<https://mp.weixin.qq.com/s/K7G9JnuG3HEc2ECVBe52Ow>

### 1、线程的状态

Java中线程中状态可分为五种：

* New（新建状态）
* Runnable（就绪状态）
* Running（运行状态）
* Blocked（阻塞状态）
* Dead（死亡状态）。
* New：新建状态，当线程创建完成时为新建状态，即new Thread(…)，还没有调用start方法时，线程处于新建状态。
* Runnable：就绪状态，当调用线程的的start方法后，线程进入就绪状态，等待CPU资源。处于就绪状态的线程由Java运行时系统的线程调度程序(thread scheduler)来调度。
* Running：运行状态，就绪状态的线程获取到CPU执行权以后进入运行状态，开始执行run方法。
* Blocked：阻塞状态，线程没有执行完，由于某种原因（如，I/O操作等）让出CPU执行权，自身进入阻塞状态。
* Dead：死亡状态，线程执行完成或者执行过程中出现异常，线程就会进入死亡状态。

这五种状态之间的转换关系如下图所示：



Java中是如何实现这几种状态的转换的。

### 2、wait/notify/notifyAll方法的使用

#### 2.1 wait方法

#### 2.2 notify/notifyAll方法

### 3、sleep/yield/join方法解析

这几个方法都位于Thread类中，而上面三个方法都位于Object类中。

#### 3.1 sleep

#### 3.2 yield方法

#### 3.3 join方法

### 4、总结

最后回答一下上面提出的问题：wait/notify/notifyAll方法的作用是实现线程间的协作，那为什么这三个方法不是位于Thread类中，而是位于Object类中？位于Object中，也就相当于所有类都包含这三个方法（因为Java中所有的类都继承自Object类）。

要回答这个问题，还是得回过来看wait方法的实现原理，大家需要明白的是，wait等待的到底是什么东西？如果对上面内容理解的比较好的话，我相信大家应该很容易知道wait等待其实是对象monitor，由于Java中的每一个对象都有一个内置的monitor对象，自然所有的类都理应有wait/notify方法。