1.线程休眠sleep();:线程有优先级，但是我们可以用此方法人为的改变它们的优先级，让线程暂停，**它其他线程获得分配空间。**  
用法：Thread.sleep（2000）;//休眠两秒   
2.线程让步yield();**就是让出自己的分配空间给其他线程，**那么问题来了，让步多久呢？如果有两条线程的话，是不是让步到另外一条线程执行完呢？经测试，不是让另外的线程执行，让步时间是不确定的；   
注意：当某个线程调用yield()方法之后，**只有与当前线程优先级形同或者更高的才能获得执行机会。**(直接进入可运行状态，有可能调用完自己又被CPU调度到，又直接变成运行状态了。)  
用法：一般都是指定条件，如if（i==10）{Thread.yield();}   
3.线程插队join（）：当某个程序调用其他线程的join（）时，调用线程将会阻塞，直到插入的线程运行完毕，才运行该线程，如main线程中for(int i = 0; i < 100; i++){if(i == 2){t.join();}}，当i等于2时，线程t将执行完毕再执行main中余下的 i= 3 i=4 ……   
用法：一般都是指定条件，如if（i==10）{t.join();} 注意插**队肯定是在别的线程中插别人的队，不可能在自己的线程中写join()**; 如 ：t线程中写t.join();，这种方法是不正确的。   
4.wait()：当前线程放弃同步锁进入等待状态，直到其他线程进入此同步锁**notify（）或notifyAll（）唤醒线程为止。**5.notify（）：**唤醒锁上等待的第一个调用wait（）方法的线程。**   
6.notifyAll（）：唤醒同步锁上调**用wait（）的所有线程**。   
注意：wait()、notify（）、notifyAll（）必须是同步锁对象。