

java的多线程同步初探 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/240/2021\\_2022\\_java\\_E7\\_9A\\_84\\_E5\\_A4\\_9A\\_c104\\_240571.htm](https://www.100test.com/kao_ti2020/240/2021_2022_java_E7_9A_84_E5_A4_9A_c104_240571.htm) java在语言层级实现的多线程操作的一些原语，用起来比较方便，不像C、C++之类的要调用操作系统功能才能实现多线程。这里展示一个样例show一下java多线程就是这样简单，一点也不神秘，java让在课本上经常讲的多任务多线程处理很方便的走到初学者的代码中来。（但是也很方便创造deadlock，死锁很难发现，如果不了解就最好不要用，不是必要场合不要用）。至于书上云的一些线程状态以及各状态间的转换则都是一样的，java多了一个线程组，便于以group的方式管理大量线程，更多介绍还是去看API或者google，baidu吧，记录之供参考。如果你能够自己写一遍生产者消费者（也有称读者写者）就有了3、5成功力了吧，那样才能够考虑把它用到你的production代码中去。主要的就是一个Thread类，Thread类拥有一个Run（）方法，还有一个Thread.start()来启动线程开工。实现多线程有两个方法：  
1.将要并行执行的操作封装到一个继承自Thread的类中去  

```
public class thread_ex extends Thread...{ public void init() ...{ } public void start() ...{ /**will call run() default. } public void stop() ...{ } public void run()...{ /**do some parallelization works. }}
```

  
2.延伸Runnable接口,满足一些继承了其他类又想多线程的（因为java不像C++一样支持多继承）  

```
public class ThreadTest extends FatherClass implements Runnable public void init() .....{ } public void start() .....{ /**will call run() default. } public void stop() .....{ } public void run().....{ /**do some parallelization works. }}
```

与线程

有关的几个关键词是很重要的：synchronize wait notify notifyAll 如果不能很好的理解他们还是不要用的好。简单的介绍如下

synchronize 修饰定义一个对象或者方法或者属性的访问必须串行化访问，顾名思义像在银行取钱要排队一样，那个业务员就是一个synchronize 对象。只有像我等无钱之辈才去排队，那些个VIP自然不用了。wait 和notify notifyAll是一家的，他们是对象级的锁，余天生驽钝，这个偶翻了许多资料才理解，如果不能理解就看代码吧。多线程中的一个去访问共享资源发现有人使用，于是就在这个资源上面wait，占用者如果使用完了就notify（）一下通知下一个等待者可以进来了，notifyAll（）就是说在外面等的一起上吧！可是只能服务与一个人，于是大家抢着进去，力气大的自然占便宜。可是在java世界里面JVM说了算。有wait和notify的地方一定有synchronize，反过来不成立，wait 和notify不在Thread的势力范围

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](http://www.100test.com)