

Java多线程程序设计入门PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E5_A4_9A_E7_BA_BF_c104_144361.htm 在Java语言产生前，传统的程序设计语言的程序同一时刻只能单任务操作，效率非常低，例如程序往往在接收数据输入时发生阻塞，只有等到程序获得数据后才能继续运行。随着Internet的迅猛发展，这种状况越来越不能让人们忍受：如果网络接收数据阻塞，后台程序就处于等待状态而不继续任何操作，而这种阻塞是经常会碰到的，此时CPU资源被白白的闲置起来。如果在后台程序中能够同时处理多个任务，该多好啊！应Internet技术而生的Java语言解决了这个问题，多线程程序是Java语言的一个很重要的特点。在一个Java程序中，我们可以同时并行运行多个相对独立的线程，例如，我们如果创建一个线程来进行数据输入输出，而创建另一个线程在后台进行其它的数据处理，如果输入输出线程在接收数据时阻塞，而处理数据的线程仍然在运行。多线程程序设计大大提高了程序执行效率和处理能力。

线程的创建 我们知道Java是面向对象的程序语言，用Java进行程序设计就是设计和使用类，Java为我们提供了线程类Thread来创建线程，创建线程与创建普通的类的对象的操作是一样的，而线程就是Thread类或其子类的实例对象。下面是一个创建启动一个线程的语句：

```
Thread thread1=new Thread(). file://声明一个对象实例，即创建一个线程；  
Thread1.run(). file://用Thread类中的run()方法启动线程；
```

从这个例子，我们可以通过Thread()构造方法创建一个线程，并启动该线程。事实上，启动线程，也就是启动线程的run()方法

，而Thread类中的run()方法没有任何操作语句，所以这个线程没有任何操作。要使线程实现预定功能，必须定义自己的run()方法。Java中通常有两种方式定义run()方法：通过定义一个Thread类的子类，在该子类中重写run()方法。Thread子类的实例对象就是一个线程，显然，该线程有我们自己设计的线程体run()方法，启动线程就启动了子类中重写的run()方法。通过Runnable接口，在该接口中定义run()方法的接口。所谓接口跟类非常类似，主要用来实现特殊功能，如复杂关系的多重继承功能。在此，我们定义一个实现Runnable()接口的类，在该类中定义自己的run()方法，然后以该类的实例对象为参数调用Thread类的构造方法来创建一个线程。线程被实际创建后处于待命状态，激活（启动）线程就是启动线程的run()方法，这是通过调用线程的start()方法来实现的。下面一个例子实践了如何通过上述两种方法创建线程并启动它们：

```
// 通过Thread类的子类创建的线程； class thread1
extends Thread { file://自定义线程的run()方法； public void
run() { System.out.println("Thread1 is running..."). } } file://通
过Runnable接口创建的另外一个线程； class thread2 implements
Runnable { file://自定义线程的run()方法； public void run() {
System.out.println("Thread2 is running..."). } } file://程序的主类
class Multi_Thread file://声明主类； { plubic static void main(String
args[]) file://声明主方法； { thread1 threadone=new thread1().
file://用Thread类的子类创建线程； Thread threaddtwo=new
Thread(new thread2()). file://用Runnable接口类的对象创建线程
； threadone.start(). threaddtwo.start(). file://strat()方法启动线程
； } } 运行该程序就可以看出，线程threadone和threaddtwo交替
```

占用CPU，处于并行运行状态。可以看出，启动线程的run()方法是通过调用线程的start()方法来实现的(见上例中主类)，调用start()方法启动线程的run()方法不同于一般的调用方法，调用一般方法时，必须等到一般方法执行完毕才能够返回start()方法，而启动线程的run()方法后，start()告诉系统该线程准备就绪可以启动run()方法后，就返回start()方法执行调用start()方法语句下面的语句，这时run()方法可能还在运行，这样，线程的启动和运行并行进行，实现了多任务操作。

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