

Java语言深入JAVA之精髓IO流 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E8_AF_AD_E8_A8_80_c104_144866.htm —.Input和Output1.stream代表

的是任何有能力产出数据的数据源，或是任何有能力接收数据的接收源。在Java的IO中，所有的stream（包括Input和Out stream）都包括两种类型：1.1 以字节为导向的stream以字节为

导向的stream，表示以字节为单位从stream中读取或往stream中写入信息。以字节为导向的stream包括下面几种类型：1)

input stream：1) ByteArrayInputStream：把内存中的一个缓冲区作为InputStream使用2) StringBufferInputStream：把一

个String对象作为InputStream3) FileInputStream：把一个文件作为InputStream，实现对文件的读取操作4) PipedInputStream：实现了pipe的概念，主要在线程中使用5) SequenceInputStream

：把多个InputStream合并为一个InputStream2) Out stream1) ByteArrayOutputStream：把信息存入内存中的一个缓冲区中2) FileOutputStream：把信息存入文件中3) PipedOutputStream：实现了pipe的概念，主要在线程中使用4)

SequenceOutputStream：把多个OutStream合并为一个OutStream1.2 以Unicode字符为导向的stream以Unicode字符

为导向的stream，表示以Unicode字符为单位从stream中读取或往stream中写入信息。以Unicode字符为导向的stream包括下面

几种类型：1) Input Stream1) CharArrayReader：与ByteArrayInputStream对应2) StringReader：与StringBufferInputStream对应3) FileReader：与FileInputStream对应4) PipedReader：与PipedInputStream对应2) Out Stream1)

SequenceOutputStream：把多个OutStream合并为一个OutStream

1.2 以Unicode字符为导向的stream以Unicode字符为导向的stream，表示以Unicode字符为单位从stream中读取或往stream中写入信息。以Unicode字符为导向的stream包括下面

几种类型：1) Input Stream1) CharArrayReader：与ByteArrayInputStream对应2) StringReader：与StringBufferInputStream对应3) FileReader：与FileInputStream对应4) PipedReader：与PipedInputStream对应2) Out Stream1)

与ByteArrayInputStream对应2) StringReader：与StringBufferInputStream对应3) FileReader：与FileInputStream对应4) PipedReader：与PipedInputStream对应2) Out Stream1)

与StringBufferInputStream对应3) FileReader：与FileInputStream对应4) PipedReader：与PipedInputStream对应2) Out Stream1)

CharArrayWrite : 与ByteArrayOutputStream对应2) StringWrite : 无与之对应的以字节为导向的stream3) FileWrite : 与FileOutputStream对应4) PipedWrite : 与PipedOutputStream对应以字符为导向的stream基本上对有与之相对应的以字节为导向的stream。两个对应类实现的功能相同，只是在操作时的导向不同。如CharArrayReader : 和ByteArrayInputStream的作用都是把内存中的一个缓冲区作为InputStream使用，所不同的是前者每次从内存中读取一个字节的信息，而后者每次从内存中读取一个字符。

1.3 两种不同导向的stream之间的转换

InputStreamReader和OutputStreamReader : 把一个以字节为导向的stream转换成一个以字符为导向的stream。

2. stream添加属性

2.1 “为stream添加属性”的作用

运用上面介绍的Java中操作IO的API，我们就可完成我们想完成的任何操作了。但通过FilterInputStream和FilterOutputStream的子类，我们可以为stream添加属性。下面以一个例子来说明这种功能的作用。如果我们要往一个文件中写入数据，我们可以这样操作

```
FileOutputStream fs = new FileOutputStream( " test.txt " );
```

然后就可以通过产生的fs对象调用write()函数来往test.txt文件中写入数据了。但是，如果我们想实现“先把要写入文件的数据先缓存到内存中，再把缓存中的数据写入文件中”的功能时，上面的API就没有一个能满足我们的需求了。但是通过FilterInputStream和FilterOutputStream的子类，为FileOutputStream添加我们所需要的功能。

2.2 FilterInputStream的各种类型

2.2.1 用于封装以字节为导向的InputStream

- 1) DataInputStream : 从stream中读取基本类型 (int、char等) 数据。
- 2) BufferedInputStream : 使用缓冲区
- 3) LineNumberInputStream :

会记录input stream内的行数，然后可以调用getLineNumber()和setLineNumber(int)4) PushbackInputStream：很少用到，一般用于编译器开发 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