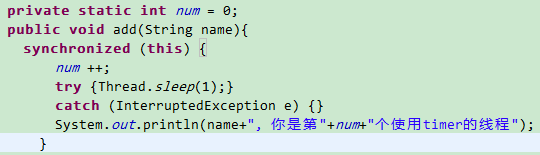
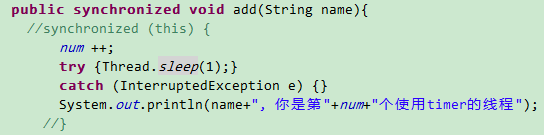
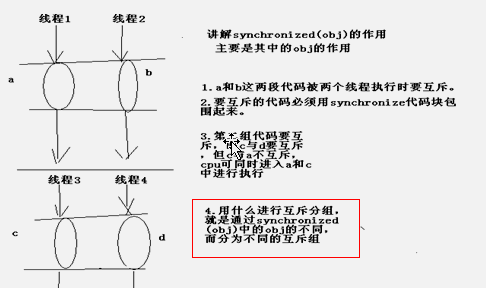
1. 在Java语言中，引入了对象互斥锁的概念，保证共享数据操作的完整性，每个对象都对应于一个可称为“互斥锁”的标记，这个标记保证在任意时刻，只能有一个线程访问该对象的同步方法。
2. 关键词synchronized来表示对象的互斥锁联系，当某个对象synchronized修饰时，表明该对象在任意时刻只能由一个线程来访问(锁住的不是数据，而是存取数据的**方法**)。
3. 关键字synchronized(Object obj){}：锁定某一段代码，执行该段代码时候**锁定obj对象(里面的代码被组合成原子单位)**



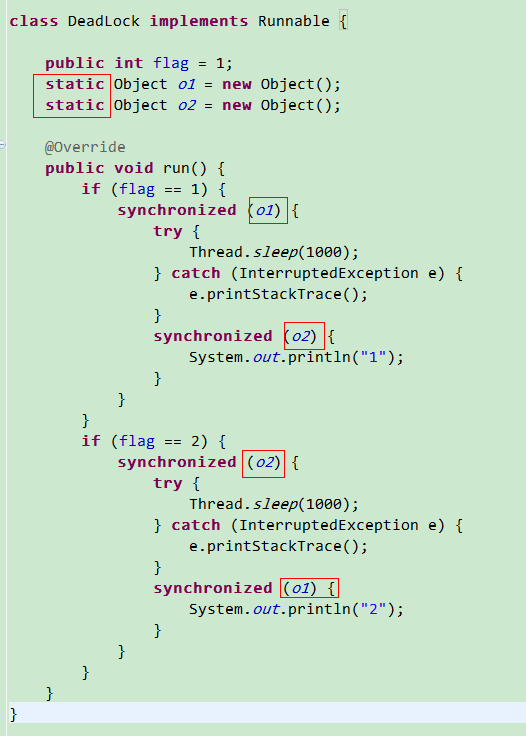
或者采用synchronized关键词(方法内的代码组成了原子单位)修饰整个方法，表示整个方法为同步方法，执行该方法的过程中，**当前对象被锁定，方法内的各种资源被加上了锁，其他同步方法不能访问其中的资源，但是非同步方法可以访问其中的资源。**

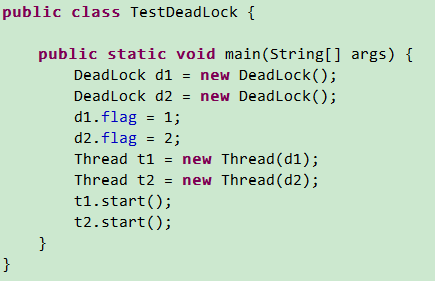




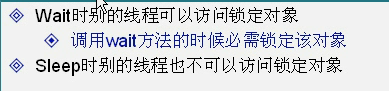
**在方法上采用synchronized关键字修饰，非static方法的同步锁为this，即当前调用对象；static方法的同步锁是当前类的字节码。**

1. 模拟死锁程序

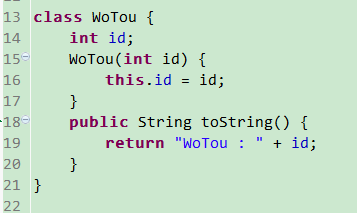




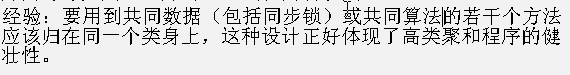
1. 同步化的意义是指定某段工作要在不能分割的状态(**原子状态**)下执行，也就是说单独的操作不重要，重要的是有多个步骤的方法。
2. wait、sleep方法的区别



1. 模拟生产者-消费者问题



1. 经验



1. 例：子线程循环10次，接着主线程循环100，接着又回到子线程循环10次，接着再回到主线程又循环100，如此循环50次，请写出程序。

