

等级考试二级JAVA考点分析之线程与对象串行化 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__E7_AD_89_E7_BA_A7_E8_80_83_E8_c97_138694.htm 8.1线程的概念 考点

：1 线程的概念 1. 线程 (1)线程是由表示程序运行状态的寄存器(包括程序计数器和堆栈)组成的。(2)线程是程序执行过程中的某一时刻的状态。(3)线程是一个用户级的实体，在内存中驻留在普通用户级方法可以直接访问的区域。(4)每个线程都是一个能够独立执行自身指令的控制流程。(5)操作系统通过对多线程的调度实现线程的并发执行。(6)线程本身并不是一个程序而是运行于一个程序或者进程中。(7)线程是个动态的概念，有其自身的产生、存在和消亡的过程。(8)Java通过在程序中提供多线程支持来提高线程的安全性。(9)线程是程序中的一个单个执行流，多线程是程序中的多个执行流。

2. 程序、线程和进程 (1)程序是一段静态的代码、是应用软件执行的蓝本。(2)进程是程序的一次动态执行过程，包括程序加载、执行到结束。(3)线程是比进程更小的执行单位，一个进程可以包含多个线程。(4)进程是由代码、数据、内核状态和一组寄存器组成的。(5)线程不包含进程的地址空间中的代码和数据。

考点：2 Java语言中的线程 (1)Java语言中的线程包括3部分：虚拟CPU、该CPU执行的代码，以及代码所操作的数据。(2)Java语言中代码和数据相互独立，代码可以在不同的线程之间共享，同样数据也可以在不同的线程之间共享。

(3) java.lang.Thread类定义了Java语言中的线程模型，用户可以通过该类创建、定义和控制自己的线程。

8.2线程的创建 考点：3 线程的创建 在Java中，通过调用Thread类的构造

方法来创建线程对象，线程执行的入口方法是run()，可以有两种提供run()方法的实现方式。

1. 通过实现Runnable接口创建过程Thread类使用一个Runnable的实例作为其构造方法，该实例对象提供了线程体run()，线程从该run()方法开始执行。新建的线程必须通过调用线程的start()方法才能运行。
2. 通过继承Thread类 Thread类本身实现了Runnable接口，因此Thread类含有run()方法。通过继承Thread类，并且重写其run()方法，来创建线程。

以上两种方法的比较如下：(1) 实现Runnable接口的方法符合面向对象的思想，实现了Runnable接口的类，可以很容易地继承其他的类。一般都提倡使用这种方法。(2) 继承Thread类的方法，比较简单，可以直接调用线程的方法。可以视具体应用而定。

8.3 线程的调度与线程控制

考点：4 线程优先级和调度策略 线程的优先级，在Thread类中提供了3个静态常量：MIN_PRIORITY = 1、MAX_PRIORITY = 10、NORM_PRIORITY = 5来控制线程的优先级，数值越大优先级越高。子线程继承父线程的优先级，主线程具有普通优先级。可以通过getPriority()方法获取线程的优先级，通过setPriority()设定线程的优先级。

线程调度是指：在单个CPU上以一定策略控制多个线程的执行，这种策略是抢占式调度，简单地说，就是指高优先级的线程首先运行，低优先级的线程被高优先级线程抢占执行。在Java中，系统按照优先级的不同设置不同的线程等待池，首先运行高优先级等待池的线程，然后再运行低优先级等待池的线程。

考点：5 线程的控制 要控制线程，可以使用Thread类提供的如下方法：

- (1) sleep()，一个线程通过调用此方法暂停运行线程一定时间
- (2) yield()，一个线程通过调用此方法使其具有

相同的优先级的线程有运行的机会。(3)join()，若在当前线程中执行th.join()方法，则当前线程暂停运行，等待另一线程th运行结束，才恢复到可运行状态。(4)interrupt()，调用该方法可使线程从阻塞状态中恢复。(5)currentThread()，该方法是静态方法，用于返回当前线程的引用。(6)isAlive()，用于判断线程时候还处于活动状态。(7)stop()，强行终止线程。(8)suspend()和resume()，前者用于暂停一个线程，后者用于恢复该线程。

8.4 线程的同步

考点：6 对象的加锁及其操作

多线程并发操作时，由于各线程对共享数据的操作顺序不同，会影响程序的执行结果，因此Java语言使传统的加锁技术对共享数据的操作进行并发控制。

1. 对象加锁及其操作

临界区是指程序中的一个代码段，在这段代码中，单独并发的线程对同一个对象进行访问。在Java中，用关键字“synchronized”标识一个临界区。对象锁是指Java为synchronized(Object)语句指定的对象进行加锁，对象锁是独占排他锁。

2. 对象加锁的注意事项

使用对象锁需要注意如下几点：

- (1)释放对象锁的情况：synchronized()语句块执行完，synchronized()语句块出现异常、持有锁的线程调用该对象的wait()方法。
- (2)所有被访问的共享数据及访问代码都必须作为临界区，用synchronized加锁。

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