

JAVA编程技术基础：探究JAVA串行化的细节 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/180/2021_2022_JAVA_E7_BC_96_E7_A8_8B_c97_180992.htm 串行化是java.io包中的一部分，它被用来将对象转换成一串字节。在串行化的过程中，将一个实体转变成一系列表示对象的字节，这些字节可以写入文档以备后用，通过网络连接传输到其他程序，用来对初始对象进行拷贝等等。如果你需要串行化一个类，那么就必须对这个类执行java.io.Serializable，并且提供一个无争议的构造器。Serializable接口不需要任何方法，因此你的这个类不再需要别的了。如果你有一个简单的类似bean的类，而这个类仅仅是把原始的、可串行化对象作为属性，那么你就不需要再做别的了。当你需要将对象串行化为一系列字节或由一系列字节串行化而来时，Java可以负责所有的细节。你可能会遇到的两个问题串行化可能变得复杂得多。对于串行化，你通常会遇到两个常有的问题：类版本和复杂对象。版本可能是你将碰到的第一个问题。譬如说，如果你在将实体串行化到一个磁盘之后，给你的类添加一种方法或属性，然后想将串行化实体存储回内存中，你将得到一个java.io.InvalidClassException。之所以会出现这个错误，是因为你的类的版本已经改变。这个问题可以得到解决，但是你必须花一些功夫。说得更详细一点就是检查Java对象串行化细节。第二个可能的问题是，你的类是否有一些虚拟机不知道怎么串行化的状态信息。如果是那样，你可以通过以下两种方法来完成读写类的实例的任务。private void writeObject(java.io.ObjectOutputStream out) throws IOException

private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException. 注意，这两种方法是私有的，序列化是特殊的。如果你想了解情况，就请看java.io.ObjectOutputStream和java.io.ObjectInputStream的源代码，并且一定要阅读关于java.io.Serializable的javadoc的内容。以下是一个可以供你进行实验的完整序列化例子：

```
import java.io.FileInputStream; import java.io.FileOutputStream; import java.io.EOFException; import java.io.IOException; import java.io.ObjectInputStream; import java.io.ObjectOutputStream; import java.io.Serializable; import java.util.Random; public class SerializableTipHelper { private static final String STORE_FILE = "bytes.out"; private static final boolean VERBOSE = true; public static void main(String[] args) { if ( args.length != 1 & args.length != 2 ) { System.out.println("use: SerializableTipHelper (read|write) [numberToWrite]"). return; } if ( args[0].equals("write") ) { try { write(Integer.parseInt(args[1])); } catch (IOException e) { e.printStackTrace(); } } // defaults to read else { try { read(); } catch (IOException e) { e.printStackTrace(); } catch (ClassNotFoundException e) { e.printStackTrace(); } } } }
```

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