

第八章

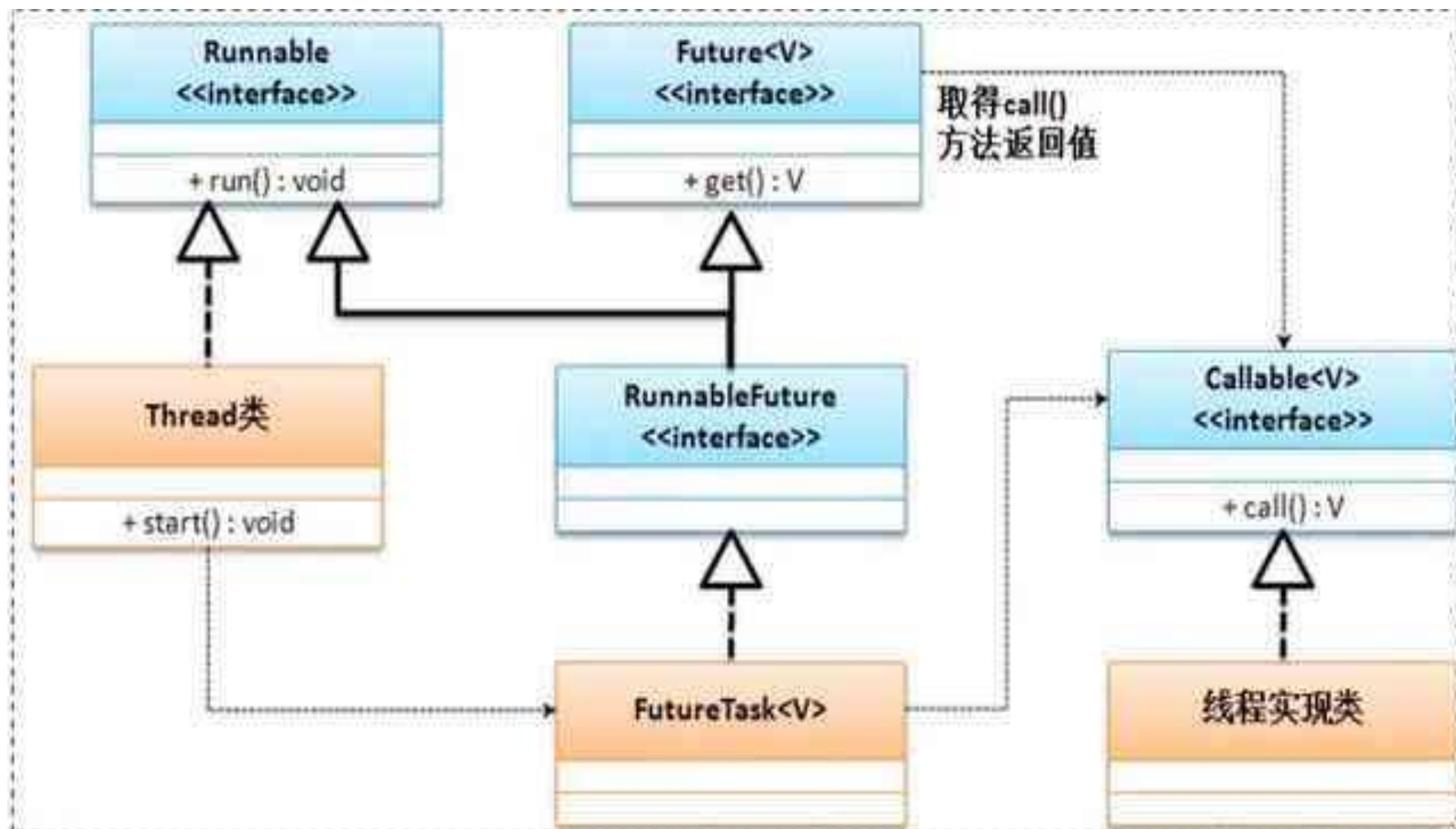
线程-java的线程的创建



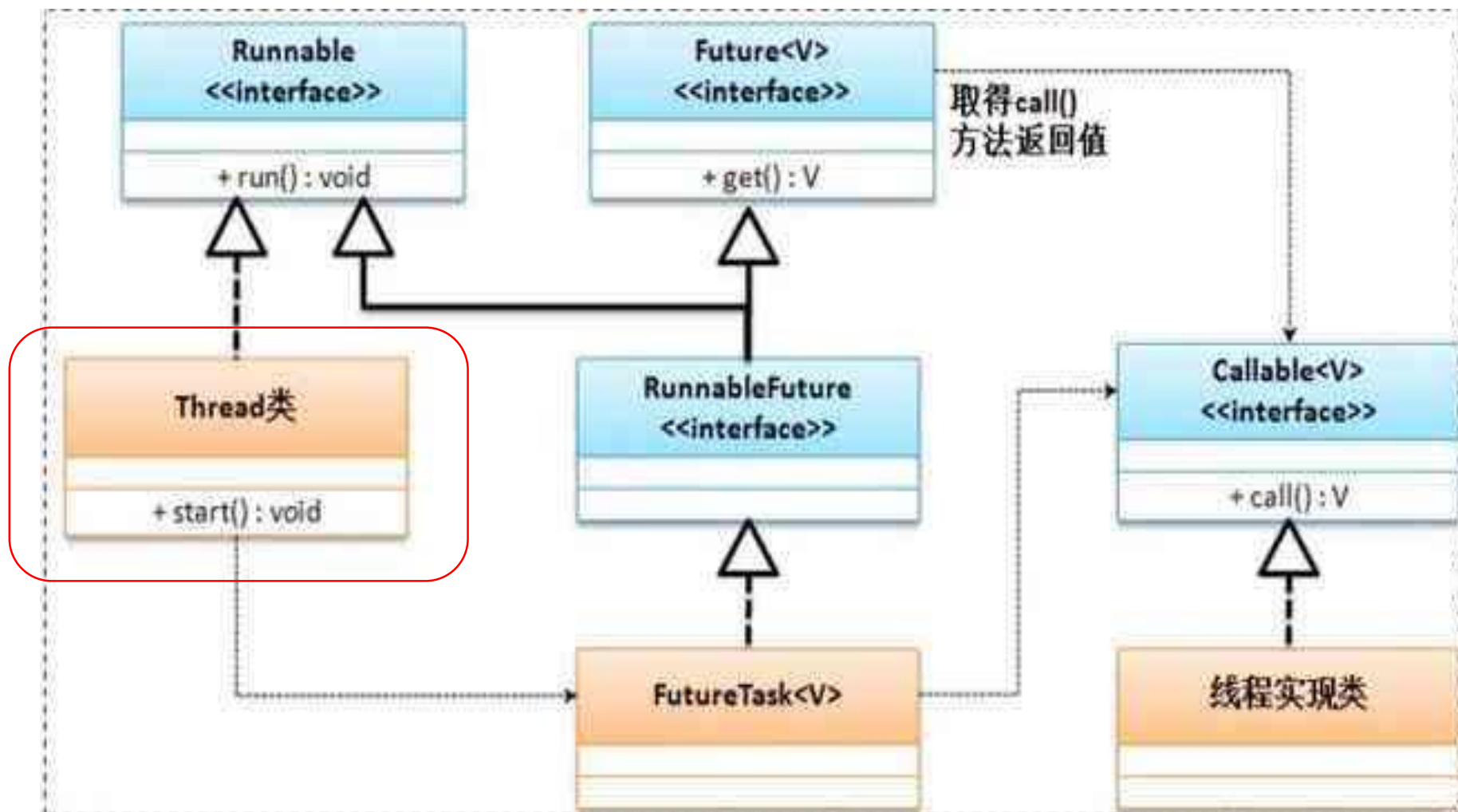
东北林业大学

NORTHEAST FORESTRY UNIVERSITY

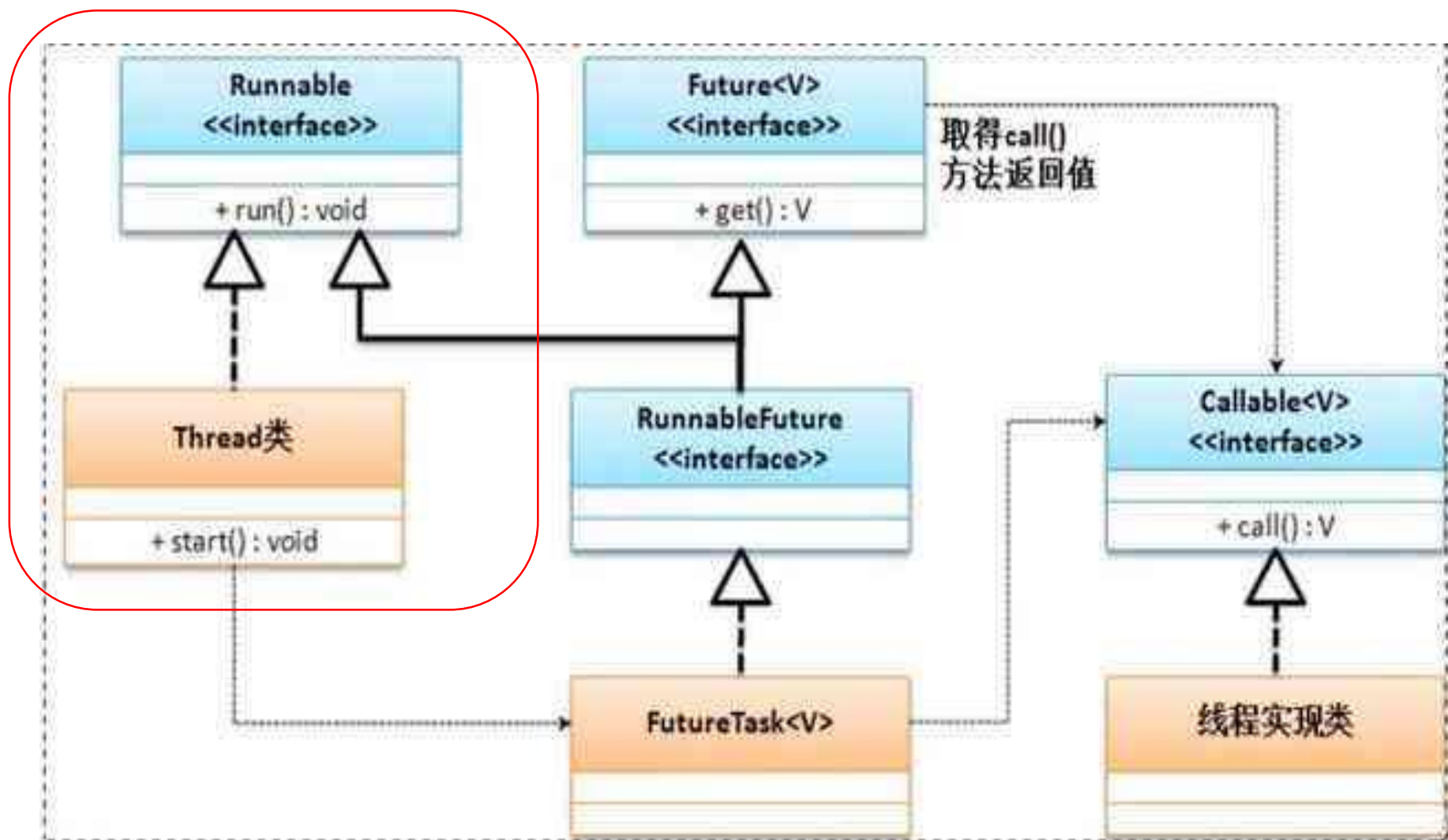
- 1) 继承Thread类创建线程
- 2) 实现Runnable接口创建线程
- 3) 使用Callable和Future创建线程
- 4) 使用线程池例如用Executor框架
(线程池部分后面小结讲解)



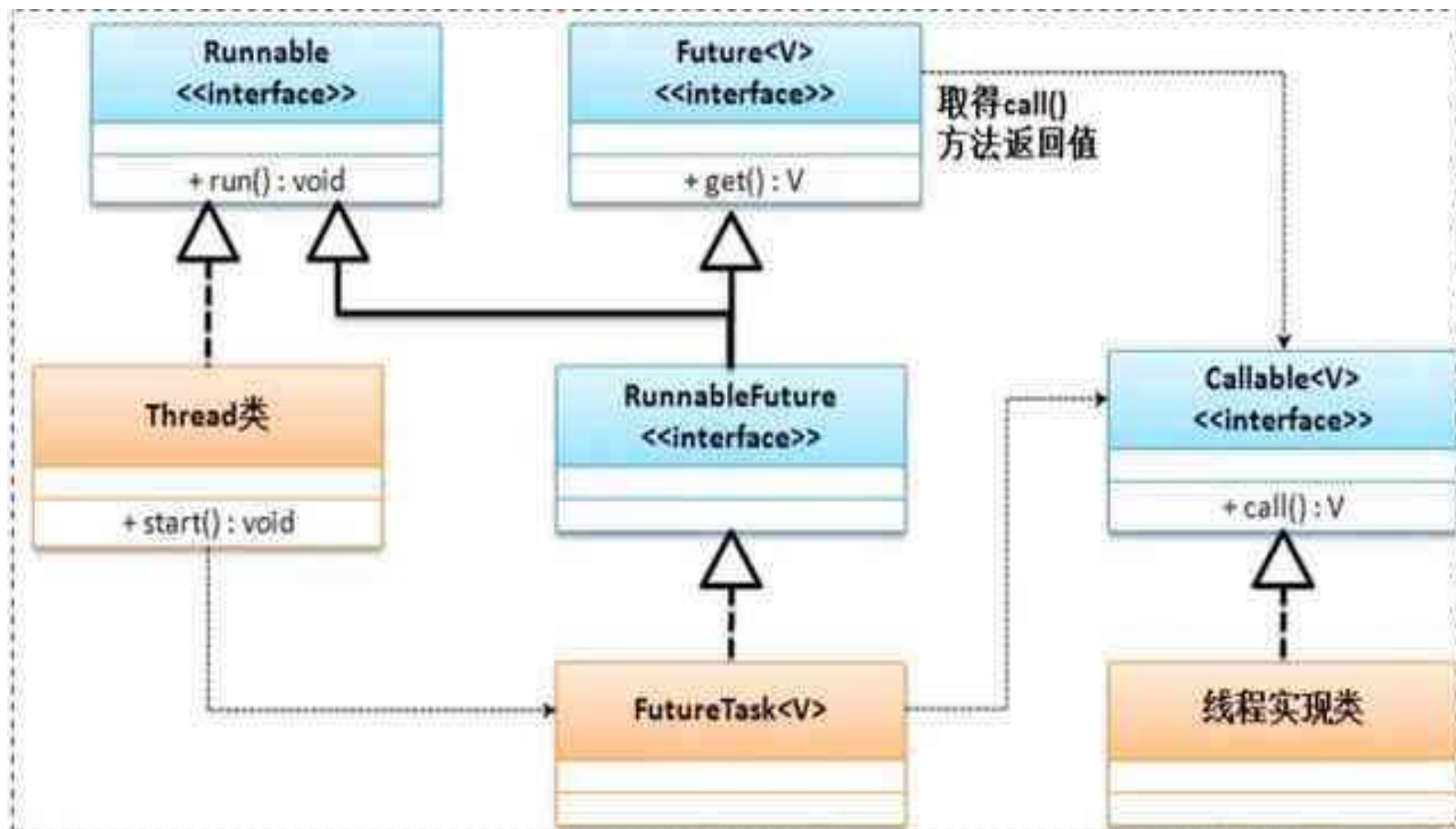
继承Thread类创建线程



实现Runnable接口创建线程



使用Callable和Future创建线程

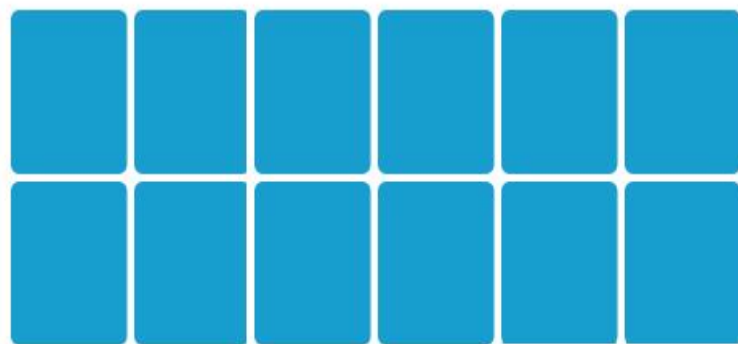


- 1. 采用实现 Runnable、Callable 接口的方式创建多线程时，线程类只是实现了 Runnable 接口或 Callable 接口，还可以继承其他类。
- 2. 使用继承 Thread 类的方式创建多线程时，编写简单，如果需要访问当前线程，则无需使用 Thread.currentThread() 方法，直接使用 this 即可获得当前线程。

会看到用匿名内部类实现

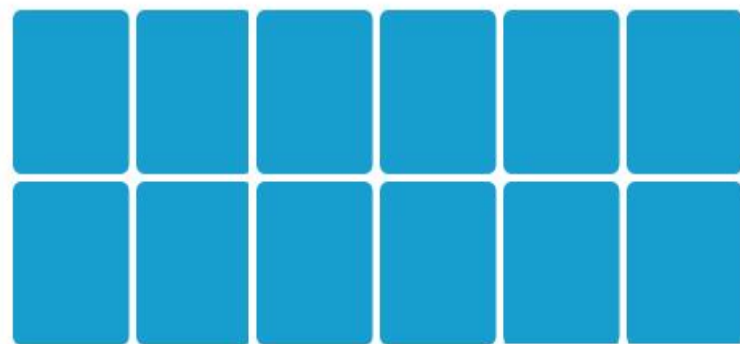
- 1. 匿名方式的Thread类
- 2. 匿名方式的Runnable接口

lambda表达式线程创建



东北林业大学

NORTHEAST FORESTRY UNIVERSITY



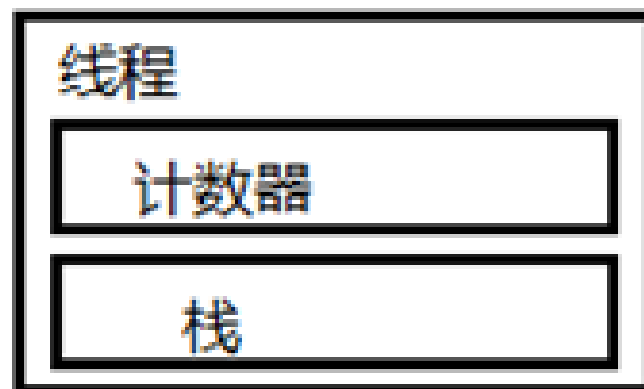
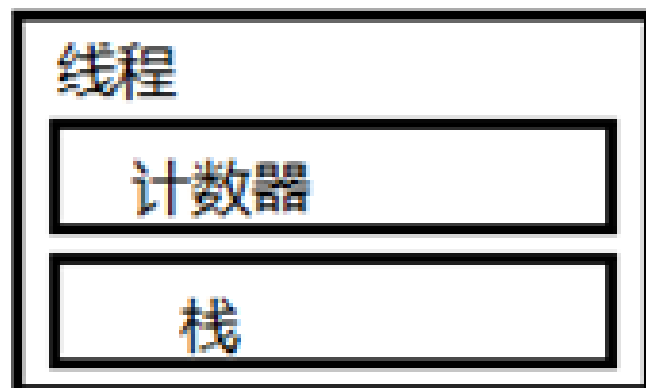
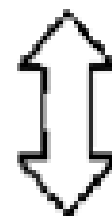
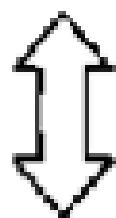
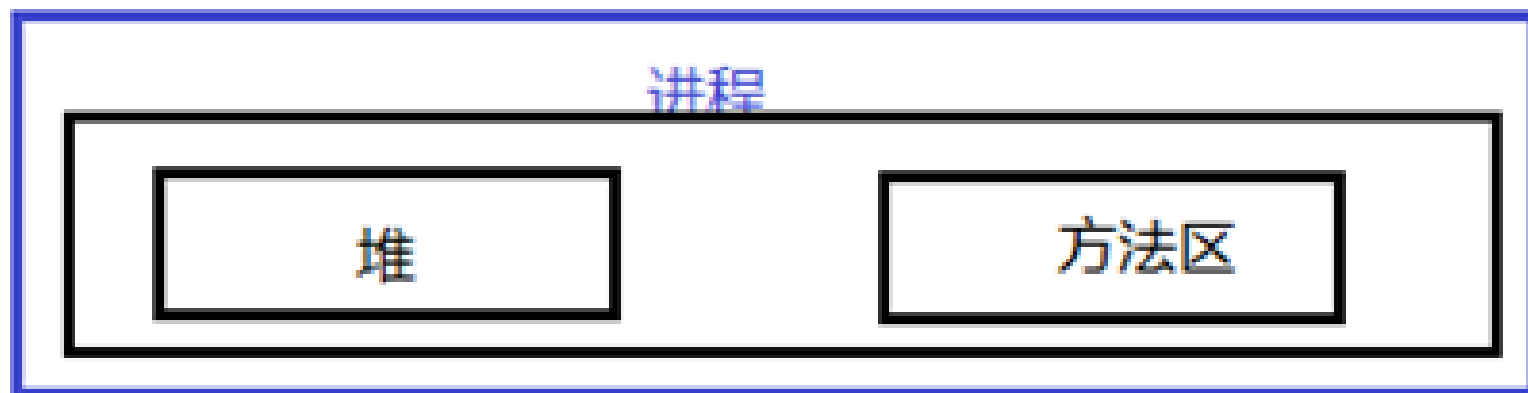
第八章

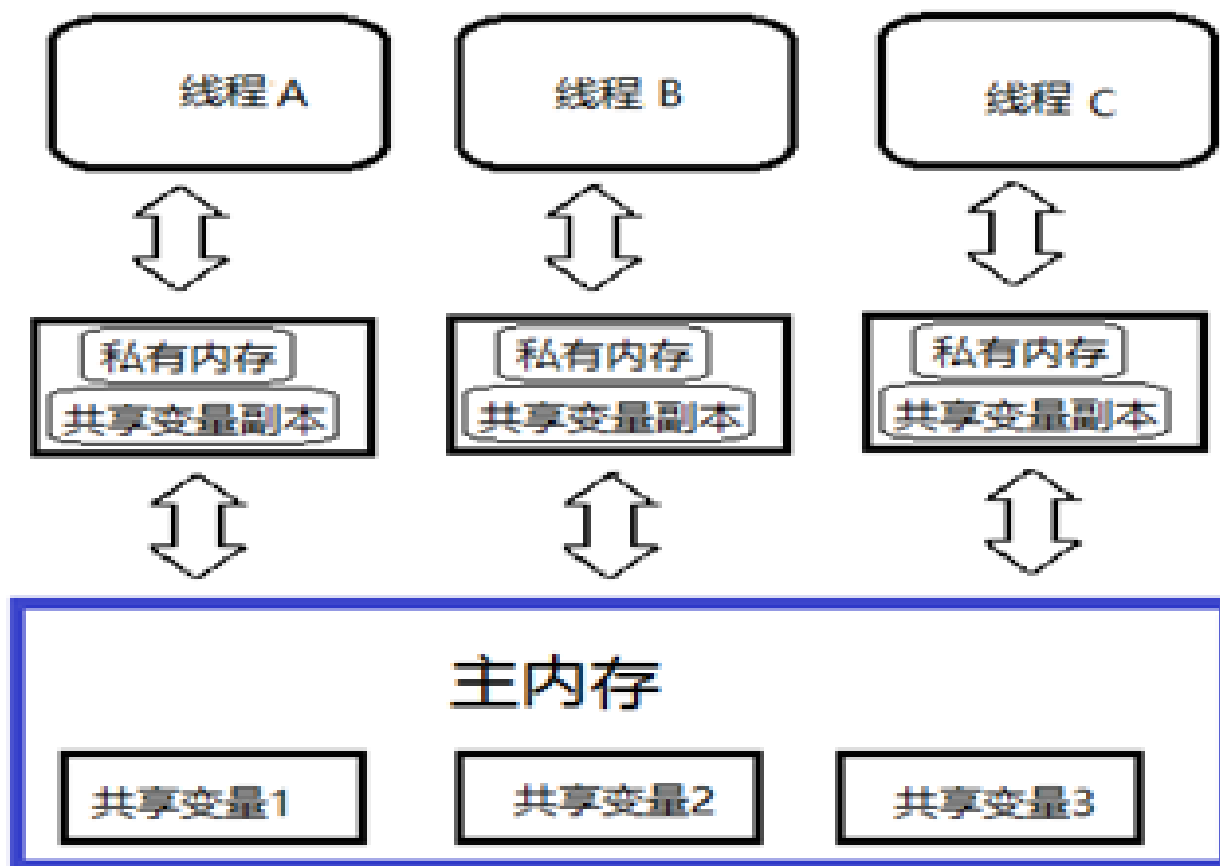
线程-线程的同步



东北林业大学

NORTHEAST FORESTRY UNIVERSITY





○ 线程并发会有什么问题？



在并发编程中存在线程安全问题，主要原因有：

1.存在共享数据

2.多线程共同操作共享数据。

volatile关键词---可见性(Visibility)

volatile关键字的作用很简单，就是一个线程在对主内存的某一份数据进行更改时，改完之后会立刻刷新到主内存。并且会强制让缓存了该变量的线程中的数据清空，必须从主内存重新读取最新数据。这样一来就保证了可见性。

P1

int a=10

T1

a=a+1;

读取a的值

a+1

写入a的值

a:

做临界资源

**不能同时被
两个线程操
作**

(原子性)

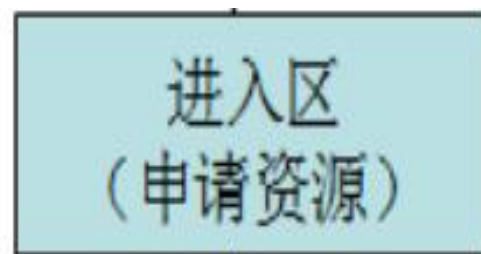
T2

a=a+1;

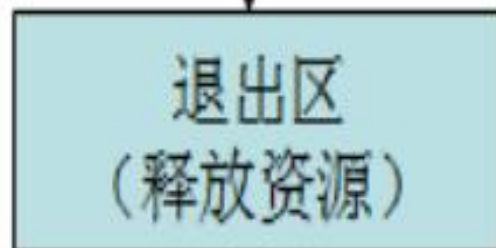
读取a的值

a+1

写入a的值



资源被占有，暂时不能访问，进入阻塞等待队列



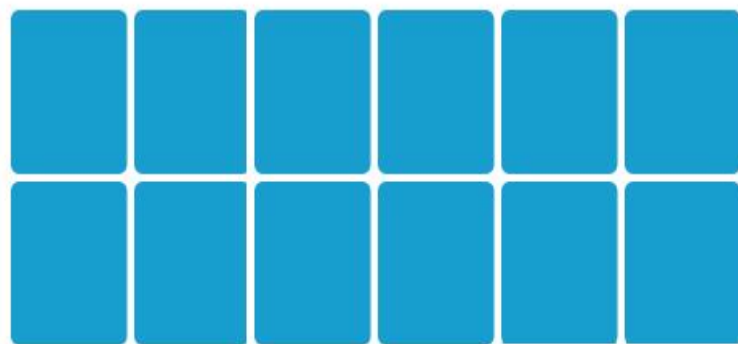
synchronized关键词

synchronized的三种应用方式，Java中每一个对象都可以作为锁，这是synchronized实现同步的基础：

- 1、普通同步方法（实例方法），锁是当前实例对象，进入同步代码前要获得当前实例的锁**
- 2、静态同步方法，锁是当前类的class对象，进入同步代码前要获得当前类对象的锁**
- 3、同步方法块，锁是括号里面的对象，对给定对象加锁，进入同步代码库前要获得给定对象的锁。**

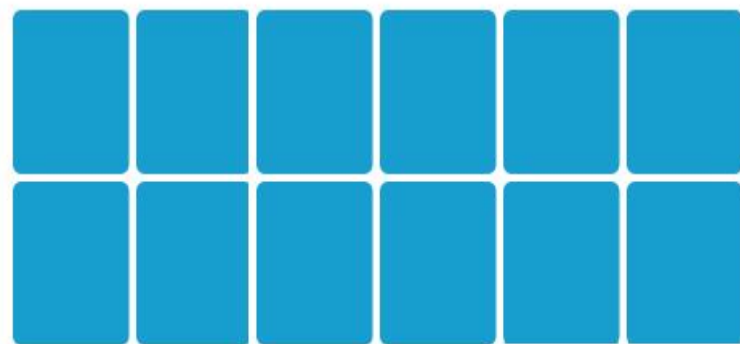
synchronized 的缺点:

- 效率低
- 锁的释放情况少，只在程序正常执行完成和抛出异常时释放锁；
- 试图获得锁是不能设置超时；
- 不能中断一个正在试图获得锁的线程；
- 无法知道是否成功获取到锁；



东北林业大学

NORTHEAST FORESTRY UNIVERSITY



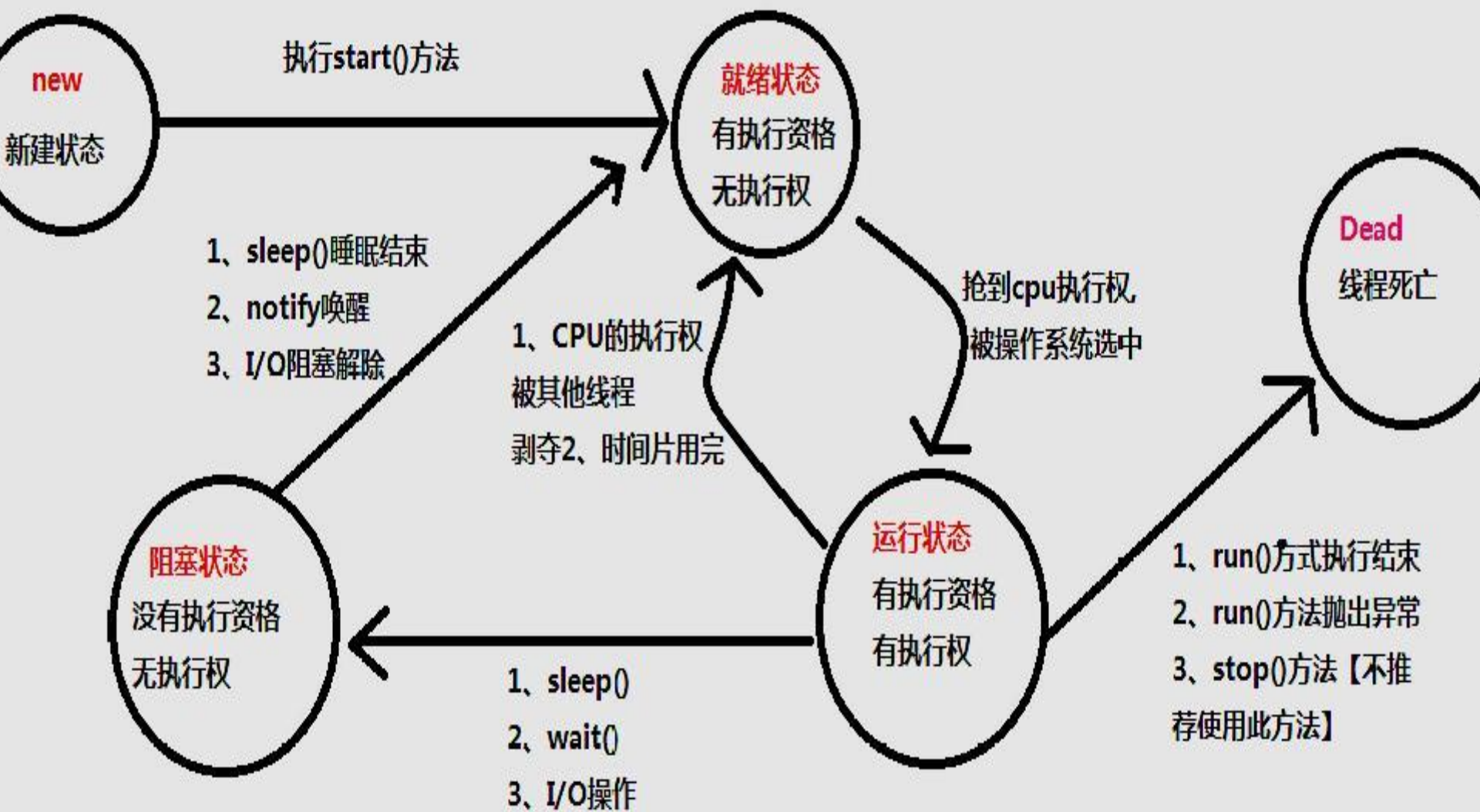
第八章

线程-线程的生命周期



东北林业大学

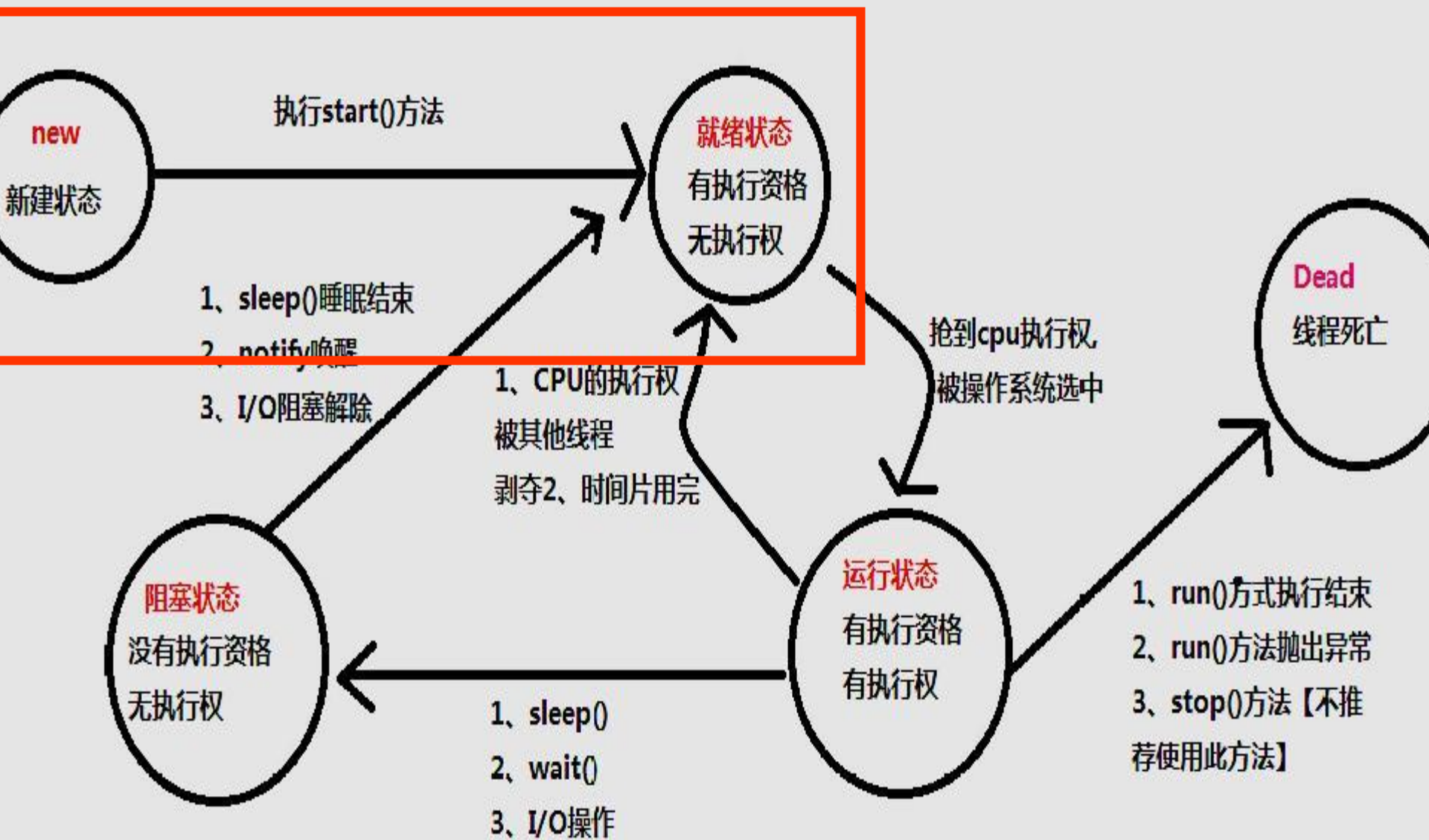
NORTHEAST FORESTRY UNIVERSITY





使用start()

`start()`：创建启动新的线程，可以实现多线程工作。通过`start()`使得线程处于就绪状态，在获得CPU时间片后通过`run()`开始运行。

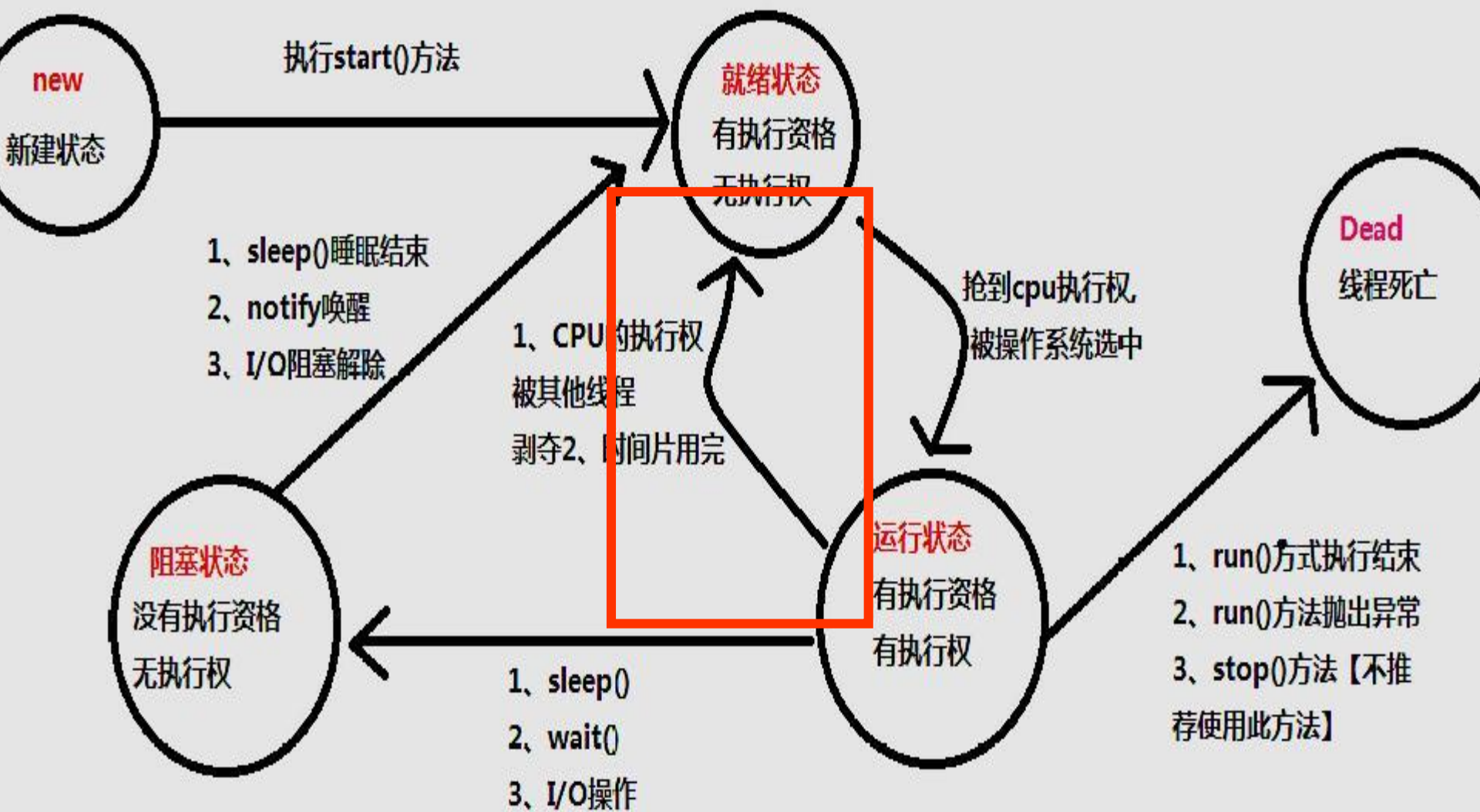





使用yield()

yield() 应该做的是让当前运行线程回到可运行状态，以允许具有相同优先级的其他线程获得运行机会。因此，使用yield()的目的是让相同优先级的线程之间能适当的轮转执行。但是，实际中无法保证yield()达到让步目的，因为让步的线程还有可能被线程调度程序再次选中。

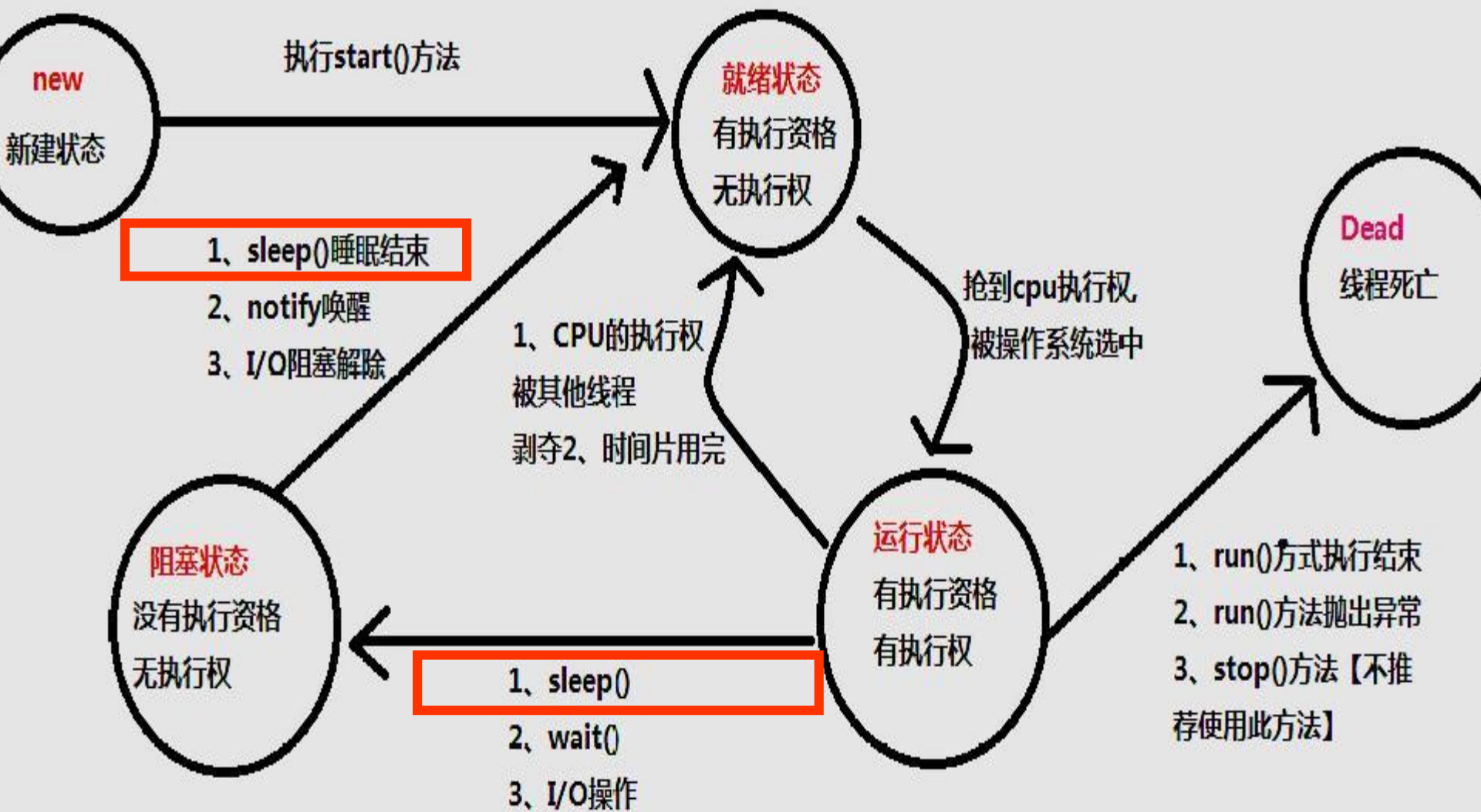
注意：yield() 将导致线程从运行状态转到可运行状态，但有可能没有效果。





使用sleep()

sleep() 方法的作用是在指定的毫秒数内让当前“正在执行的线程”休眠（暂停执行）。等休眠时间结束，从阻塞到就绪状态

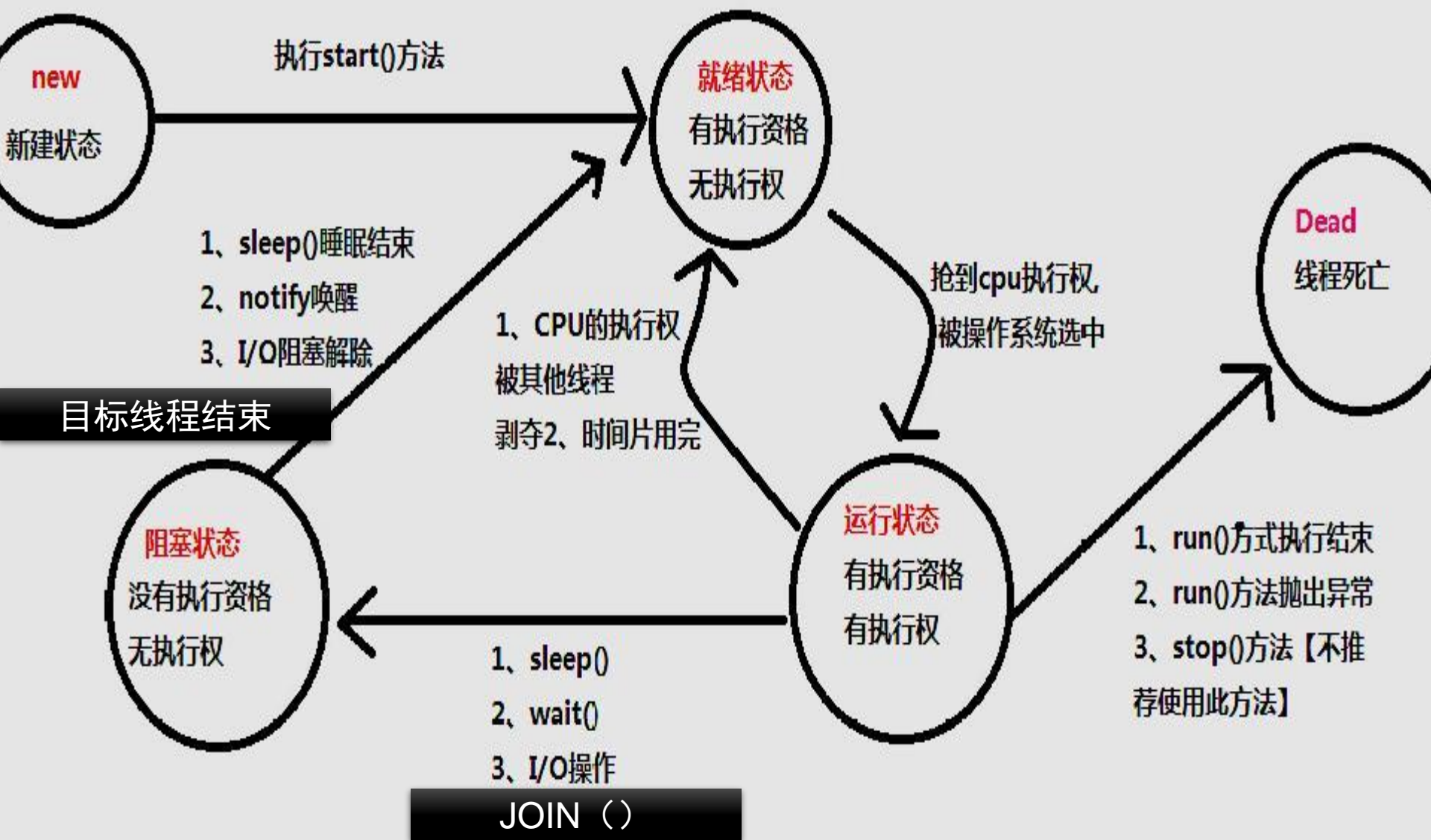




使用join()

继承不适合问题

join()方法的作用，是等待这个线程结束；也就是说，t.join()方法阻塞调用此方法的线程(calling thread)进入 TIMED_WAITING 状态，直到线程t完成，此线程再继续；通常用于在main()主线程内，等待其它线程完成再结束main()主线程。



使用wait(),notify()

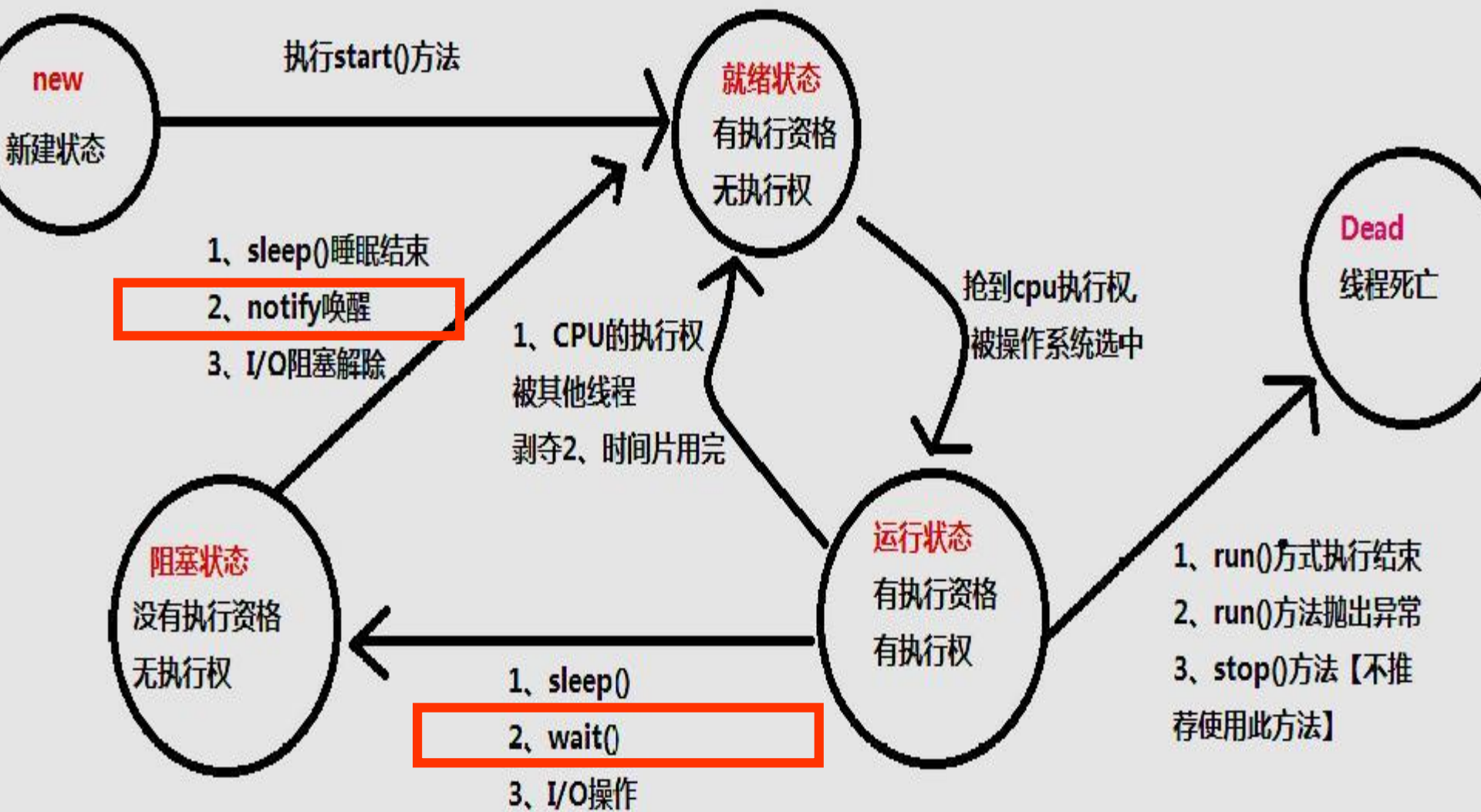
继承不适合问题

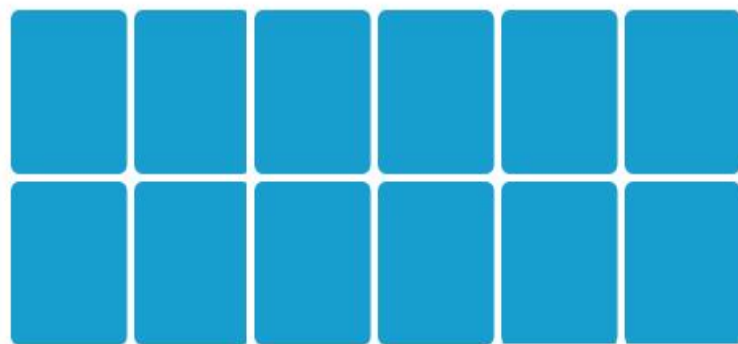
wait() 的作用是让当前线程进入等待状态，同时，wait() 也会让当前线程释放它所持有的锁。

“直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法”，当前线程被唤醒（进入“就绪状态”）

notify() 和 notifyAll() 的作用，则是唤醒当前对象上的等待线程；notify() 是唤醒单个线程，而 notifyAll() 是唤醒所有的线程。

wait(long timeout) 或者超过指定的时间量”，当前线程被唤醒（进入“就绪状态”）。





东北林业大学

NORTHEAST FORESTRY UNIVERSITY