

Java语言规范线程形式范例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_E8_AF_AD_E8_A8_80_c104_645022.htm 在一开始接触Java的时候我们没有很注意Java语言规范，其实这是不对的。下面我们就来看看开始执行Java语言规范程序后，至少会有一个线程开始操作，有操作的是被称为主线程的线程，主线程执行输入类的main()。当main()里的所有处理均结束后，则主线程也同时结束。当应用程序的规模大到一定程度，程序里的多线程会以每种形式存在。以下是几个常见的范例：1) GUI应用程序。2) 比较花费时间的I/O处理。3) 多个客户端。如欲启动线程时，有下列两种方法：1) 利用Thread类的子类的实例，启动线程。2) 利用Runnable接口的实现类的实例，启动线程。要注意的是，“Thread的实例”和“线程本身”是两个不同的部分。即使建立了Thread的实例，也还没有启动线程，而且就算线程已经结束，Thread实例也不会就这样消失。建立一个实现Runnable接口的类，将该类的实例传给Thread的构造函数，调用start()...，这就是利用Runnable接口来启动线程的方法。记住：无论哪种方式，启动新线程的方法永远是Thread类的start()。利用Thread类的sleep()可以暂停线程的执行。执行下面的语句时，即可让当前的线程（执行此语句的线程）暂时停止越1000ms。Thread.sleep(1000)。当实例方法加上关键字synchronized声明之后，就可以只让一个线程操作某类实例的这类方法。“让一个线程操作”并不是说只能让某一个特定的线程操作而已，而是指一次只能让一个线程执行。这种方法称为synchronized方法（同步方法）。

synchronized实例方法是使用this锁定去做线程的共享互斥。
。synchronized类方法是使用该类的类对象的锁定去做线程的共享互斥。Java语言规范线程的协调（这三种方法执行的前提是执行线程手中有obj的锁）：
1) obj.wait()是把现在的线程放到obj的wait set；
2) obj.notify()是从obj的wait set里唤醒一个线程；
3) obj.notifyAll()是唤醒所有在obj的wait set里的线程。
被唤醒的线程只是处于等锁状态（但已经不再wait set中了），当时的obj的锁还掌握在执行notify()或notifyAll()的线程手上。
当唤醒的线程得到锁之后，将从wait()之后继续执行。
。Introduction 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