

二级Java考试辅导教程：6.1线程简介[1] 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_138768.htm

6.1 线程简介 随着计算机的飞速发展，个人计算机上的操作系统也纷纷采用多任务和分时设计，将早期只有大型计算机才具有的系统特性带到了个人计算机系统中。一般可以在同一时间内执行多个程序的操作系统都有进程的概念。一个进程就是一个执行中的程序，而每一个进程都有自己独立的一块内存空间、一组系统资源。在进程概念中，每一个进程的内部数据和状态都是完全独立的。Java程序通过流控制来执行程序流，程序中单个顺序的流控制称为线程，多线程则指的是在单个程序中可以同时运行多个不同的线程，执行不同的任务。多线程意味着一个程序的多行语句可以看上去几乎在同一时间内同时运行。线程与进程相似，是一段完成某个特定功能的代码，是程序中单个顺序的流控制；但与进程不同的是，同类的多个线程是共享一块内存空间和一组系统资源，而线程本身的数据通常只有微处理器的寄存器数据，以及一个供程序执行时使用的堆栈。所以系统在产生一个线程，或者在各个线程之间切换时，负担要比进程小的多，正因如此，线程被称为轻负荷进程（light-weight process）。一个进程中可以包含多个线程。一个线程是一个程序内部的顺序控制流。来源

：www.examda.com

1. 进程：每个进程都有独立的代码和数据空间（进程上下文），进程切换的开销大。
2. 线程：轻量的进程，同一类线程共享代码和数据空间，每个线程有独立的运行栈和程序计数器（PC），线程切换的开销小。
3. 多进程

：在操作系统中，能同时运行多个任务程序。 4. 多线程：在同一应用程序中，有多个顺序流同时执行。 6.1.1 线程的概念模型 Java内在支持多线程，它的所有类都是在多线程下定义的，Java利用多线程使整个系统成为异步系统。Java中的线程由三部分组成，如图6.1所示。 1. 虚拟的CPU，封装在java.lang.Thread类中。 2. CPU所执行的代码，传递给Thread类。 3. CPU所处理的数据，传递给Thread类。 6.1.2 线程体(1) Java的线程是通过java.lang.Thread类来实现的。当我们生成一个Thread类的对象之后,一个新的线程就产生了。此线程实例表示Java解释器中的真正的线程，通过它可以启动线程、终止线程、线程挂起等，每个线程都是通过类Thread在Java的软件包Java.lang中定义，它的构造方法为：`public Thread (ThreadGroup group , Runnable target , String name)`；其中，`group`指明该线程所属的线程组；`target`实际执行线程体的目标对象，它必须实现接口Runnable；`name`为线程名。Java中的每个线程都有自己的名称，Java提供了不同Thread类构造器，允许给线程指定名称。如果`name`为null时，则Java自动提供唯一的名称。当上述构造方法的某个参数为null时，我们可得到下面的几个构造方法：`public Thread ()`；`public Thread (Runnable target)`；`public Thread (Runnable target , String name)`；`public Thread (String name)`；`public Thread (ThreadGroup group , Runnable target)`；`public Thread (ThreadGroup group , String name)`；一个类声明实现Runnable接口就可以充当线程体，在接口Runnable中只定义了一个方法`run ()`：`public void run ()`；任何实现接口Runnable的对象都可以作为一个线程的目标对象，类Thread

本身也实现了接口Runnable，因此我们可以通过两种方法实现线程体。（一）定义一个线程类，它继承线程类Thread并重写其中的方法run（），这时在初始化这个类的实例时，目标target可为null，表示由这个实例来执行线程体。由于Java只支持单重继承，用这种方法定义的类不能再继承其它父类。（二）提供一个实现接口Runnable的类作为一个线程的目标对象，在初始化一个Thread类或者Thread子类的线程对象时，把目标对象传递给这个线程实例，由该目标对象提供线程体run（）。这时，实现接口Runnable的类仍然可以继承其它父类。每个线程都是通过某个特定Thread对象的方法run()来完成其操作的，方法run()称为线程体。图6.2表示了java线程的不同状态以及状态之间转换所调用的方法。

1. 创建状态(new Thread) 执行下列语句时，线程就处于创建状态：
Thread myThread = new MyThreadClass(). 当一个线程处于创建状态时，它仅仅是一个空的线程对象，系统不为它分配资源。
2. 可运行状态(Runnable) Thread myThread = new MyThreadClass(). myThread.start(). 当一个线程处于可运行状态时，系统为这个线程分配了它需的系统资源，安排其运行并调用线程运行方法，这样就使得该线程处于可运行(Runnable)状态。需要注意的是这一状态并不是运行中状态(Running)，因为线程也许实际上并未真正运行。由于很多计算机都是单处理器的，所以要在同一时刻运行所有的处于可运行状态的线程是不可能的，Java的运行系统必须实现调度来保证这些线程共享处理器。

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