

Java对象序列化小结计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_Java_E5_AF_B9_E8_B1_A1_c97_645383.htm

序列化的过程就是对象写入字节流和从字节流中读取对象。将对象状态转换成字节流之后，可以用java.io包中的各种字节流类将其保存到文件中，管道到另一线程中或通过网络连接将对象数据发送到另一主机。对象序列化功能非常简单、强大，在RMI、Socket、JMS、EJB都有应用。对象序列化问题在网络编程中并不是最激动人心的课题，但却相当重要，具有许多实用意义。对象序列化可以实现分布式对象。主要应用例如：RMI要利用对象序列化运行远程主机上的服务，就像在本地机上运行对象时一样。java对象序列化不仅保留一个对象的数据，而且递归保存对象引用的每个对象的数据。可以将整个对象层次写入字节流中，可以保存在文件中或在网络连接上传递。利用对象序列化可以进行对象的“深复制”，即复制对象本身及引用的对象本身。序列化一个对象可能得到整个对象序列。从上面的叙述中，我们知道了对象序列化是java编程中的必备武器，那么让我们从基础开始，好好学习一下它的机制和用法。java序列化比较简单，通常不需要编写保存和恢复对象状态的定制代码。实现java.io.Serializable接口的类对象可以转换成字节流或从字节流恢复，不需要在类中增加任何代码。只有极少数情况下才需要定制代码保存或恢复对象状态。这里要注意：不是每个类都可序列化，有些类是不能序列化的，例如涉及线程的类与特定JVM有非常复杂的关系。

```
import java.io.FileInputStream. import java.io.FileOutputStream. import
```

```
java.io.IOException. import java.io.ObjectInputStream. import
java.io.ObjectOutputStream. import java.io.Serializable. public class
SerialCtl implements Serializable { private String a. private transient
String b. public SerialCtl(String aa, String bb) { a = "Not Transient: "
aa. b = "Transient: " bb. } public String toString() { return a " 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