Java 并发工具类

1. 等待多线程完成的countdownlatch

Countdownlatch 允许一个或多个线程等待其他线程完成操作。

Countdownlatch 的构造函数提供了一个int 类型的参数作为计数器，当我们调用countdownlatch的countdown方法时，N就会减1，countdownlatch的await方法会阻塞当前线程，直到N变为零。

1. 同步屏障 CyclicBarrier

让一个线程达到一个屏障（同步点）的时候被阻塞，直到最后一个线程到达屏障时，所有被屏障拦截的线程才会继续运行。

Countdownlatch 的计数器只能使用一次，而cyclicBarrier的计数器可以使用reset（）方法重置。如果计算发生错误，可以充值计数器，并让线程重新执行一次。

1. 信号量

信号量是用来控制访问特定资源的线程数量。信号量是通过一个计数器控制对共享资源的访问，信号量的值是一个非负整数，所有通过它的线程都会将该整数减一。如果计数器大于0，则访问被允许，计数器减1；如果为0，则访问被禁止，所有试图通过它的线程都将处于等待状态。

1. 信号量和互斥锁的区别？

互斥用于线程的互斥；信号量用于线程的同步。互斥量只能为0/1，信号量值可以为非负整数。

互斥：是指某一资源同时只允许一个访问者对其进行访问，具有唯一性和排它性。但互斥无法限制访问者对资源的访问顺序，即访问是无序的。

同步：是指在互斥的基础上（大多数情况），通过其它机制实现访问者对资源的有序访问。在大多数情况下，同步已经实现了互斥，特别是所有写入资源的情况必定是互斥的。少数情况是指可以允许多个访问者同时访问资源。

1. 线程交换

Exchanger 是一个线程间携作的工具类。Exchanger用于线程间的数据交换。它提供一个同步点，在这个同步点，两个线程可以交换彼此的数据。这两个线程通过exchange方法交换数据。

如果一个线程执行了exchange方法交换数据，它会一直等待第二个线程也执行exchange方法。