

Java多线程程序设计详细解析 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_Java\\_E5\\_A4\\_9A\\_E7\\_BA\\_BF\\_c104\\_252433.htm](https://www.100test.com/kao_ti2020/252/2021_2022_Java_E5_A4_9A_E7_BA_BF_c104_252433.htm)

一、理解多线程 多线程是这样一种机制，它允许在程序中并发执行多个指令流，每个指令流都称为一个线程，彼此间互相独立。线程又称为轻量级进程，它和进程一样拥有独立的执行控制，由操作系统负责调度，区别在于线程没有独立的存储空间，而是和所属进程中的其它线程共享一个存储空间，这使得线程间的通信远较进程简单。多个线程的执行是并发的，也就是在逻辑上“同时”，而不管是否是物理上的“同时”。如果系统只有一个CPU，那么真正的“同时”是不可能的，但是由于CPU的速度非常快，用户感觉不到其中的区别，因此我们也不用关心它，只需要设想各个线程是同时执行即可。多线程和传统的单线程在程序设计上最大的区别在于，由于各个线程的控制流彼此独立，使得各个线程之间的代码是乱序执行的，由此带来的线程调度，同步等问题，将在以后探讨。

二、在Java中实现多线程 我们不妨设想，为了创建一个新的线程，我们需要做些什么？很显然，我们必须指明这个线程所要执行的代码，而这就是在Java中实现多线程我们所需要做的一切！真是神奇！Java是如何做到这一点的？通过类！作为一个完全面向对象的语言，Java提供了类Java.lang.Thread来方便多线程编程，这个类提供了大量的方法来方便我们控制自己的各个线程，我们以后的讨论都将围绕这个类进行。那么如何提供给Java我们要线程执行的代码呢？让我们来看一看Thread类。Thread类最重要的方法是run()，它为Thread类的

方法start()所调用，提供我们的线程所要执行的代码。为了指定我们自己的代码，只需要覆盖它！方法一：继承 Thread 类，覆盖方法 run()，我们在创建的 Thread 类的子类中重写 run()，加入线程所要执行的代码即可。下面是一个例子：

```
public class MyThread extends Thread { int count= 1, number.  
public MyThread(int num) { number = num. System.out.println ("  
创建线程 " number). } public void run() { while(true) {  
System.out.println ("线程 " number ":计数 " count). if( count== 6)  
return. } } public static void main(String args[]) { for(int i = 0. i    5.  
i ) new MyThread(i 1).start(). } }
```

这种方法简单明了，符合大家的习惯，但是，它也有一个很大的缺点，那就是如果我们的类已经从一个类继承（如小程序必须继承自 Applet 类），则无法再继承 Thread 类，这时如果我们又不想建立一个新的类，应该怎么办呢？我们不妨来探索一种新的方法：我们不创建 Thread 类的子类，而是直接使用它，那么我们只能将我们的方法作为参数传递给 Thread 类的实例，有点类似回调函数。但是 Java 没有指针，我们只能传递一个包含这个方法的类的实例。那么如何限制这个类必须包含这一方法呢？当然是使用接口！（虽然抽象类也可满足，但是需要继承，而我们之所以要采用这种新方法，不就是为了避免继承带来的限制吗？）Java 提供了接口 `java.lang.Runnable` 来支持这种方法。

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)