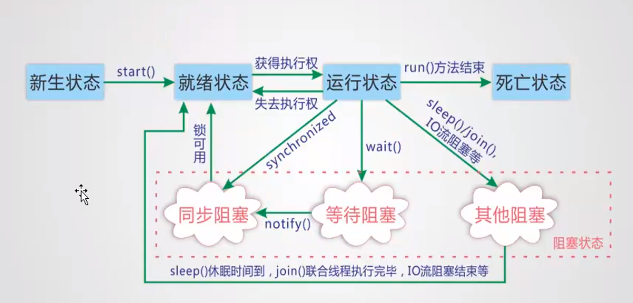
Java多线程：

线程的五大状态：



注意：线程进入阻塞状态后，不能直接回到运行状态，而是先进入就绪状态，等待CPU调度成功之后，在进入运行状态，一个线程进入死亡状态之后，不能再次进入执行，再次执行为新的线程

1. 新生状态：当创建一个线程对象时，线程进入新生状态。
2. 就绪状态：当调用start()方法时，线程进入就绪状态，不代表立即执行，等待cpu的调度成功才会执行，进入就绪状态原因有四个：
3. 调用start()方法
4. 阻塞解除
5. Yield让出调度
6. Jvm算法分配
7. 阻塞状态：不会直接进入运行状态

4种原因：（1）sleep （3）join

(2) wait (4)read, write

4, 死亡状态; (1)运行结束 （2）强制终止

1. 终止状态
2. 线程正常结束，一般线程执行次数较少。
3. 外部干涉，一般线程不能正常结束，stop()/destroy() 不推荐使用，常见的终止方法为加入flag标识。
4. 阻塞

暂停sleep

Sleep(时间)指当前线程阻塞的毫秒数

Sleep存在异常InterruptedException

Sleep时间到达后进入就绪状态

Sleep可以模拟网络延时，计时器等

每一个对象都有一把锁，sleep不会释放锁

礼让yield

让当前正在执行的线程暂停 不是阻塞线程，而是将线程从运行状态转为就绪状态，让cpu重新调度。

Join

合并线程，待此线程执行完毕后，再执行其他线程，其他线程阻塞。Join写在哪个线程，那个线程就会被阻塞。

Java线程Thread提供的五种状态，可以用Thread.State获取，例如：Thread.State state = t.getState()获取线程当前状态，主要由NEW(新生状态) RUNNABLE(就绪或执行状态) TIMED\_WAITING阻塞状态 TERMINATED(终止状态)

线程优先级：java提供一个线程调度器来监控程序启动后进入就绪状态的所有线程，线程调度器按线程的优先级决定应调度哪个线程来执行。（只决定调用概率，不强行指定执行顺序）范围1-10

Thread.MIN\_PRIORITY = 1

Thread.MAX\_PRIORITY = 10

Thread.NORM\_PRIORITY = 5（默认）

建议在start()前调用。