

java初学者实践教程25 - 多线程 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/244/2021_2022_java_E5_88_9D_E5_AD_A6_c104_244682.htm Java语言中有一个重要的特性是支持多线程。多线程是java的一项高级技术，它涉及到操作系统里面的知识，层次贴近系统层面。对于普通程序员一般很少碰它。而且目前就是在java EE（原来的J2EE）的相关框架里，对线程这个东西都是尽量回避。程序员最理想的状态是专注业务逻辑，而不是天天想着线程这个东西怎么写。思考一个问题程序的本质是什么？是CPU的指令序列的集合。到底什么顺序是程序员编写的让计算机赋值，它就赋值、写个循环它就循环、写个分支语句它就分支、写个跳转它就跳转。每个指令流就是一个线程，并发执行多个指令流就是多线程。大家想，只有一个CPU怎么可能同时发出多个指令流呢？是的，并发只是“逻辑”上的，物理上是不可能的除非是两个以上的CPU. 多线程和传统的单线程的区别是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，出现了并发访问带来的一切问题。正像是三个和尚的故事，和尚多了未必是好事。也就是刚才说的，程序员一般都不让他们碰这个东西。在java中如何写线程呢，在java中就是很简单了。有两种方法：第一、继承java.lang.Thread第二、实现Runnable接口。实践：//继承Thread而重写了run()方法

```
public class Hello extends Thread{ int i. public void run(){ while(true){ System.out.println("Hello " i). if(i==10) break. }}}public class HelloThread { public static void main(String[] args){ Hello h1 = new Hello(). Hello h2 = new Hello(). h1.start(). //用两
```

个线程执行那10次循环 h2.start().}} 上面的例子是第一种方法，下面是第二种方法

```
public class TestThread { public static void main(String args[]) { Xyz r = new Xyz(). Xyz r1 = new Xyz(). Thread t1 = new Thread(r). Thread t2 = new Thread(r1). t1.start().//用两个线程执行那50次循环 t2.start(). }} //实现Runnable接口class Xyz implements Runnable { int i. public void run() { i = 0. while (true) { System.out.println("Hello " i). if ( i == 50 ) { break.}}}}
```

上面两种方法继承Thread类，是比较简单的，代码也比较少。但是我们不提倡使用这种方法。而第二种实现Runnable接口，更符合面向对象思想，Thread是把虚拟的CPU看成一个对象，封装了CPU的细节。但是Thread的构造线程的子类的方法中与CPU不相关，没有必要把CPU的细节都继承来。而实现Runnable则不影响java.lang.Thread的体系。而且便于其它类的继承。线程并发的代码和数据的执行顺序混乱，我们也需要自己调度和控制它们。请看附加教程，线程调度和并发。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