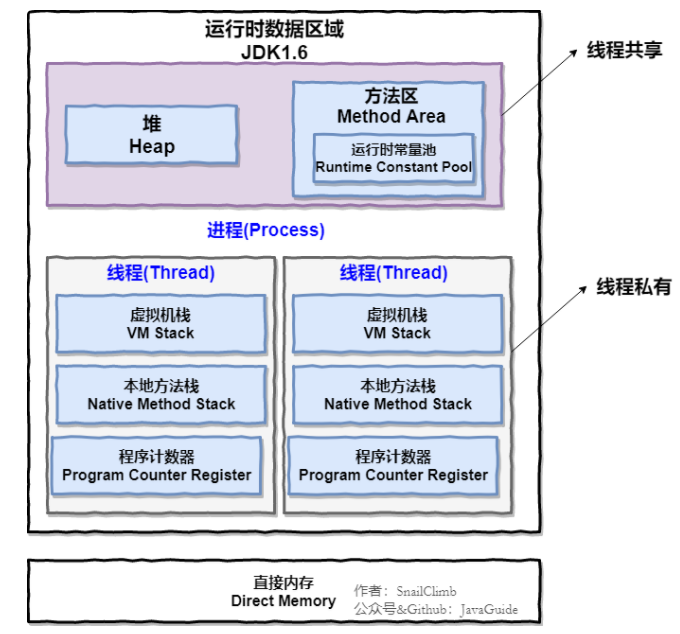
1、进程与线程

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进程的堆和方法区资源，但每个线程有自己的程序计数器、虚拟机栈和本地方法栈，



程序计数器私有主要是为了**线程切换后能恢复到正确的执行位置**。

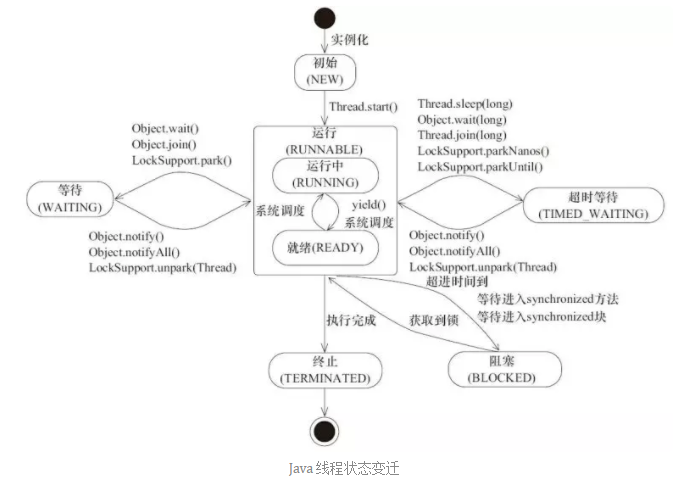
为了**保证线程中的局部变量不被别的线程访问到**，虚拟机栈和本地方法栈是线程私有的。

2、堆和方法区

堆和方法区是所有线程共享的资源，其中堆是进程中最大的一块内存，主要用于存放新创建的对象 (所有对象都在这里分配内存)，方法区主要用于存放已被加载的类信息、常量、静态变量、即时编译器编译后的代码等数据。

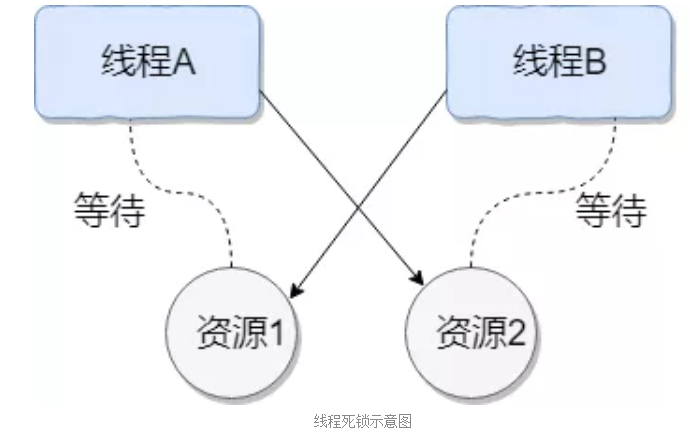
2、堆和方法区

**线程的生命周期**



3、死锁

多个线程同时被阻塞，它们中的一个或者全部都在等待某个资源被释放。由于线程被无限期地阻塞，因此程序不可能正常终止。



4、说说 sleep() 方法和 wait() 方法区别和共同点?

（1）两者最主要的区别在于：sleep 方法没有释放锁，而 wait 方法释放了锁 。

（2）两者都可以暂停线程的执行。

（3）Wait 通常被用于线程间交互/通信，sleep 通常被用于暂停执行。

（4）wait() 方法被调用后，线程不会自动苏醒，需要别的线程调用同一个对象上的 notify() 或者 notifyAll() 方法。sleep() 方法执行完成后，线程会自动苏醒。或者可以使用 wait(long timeout)超时后线程会自动苏醒。

5、为什么我们调用 start() 方法时会执行 run() 方法，为什么我们不能直接调用 run() 方法？

总结：调用 start 方法方可启动线程并使线程进入就绪状态，而 run 方法只是 thread 的一个普通方法调用，还是在主线程里执行。