# Java多线程面试题

## 多线程有什么用？

### 发挥多核CPU的优势

单核CPU上所谓的"多线程"那是假的多线程，多核CPU上的多线程才是真正的多线程

### **防止阻塞**

## 创建线程的方式

一般就是两种：

1）继承Thread类

2）实现Runnable接口

还有实现Callable接口和使用线程池创建

至于哪个好，不用说肯定是后者好，因为实现接口的方式比继承类的方式更灵活，也能减少程序之间的耦合度，面向接口编程也是设计模式6大原则的核心

## start()方法和run()方法的区别

只有调用了start()方法，才会表现出多线程的特性，不同线程的run()方法里面的代码交替执行。如果只是调用run()方法，那么代码还是同步执行的，必须等待一个线程的run()方法里面的代码全部执行完毕之后，另外一个线程才可以执行其run()方法里面的代码。

Start()启动线程，线程执行run()方法。

## Runnable接口和Callable接口的区别

Runnable接口中的run()方法的返回值是void，它做的事情只是纯粹地去执行run()方法中的代码而已；

Callable接口中的call()方法是有返回值的，是一个泛型，和Future、FutureTask配合可以用来获取异步执行的结果。Callable+Future/FutureTask却可以获取多线程运行的结果，可以在等待时间太长没获取到需要的数据的情况下取消该线程的任务，真的是非常有用。

## CyclicBarrier和CountDownLatch的区别

两个看上去有点像的类，都在java.util.concurrent下，都可以用来表示代码运行到某个点上，二者的区别在于：

1）CyclicBarrier的某个线程运行到某个点上之后，该线程即停止运行，直到所有的线程都到达了这个点，所有线程才重新运行；CountDownLatch则不是，某线程运行到某个点上之后，只是给某个数值-1而已，该线程继续运行。

2）CyclicBarrier只能唤起一个任务，CountDownLatch可以唤起多个任务。

3) CyclicBarrier可重用，CountDownLatch不可重用，计数值为0该CountDownLatch就不可再用了。

****volatile关键字的作用****

多线程主要围绕可见性和原子性两个特性而展开，使用volatile关键字修饰的变量，保证了其在多线程之间的可见性，即每次读取到volatile变量，一定是最新的数据。

****什么是线程安全****

****一个线程如果出现了运行时异常会怎么样****

****如何在两个线程之间共享数据****

****sleep方法和wait方法有什么区别****

****ThreadLocal有什么用****

****为什么wait()方法和notify()/notifyAll()方法要在同步块中被调用****

****为什么要使用线程池****

****怎么检测一个线程是否持有锁****

****synchronized和ReentrantLock的区别****

****ConcurrentHashMap的并发度是什么****

****FutureTask是什么****

****怎么唤醒一个阻塞的线程****

****什么是多线程的上下文切换****

****如果你提交任务时，线程池队列已满，这时会发生什么****

****Java中用到的线程调度算法是什么****

****Thread.sleep(0)的作用是什么****

****什么是CAS****

****什么是乐观锁和悲观锁****

****Semaphore有什么作用****

****同步方法和同步块，哪个是更好的选择****