# 线程的生命周期

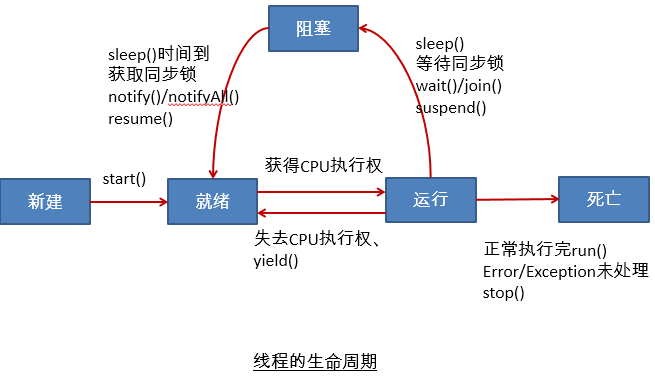
## 六种状态

JDK中用Thread.State枚举表示了线程的几种状态

要想实现多线程，必须在主线程中创建新的线程对象。Java语言使用Thread类及其子类的象来表示线程，在它的一个完整的生命周期中通常要经历如下的五种状态：



## 线程声明周期

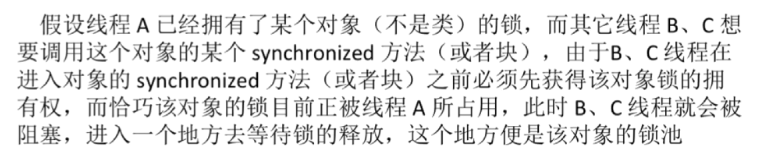


## **阻塞(BLOCKED)状态**

(一). 等待阻塞：运行(running)的线程执行o.wait()方法，JVM会把该线程放入等待队列(waitting queue)中。  
(二). 同步阻塞：运行(running)的线程在获取对象的同步锁时，若该同步锁被别的线程占用，则JVM会把该线程放入锁池(lock pool)中。  
(三). 其他阻塞：运行(running)的线程执行Thread.sleep(long ms)或t.join()方法，或者发出了I/O请求时，JVM会把该线程置为阻塞状态。当sleep()状态超时、join()等待线程终止或者超时、或者I/O处理完毕时，线程重新转入可运行(runnable)状态。

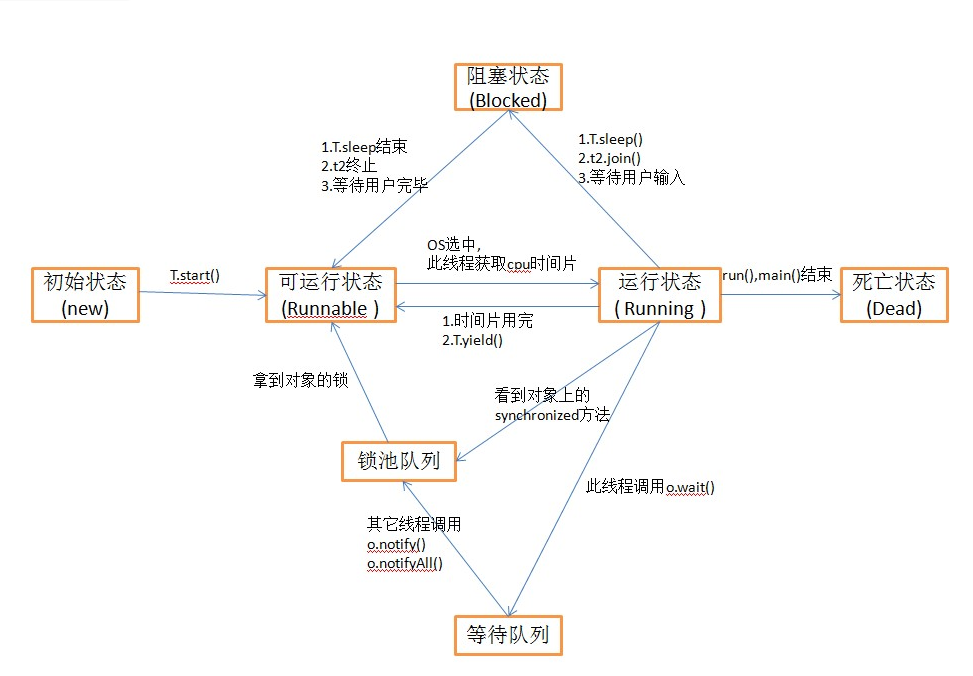
## 锁池和等待池

锁池EntryList



等待池WaitSet

调用进入等待池

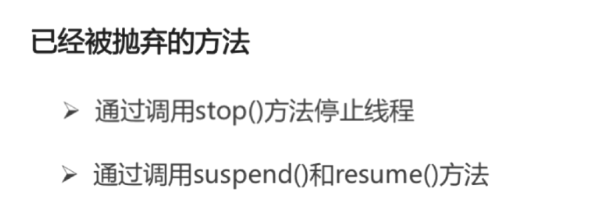


# 线程的使用

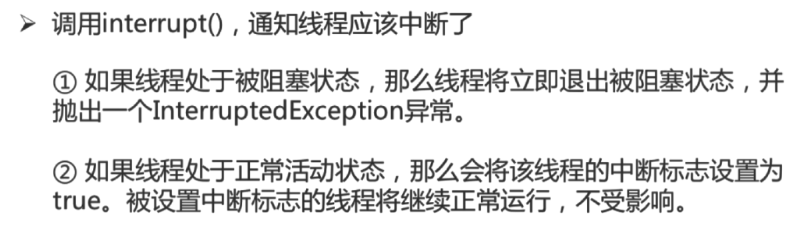
## 停止线程interrupt

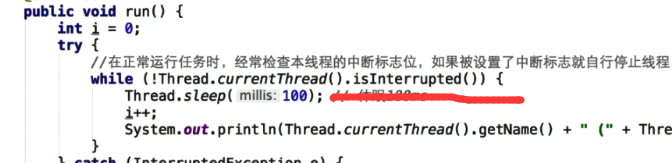
https://blog.csdn.net/jiadajing267/article/details/80137006?depth\_1-utm\_source=distribute.pc\_relevant.none-task&utm\_source=distribute.pc\_relevant.none-task

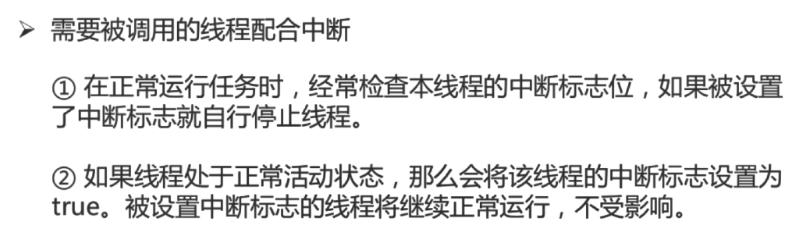
### 过时的



### Interrupt







### 使用标志位

https://www.cnblogs.com/SmilingEye/p/11757192.html