

描述

java.io 包包括了三个主要的类和接口组：

- 建立数据流的类
- 用于序列化的类和接口
- 处理文件系统的类和接口

数据流

一个数据流可以是一个用于从一个数据源中读取数值的输入流，数据源如：HTTP服务器或者Java串；也可以是一个用于写入一个数据仓库的输出流，数据仓库如：一个文件或一个字节数组。数据可以是字节或者是字符。java.io 包包含了分别处理字节流和字符流的不同类。它

还包含了用于在字节流和字符流之间转换的类。

一个数据容器（例如一个文件），在典型的情况下，它会提供一个方法用于返回一个用于读取其内容的输入流，或一个向其内部存储数值的输出流。这些流可以根据在其中流过的内容组合成一个数据流链，并且它们可以被转换成任何一种流。例如，在获得一个输入流后，把输入的字节流转换成更高层次的 Java 类型，如字符串和整数（如图 1 所示），之后将数据输入流添加到文件中。

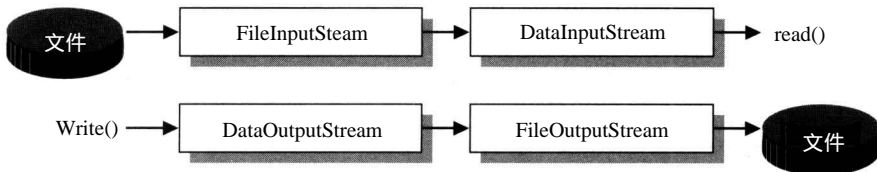


图1 复合IO流

序列化

对一个对象的序列化就意味着把其状态转化成一个字节流，这样该字节流可以被存储为这个对象的一个拷贝。并不是所有的对象都可以序列化的。一个对象如果能够序列化就将其称为可序列化的。如果一个 Java 对象所属的类或它的任意超类实现了 `Serializable` 接口或其子接口是 `Externalizable` 的，那么这个对象是可序列化的。

反序列化是将一个已序列化对象的结构转化为其原有形式的过程。

其数值为 Java 基本类型（如数组、字符串）的对象都是可序列化的。图 2 所示为对象的序列化及反序列化的过程。基本类型可以通过 `DataInputStream/DataOutputStream` 类中的方法来实现序列化及反序列化。

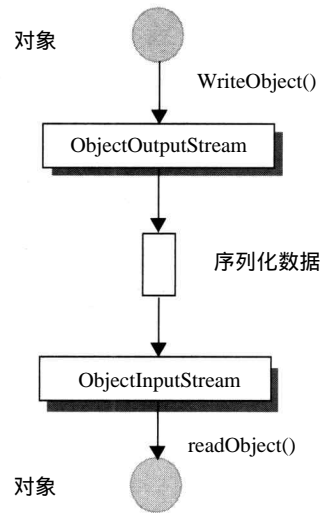


图2 序列化与反序列化

文件

`java.io` 包还包含了对文件进行操作的类，其操作如：以独立于平台的方式创建和删除文件。文件中的数据可以被看成字节或字符。

版本1.2中所作的修改

数据流

对数据流类的修改主要包括了对如 `close()`、`read()`、`ready()` 和 `write()` 方法的扩充，以便使这些方法更加符合它们在类中语义。

`PrintStream` 的构造函数已不再被禁止。

PrintWriter的内部实现已经被修改，以便使其更加容易子类化。

序列化

ObjectInputStream和ObjectOutputStream使子类化变得更加灵活。通过使用新的抽象类ObjectInputStream.GetField和ObjectOutputStream.PutField为取得和设置可序列化的域提供一个更加统一的方式。对象的序列化使用版本 1.2的安全模式，以及用新的 Serializable Permission类来控制定制的序列化过程。

文件类

File（文件）类被扩展用于支持对临时文件及锁式文件的创建。其中还增加了用于删除文件、改变文件修改时间和只读标记的新方法。

File、FileOutputStream和RandomAccessFile构造函数已经被修改，当不存在与所提供字符串标识名相同的文件时，它们会不断地抛出 FileNotFoundException而不是IOException。

RandomAccessFile添加了一个新的方法用于设置文件的长度，这样可以使文件易于删减和扩充。

所有对文件的访问都通过新的 FilePermission类来实现，在访问过程中使用版本 1.2的安全模式。

类与接口概述

超类

下面的抽象类定义了一个流所必须实现的最小操作集。 InputStream和OutputStream类用于字节流，而Reader和Writer用于字符流。

| | |
|--------------|----------------------------------|
| InputStream | 所有字节输入流的超类，提供从一个字节流中读取数据的基本输入方法。 |
| OutputStream | 所有字节输出流的超类，提供向一个字节流中写入数据的基本输出方法。 |
| Reader | 所有字符输入流的超类，提供从一个字符流中读取字符的基本输入方法。 |
| Writer | 所有字符输出流的超类，提供向一个字符流中写入字符的基本输出方法。 |

字节流字符流转换器

下面的类可以让程序把字节流转换成字符流。

| | |
|--------------------|----------------------------|
| InputStreamReader | 从一个字节流之中读取字节，然后把它们转换成字符。 |
| OutputStreamWriter | 把字符转换成一个字节编码，然后把它们写进一个字节流。 |

过滤流


当数据通过过滤流时，过滤流执行一些“处理”或“过滤”。一个过滤输出流在数据写到

最终目的地之前对其进行处理，而一个过滤输入流在数据从它的数据源读入后对其进行处理。FilterInputStream和FilterOutputStream类用于字节流，而FilterReader和FilterWriter类用于字符流。

| | |
|--------------------|---------------|
| FilterInputStream | 所有字节输入过滤流的超类。 |
| FilterOutputStream | 所有字节输出过滤流的超类。 |
| FilterReader | 所有字符输入过滤流的超类。 |
| FilterWriter | 所有字符输出过滤流的超类。 |




缓冲流

一个缓冲流是一个过滤流，它把流过它的数据缓存下来，以提高小的读写操作的执行效率。BufferedInputStream和BufferedOutputStream类用于字节流，BufferedReader和BufferedWriter类是对字符流进行操作的。

| | |
|--|---------------------------------|
|  BufferedInputStream | 一个过滤输入流，为从源输入流中读入的字节保留一个缓冲区。 |
| BufferedOutputStream | 一个过滤输出流，为向目的输出流中写入的字节保留一个缓冲区。 |
| BufferedReader | 一个过滤输入流，为从源字符输入流中读入的字符保留一个缓冲区。 |
| BufferedWriter | 一个过滤输出流，为向目的字符输出流中写入的字符保留一个缓冲区。 |

文件

下面的类提供了在Java中可用的与文件相关的操作。FileInputStream和FileOutputStream类是对文件中的字节进行读/写操作的，而FileReader和FileWriter类是对文件中的字符进行读/写操作的。RandomAccessFile类允许对文件的内容进行随机访问。

| | |
|---|------------------------|
|  File | 拥有能够对其进行操作的方法的文件。 |
| FileDescriptor | 一个已打开文件的句柄。 |
| 1.2 FileFilter | 使文件能够通过其File对象得到过滤的接口。 |
| FileInputStream | 一个用于从文件中读取字节的过滤输入流。 |
|  FileOutputStream | 一个用于向文件中写入字节的过滤输出流。 |
| FilenameFilter | 通过文件名对其进行过滤的接口。 |
| 1.2 FilePermission | 允许对文件和设备进行访问的属性。 |
| FileReader | 一个用于从文件中读取字符的过滤输入流。 |
| FileWriter | 一个用于向文件中写入字符的过滤输出流。 |
|  RandomAccessFile | 一个允许非顺序访问文件内容的类。 |

序列化

下面的类用于以一种紧凑的二进制方式对Java的基本数据进行序列化(和反序列化)，基本

数据包括：数组、字符串和类。它们通常用于在文件中存储或检索数据和对象，或通过网络连接交换序列化数据。与 `DataInput/DataOutput` 相关的方法用于序列化或反序列化 Java 的基本类型。与 `ObjectInput/ObjectOutput` 相关的方法用于序列化或反序列化 Java 的对象。

| | |
|---|---|
| <code>DataInput</code> | 用于反序列化Java基本数据的接口。 |
| <code>DataInputStream</code> | 用于反序列化Java基本数据的过滤输入流。 |
| <code>DataOutput</code> | 用于序列化Java基本数据的接口。 |
| <code>DataOutputStream</code> | 用于序列化Java基本数据的过滤输出流。 |
| <code>Externalizable</code> | 用于说明一个类的实例可以被序列化为这个类的特殊“外部”格式的接口。 |
| <code>ObjectInput</code> | 用于反序列化Java基本数据和对象的接口。 |
| △ <code>ObjectInputStream</code> | 用于反序列化Java基本数据和对象的输入流。 |
| 1.2 <code>ObjectInputStream.GetField</code> | 从 <code>ObjectInputStream</code> 中读出的可序列化域。 |
| <code>ObjectInputValidation</code> | 用于合法化反序列化对象的接口。 |
| <code>ObjectOutput</code> | 用于序列化Java基本数据和对象的接口。 |
| △ <code>ObjectOutputStream</code> | 用于序列化Java基本数据和对象的输出流。 |
| 1.2 <code>ObjectOutputStream.PutField</code> | 被写入一个 <code>ObjectOutputStream</code> 的可序列化域。 |
| △ <code>ObjectStreamClass</code> | 类描述器。 |
| 1.2 <code>ObjectStreamConstants</code> | 用于对象序列化的常量。 |
| 1.2 <code>ObjectStreamField</code> | 一个可序列化域。 |
| <code>Serializable</code> | 说明一个类的实例是否可序列化的接口。 |
| 1.2 <code>SerializablePermission</code> | 允许执行特定序列化操作。 |

内存流

下面的类用于像 IO 流这样的内存内的数据结构。

| | |
|---|-------------------|
| △ <code>ByteArrayInputStream</code> | 从字节数组中读取数据的输入流。 |
| △ <code>ByteArrayOutputStream</code> | 向字节数组中写入数据的输出流。 |
| △ <code>CharArrayReader</code> | 从字符数组中读取数据的输入流。 |
| <code>CharArrayWriter</code> | 向字符数组中写入数据的输出流。 |
| △ <code>StringReader</code> | 从字符串缓冲区中读取数据的输入流。 |
| △ <code>StringWriter</code> | 向字符串缓冲区中写入数据的输出流。 |

管道

下面的四个类用于创建两线程之间的数据流，这样的数据流被称为管道。一个线程可以在管道的一端写入，而另一个线程可以从管道的另一端读出。 `PipedInputStream` 和 `PipedOutputStream` 用于字节流，而 `Piped Reader` 和 `PipedWriter` 用于字符流。

| | |
|-----------------------------------|-----------------|
| <code>PipedInputStream</code> | 用于从管道里读取字节的输入流。 |
| <code>PipedOutputStream</code> | 用于向管道里写入字节的输出流。 |
| △ <code>PipedReader</code> | 用于从管道里读取输入的输入流。 |
| △ <code>PipedWriter</code> | 用于向管道里写入输出的输出流。 |

语法分析流

下面的类用于建立语法分析器。

| | |
|-----------------------|--------------------------|
| ▲ PushbackInputStream | 一个允许流中字节不被读取的过滤输入流。 |
| ▲ PushbackReader | 一个允许流中字符不被读取的过滤输入流。 |
| StreamTokenizer | 一个用于把流中数据解析成一系列令牌的过滤输入流。 |

输入流序列

下面的类用于把一系列输入流结合成一个单一的输入流。当一个输入流结束了，这个类会无缝地开始从下一个输入流中读取数据。

| | |
|---------------------|-------------------|
| SequenceInputStream | 把两个或多个输入流组成一个输入流。 |
|---------------------|-------------------|

其他类型的过滤流

下面的类用于实现各种有用的类。

| | |
|------------------|---------------------------------|
| LineNumberReader | 一个用于记录行数的缓冲字符输入流。 |
| ▲ PrintStream | 一个把Java基本数据类型和对象转化为可打印格式的过滤输出流。 |
| ▲ PrintWriter | 一个把Java基本数据类型和对象转化为可打印格式的字符输出流。 |

与序列化相关的异常

下面是在这个包中声明的与序列化相关的异常。它们不是 RuntimeException 的子类，所以必须在 throws 语句中说明或在其中被捕获。

| | |
|--------------------------|---|
| InvalidClassException | 如果发生与正在进行序列化/反序列化的对象的类相关的问题则此异常被抛出。 |
| InvalidObjectException | 如果非法反序列化一个对象则此异常被抛出。 |
| NotActiveException | 当一个对象未进行序列化/反序列化时 ObjectInputStream/ObjectOutputStream 中的一些特定的方法被唤醒，此异常被抛出。 |
| NotSerializableException | 如果对没有实现 Serializable/Externalizable 接口的对象进行序列化，此异常被抛出。 |
| ObjectStreamException | 与序列化相关的异常的超类。 |
| OptionalDataException | 如果在序列化流中遇到非法数据，此异常被抛出。 |
| StreamCorruptedException | 如果序列化流中的数据与其相联的控制数据不匹配，此异常被抛出。 |
| WriteAbortedException | 当流中包含了由于编写人中断了序列化过程所产生的不完整的序列化数据时，此异常被抛出。 |

其他的IO异常

下面是在这个包中声明的与序列化无关的异常。它们不是 RuntimeException 的子类，所以

必须在throws语句中说明或在其中被捕获。

CharConversionException

当有字节向字符转换或字符向字节转换失败时，此异常被抛出。

EOFException

在一数据输入流中读到文件尾时，此异常被抛出。

FileNotFoundException

当企图访问不存在的文件时，此异常被抛出。

IOException

与输入/输出相关的异常的超类。

InterruptedIOException

如果一个流操作被中断，此异常被抛出。

SyncFailedException

如果刷新与物理设备相对应的缓冲区失败，此异常被抛出。

UnsupportedEncodingException

如果字节到字符或字符到字节的特殊编码未获支持，此异常被抛出。

UTFDataFormatException

如果一个Unicode字符串在一多型UTF (unicode transfer format) 中被发现，此异常被抛出。

禁止使用

下面的类已经被禁止了。

LineNumberInputStream

被LineNumberReader取代。

StringBufferInputStream

被StringReader取代。