

Java信号量模型实际应用手册Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_Java_E4_BF_A1_E5_8F_B7_c104_645023.htm Java信号量模型需要我们不断的进行学习，在学习的时候会有不少的问题阻碍着我们。下面我们就来看看同步锁模型只是最简单的同步模型。同一时刻，只有一个线程能够运行同步代码。有的时候，我们希望处理更加复杂的同步模型，比如生产者/消费者模型、读写同步模型等。这种情况下，同步锁模型就不够用了。我们需要一个新的模型。这就是我们要讲述的Java信号量模型。Java信号量模型的工作方式如下：线程在运行的过程中，可以主动停下来，等待某个Java信号量模型的通知；这时候，该线程就进入到该信号量的待召（Waiting）队列当中；等到通知之后，再继续运行。很多语言里面，同步锁都由专门的对象表示，对象名通常叫Monitor。同样，在很多语言中，Java信号量模型通常也有专门的对象名来表示，比如，Mutex，Semaphore。Java信号量模型要比同步锁模型复杂许多。一些系统中，信号量甚至可以跨进程进行同步。另外一些信号量甚至还有计数功能，能够控制同时运行的线程数。我们没有必要考虑那么复杂的模型。所有那些复杂的模型，都是最基本的模型衍生出来的。只要掌握了最基本的信号量模型“等待/通知”模型，复杂模型也就迎刃而解了。我们还是以Java语言为例。Java语言里面的同步锁和Java信号量模型概念都非常模糊，没有专门的对象名词来表示同步锁和信号量，只有两个同步锁相关的关键字volatile和synchronized。这种模糊虽然导致概念不清，但同时也避免了Monitor、Mutex、Semaphore等名词

带来的种种误解。我们不必执着于名词之争，可以专注于理解实际的运行原理。在Java语言里面，任何一个Object Reference都可以作为同步锁。同样的道理，任何一个Object Reference也可以作为Java信号量模型。Object对象的wait()方法就是等待通知，Object对象的notify()方法就是发出通知。具体调用方法为

(1) 等待某个Java信号量模型的通知

```
public static final Object signal = new Object(). ... f1() {  
synchronized(singal) { // 首先我们要获取这个信号量。这个信号量同时也是一个同步锁 // 只有成功获取了signal这个信号量兼同步锁之后，我们才可能进入这段代码 signal.wait(). // 这里要放弃信号量。本线程要进入signal信号量的待召 ( Waiting ) 队列 // 可怜。辛辛苦苦争取到手的Java信号量模型，就这么被放弃了 // 等到通知之后，从待召 ( Waiting ) 队列转到就绪 ( Ready ) 队列里面 // 转到了就绪队列中，离CPU核心近了一步，就有机会继续执行下面的代码了。 // 仍然需要把signal同步锁竞争到手，才能够真正继续执行下面的代码。命苦啊。需要注意的是，上述代码中的signal.wait()的意思。signal.wait()很容易导致误解。signal.wait()的意思并不是说，signal开始wait，而是说，运行这段代码的当前线程开始wait这个signal对象，即进入signal对象的待召 ( Waiting ) 队列。
```

(2) 发出某个Java信号量模型的通知

```
... f2() { synchronized(singal) { // 首先，我们同样要获取这个信号量。同时也是一个同步锁。 // 只有成功获取了signal这个信号量兼同步锁之后，我们才可能进入这段代码 signal.notify(). // 这里，我们通知signal的待召队列中的某个线程。 // 如果某个线程等到了这个通知，那个线程就会转到就绪队列中 // 但是本线程仍然继续拥有signal这个同
```

步锁，本线程仍然继续执行 // 嘿嘿，虽然本线程好心通知其他线程， // 但是，本线程可没有那么高风亮节，放弃到手的同步锁 // 本线程继续执行下面的代码 需要注意的是，`signal.notify()`的意思。`signal.notify()`并不是通知`signal`这个对象本身。而是通知正在等待`signal`信号量的其他线程。以上就是`Object`的`wait()`和`notify()`的基本用法。实际上，`wait()`还可以定义等待时间，当线程在某Java信号量模型的待召队列中，等到足够长的时间，就会等无可等，无需再等，自己就从待召队列转移到就绪队列中了。另外，还有一个`notifyAll()`方法，表示通知待召队列里面的所有线程。这些细节问题，并不对大局产生影响。编辑特别推荐: 概括Hibernate查询语言 Java多线程中wait语句的具体使用方法 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com