程则是在同一程序内几乎同时执行一个以上的程序段。

4、多线程锁的升级原理是什么？

5、final 关键字主要用在哪些地方？

(1) 对于一个 final 变量，如果是基本数据类型的变量，则其数值一旦在初始化之后便不能更改；如果是引用类型的变量，则在对其初始化之后便不能再让其指向另一个对象。

(2) 当用 final 修饰一个类时，表明这个类不能被继承。final 类中的所有成员方法都会被隐式地指定为 final 方法。

(3) 使用 final 方法的原因有两个。第一个原因是把方法锁定，以防任何继承类修改它的含义；第二个原因是效率。在早期的 Java 实现版本中，会将 final 方法转为内嵌调用。但是如果方法过于庞大，可能看不到内嵌调用带来的任何性能提升。类中所有的 private 方法都隐式地指定为 final。

6、hashCode () 与 equals () 的相关规定

(1) 如果两个对象相等，则 hashCode 一定也是相同的

(2) 两个对象相等,对两个对象分别调用 equals 方法都返回 true

(3) 两个对象有相同的 hashCode 值，它们也不一定是相等的

(4) 因此，equals 方法被覆盖过，则 hashCode 方法也必须被覆盖

(5) hashCode()的默认行为是对堆上的对象产生独特值。如果没有重写 hashCode(), 则该 class 的两个对象无论如何都不会相等（即使这两个对象指向相同的数据）

7、如何实现数组和 List 之间的转换？