

Java向线程传递数据的三种方法Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Java_E5_90_91_E7_BA_BF_c104_644648.htm

在传统的同步开发模式下，当我们调用一个函数时，通过这个函数的参数将数据传入，并通过这个函数的返回值来返回最终的计算结果。但在多线程的异步开发模式下，数据的传递和返回和同步开发模式有很大的区别。由于线程的运行和结束是不可预料的，因此，在传递和返回数据时就无法象函数一样通过函数参数和return语句来返回数据。本文就以上原因介绍了几种用于向线程传递数据的方法，在下一篇文章中将介绍从线程中返回数据的方法。欲先取之，必先予之。一般在使用线程时都需要有一些初始化数据，然后线程利用这些数据进行处理，并返回结果。在这个过程中最先要做的就是向线程中传递数据。

一、通过构造方法传递数据 在创建线程时，必须要建立一个Thread类的或其子类的实例。因此，我们不难想到在调用start方法之前通过线程类的构造方法将数据传入线程。并将传入的数据使用类变量保存起来，以便线程使用（其实就是在run方法中使用）。下面的代码演示了如何通过构造方法来传递数据：

```
package mythread. public class MyThread1 extends Thread { private String name. public MyThread1(String name) { this.name = name. } public void run() { System.out.println("hello " name). } public static void main(String[] args) { Thread thread = new MyThread1("world"). thread.start(). } }
```

由于这种方法是在创建线程对象的同时传递数据的，因此，在线程运行之前这些数据就已经到位了，这样就不会造成数据在线程运行后才

传入的现象。如果要传递更复杂的数据，可以使用集合、类等数据结构。使用构造方法来传递数据虽然比较安全，但如果要传递的数据比较多时，就会造成很多不便。由于Java没有默认参数，要想实现类似默认参数的效果，就得使用重载，这样不但使构造方法本身过于复杂，又会使构造方法在数量上大增。因此，要想避免这种情况，就得通过类方法或类变量来传递数据。

二、通过变量和方法传递数据 向对象中传入数据一般会有两次机会，第一次机会是在建立对象时通过构造方法将数据传入，另外一次机会就是在类中定义一系列的public的方法或变量（也可称之为字段）。然后在建立完对象后，通过对象实例逐个赋值。下面的代码是对MyThread1类的改版，使用了一个setName方法来设置name变量：

```
package mythread. public class MyThread2 implements Runnable { private String name. public void setName(String name) { this.name = name. } }
```

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