



## 4 特殊操作流

[4.1 标准输入输出流](#)

[4.2 打印流](#)

[4.3 对象序列化流](#)

[4.4 Properties](#)

### 4.1 标准输入输出流

System类中有两个静态的成员变量:

- `public static final InputStream in`: 标准输入流。通常该流对应于键盘输入或由主机环境或用户指定的另一个输入源
- `public static final PrintStream out`: 标准输出流。通常该流对应于显示输出或由主机环境或用户指定的另一个输出目标

自己实现键盘录入数据

- `BufferedReader br = new BufferedReader(new InputStreamReader( System.in ));`

写起来太麻烦，Java就提供了一个类实现键盘录入

- `Scanner sc = new Scanner( System.in );`

输出语句的本质：是一个标准的输出流

- `PrintStream ps = System.out;`
- PrintStream类有的方法，System.out都可以使用

### 4.2 打印流

打印流分类:

- 字节打印流: PrintStream
- 字符打印流: PrintWriter

打印流的特点:

- 只负责输出数据，不负责读取数据
- 有自己的特有方法

字节打印流

- `PrintStream(String fileName)`:使用指定的文件名创建新的打印流
- 使用继承父类的方法写数据，查看的时候会转码;使用自己的特有方法写数据，查看的数据原样输出

字符打印流

字符打印流PrintWriter的构造方法:

方法名	说明
PrintWriter(String fileName)	使用指定的文件名创建一个新的PrintWriter，而不需要自动执行刷新
PrintWriter(Writer out, boolean autoFlush)	创建一个新的PrintWriter <ul style="list-style-type: none"><li>out: 字符输出流</li><li>autoFlush: 一个布尔值，如果为真，则println， printf， 或 format方法将刷新输出缓冲区</li></ul>

## 4.3 对象序列化流

### 对象序列化

就是将对象保存到磁盘中，或者在网络中传输对象

这种机制就是使用一个字节序列表示一个对象，该字节序列包含:对象的类型、对象的数据和对对象中存储的属性等信息

字节序列写到文件之后，相当于文件中持久保存了一个对象的信息

反之，该字节序列还可以从文件中读取回来，重构对象,对它进行反序列化

要实现序列化和反序列化就要使用对象序列化流和对象反序列化流:

- 对象序列化流: ObjectOutputStream
- 对象反序列化流: ObjectInputStream

### 对象序列化流 ObjectOutputStream

- 将Java对象的原始数据类型和图形写入OutputStream。可以使用ObjectInputStream读取 (重构)对象。可以通过使用流的文件来实现对象的持久存储。如果流是网络套接字流，则可以在另一个主机上或另一个进程中重构对象

### 构造方法

- `ObjectOutputStream(OutputStream out)` :创建一个写入指定的OutputStream的ObjectOutputStream

### 序列化对象的方法

- `void writeObject(Object obj)`: 将指定的对象写入ObjectOutputStream

#### 注意

- 一个对象要想被序列化，该对象所属的类必须实现Serializable接口
- Serializable是一个标记接口， 实现该接口，不需要重写任何方法

### 对象反序列化流 ObjectInputStream

- ObjectInputStream反序列化先前使用ObjectOutputStream编写的原始数据和对象

### 构造方法

- `ObjectInputStream(InputStream in)`: 创建从指定的InputStream读取的ObjectInputStream

### 反序列化对象的方法:

- `Object readObject()`: 从ObjectInputStream读取一个对象

三个问题

用对象序列化流序列化了一个对象后，假如我们修改了对象所属的类文件,读取数据会不会出问题呢？

- 会出问题， 抛出InvalidClassException异常

如果出问题了，如何解决呢？

- 给对象所属的类加一个serialVersionUID  
private static final long serialVersionUID = 42L;

如果一个对象中的某个成员变量的值不想被序列化，又该如何实现呢？

- 给该成员变 量加transient关键字修饰，该关键字标记的成员变量不参与序列化过程

4.4 Properties

- 是一个Map体系的集合类
- Properties可以保存到流中，或从流中加载

作为集合的特有方法

方法名	说明
Object setProperty(String key, String value)	设置集合的键和值，都是String类型，底层调用Hashtable方法 put
String getProperty(String key)	使用此属性列表中指定的键搜索属性
Set<String> stringPropertyNames()	从该属性列表中返回一个不可修改的键集，其中键及其对应的值是字符串

Properties和IO流结合的方法:

方法名	说明
void load(InputStream inStream)	从输入字节流读取属性列表（键和元素对）
void load(Reader reader)	从输入字符流读取属性列表（键和元素对）
void store(OutputStream out, String comments)	将此属性列表（键和元素对）写入此 Properties表中，以适合于使用 load(InputStream)方法的格式写入输出字节流
void store(Writer writer, String comments)	将此属性列表（键和元素对）写入此 Properties表中，以适合使用 load(Reader)方法的格式写入输出字符流