

二级Java考试辅导教程：6.1线程简介[3] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022__E4_BA_8C_E7_BA_A7Java_c97_138767.htm 仔细分析一下运行结果，会发现两个线程是交错运行的，感觉就象是两个线程在同时运行。但是实际上一台计算机通常就只有一个CPU，在某个时刻只能是只有一个线程在运行，而java语言在设计时就充分考虑到线程的并发调度执行。对于程序员来说，在编程时要注意给每个线程执行的时间和机会，主要是通过让线程睡眠的办法（调用sleep()方法）来让当前线程暂停执行，然后由其它线程来争夺执行的机会。如果上面的程序中没有用到sleep()方法，则就是第一个线程先执行完毕，然后第二个线程再执行完毕。所以用活sleep()方法是学习线程的一个关键。构造线程体的两种方法的比较：1. 使用Runnable接口 1) 可以将CPU，代码和数据分开，形成清晰的模型. 2) 还可以从其他类继承. 3) 保持程序风格的一致性。 2. 直接继承Thread类 1) 不能再从其他类继承. 2) 编写简单，可以直接操纵线程，无需使用Thread.currentThread()。 6.1.3 线程的调度 Java提供一个线程调度器来监控程序中启动后进入就绪状态的所有线程。线程调度器按照线程的优先级决定应调度哪些线程来执行。来源：www.examda.com 线程调度器按线程的优先级高低选择高优先级线程（进入运行中状态）执行，同时线程调度是抢先式调度，即如果在当前线程执行过程中，一个更高优先级的线程进入可运行状态，则这个线程立即被调度执行。抢先式调度又分为：时间片方式和独占方式。在时间片方式下，当前活动线程执行完当前时间片后，如果有其他处于就绪状态

的相同优先级的线程，系统会将执行权交给其他就绪态的同优先级线程；当前活动线程转入等待执行队列，等待下一个时间片的调度。在独占方式下，当前活动线程一旦获得执行权，将一直执行下去，直到执行完毕或由于某种原因主动放弃CPU，或者是有一高优先级的线程处于就绪状态。下面几种情况下，当前线程会放弃CPU: 1. 线程调用了yield()或sleep()方法主动放弃；2. 由于当前线程进行I/O访问，外存读写，等待用户输入等操作，导致线程阻塞；或者是为等候一个条件变量，以及线程调用wait()方法；3. 抢先式系统下，由高优先级的线程参与调度；时间片方式下，当前时间片用完，由同优先级的线程参与调度。线程的优先级用数字来表示，范围从1到10，即Thread.MIN_PRIORITY到Thread.MAX_PRIORITY。一个线程的缺省优先级是5，即Thread.NORM_PRIORITY。下述方法可以对优先级进行操作：`int getPriority()` //得到线程的优先级
`void setPriority(int newPriority)` //当线程被创建后，可通过此方法改变线程的优先级

例6.3中生成三个不同线程，其中一个线程在最低优先级下运行，而另两个线程在最高优先级下运行。

6.1.4基本的线程控制

1.终止线程

线程终止后，其生命周期结束了，即进入死亡态，终止后的线程不能再被调度执行，以下几种情况，线程进入终止状态：1) 线程执行完其run()方法后，会自然终止。2) 通过调用线程的实例方法stop()来终止线程。

2.测试线程状态

可以通过Thread中的isAlive()方法来获取线程是否处于活动状态；线程由start()方法启动后，直到其被终止之间的任何时刻，都处于'Alive'状态。

3.线程的暂停和恢复

有几种方法可以暂停一个线程的执行，在适当

的时候再恢复其执行。 1) sleep() 方法 当前线程睡眠（停止执行）若干毫秒，线程由运行中状态进入不可运行状态，停止执行时间到后线程进入可运行状态。 2) suspend()和resume() 方法 线程的暂停和恢复，通过调用线程的suspend()方法使线程暂时由可运行态切换到不可运行态，若此线程想再回到可运行态，必须由其他线程调用resume()方法来实现。 注：从JDK1.2开始就不再使用suspend()和resume()。 3) join() 当前线程等待调用该方法的线程结束后,再恢复执行. TimerThread tt=new TimerThread(100). tt.start(). ... public void timeout(){ tt.join().// 当前线程等待线程tt 执行完后再继续往下执行 ... }

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