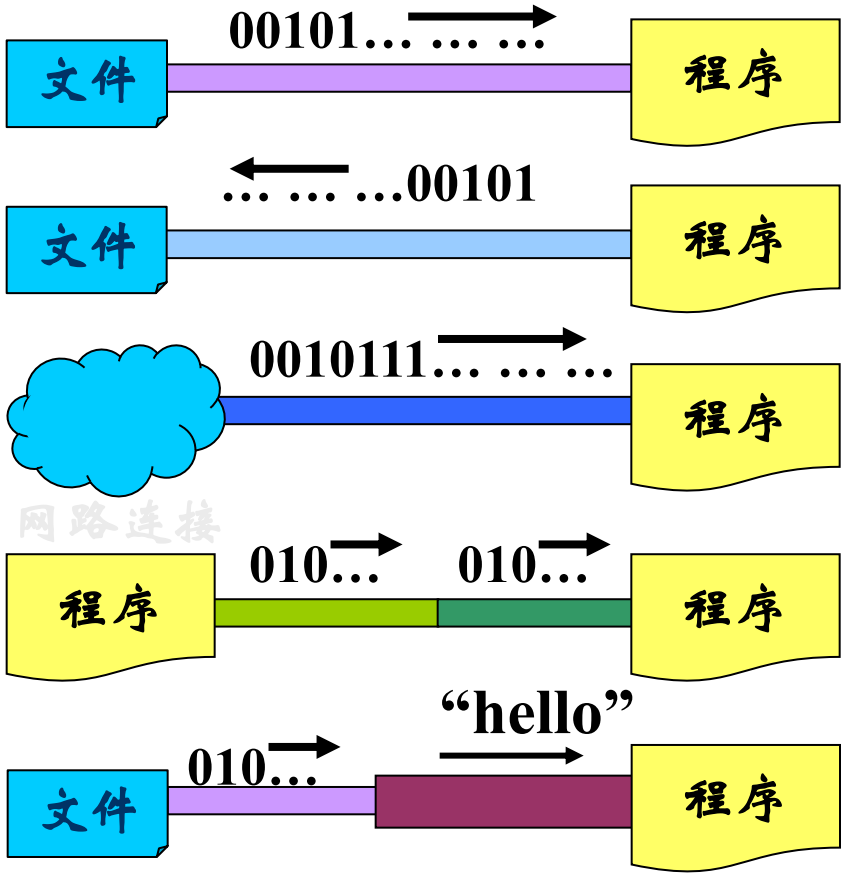
第七章 输入输出流

本章内容

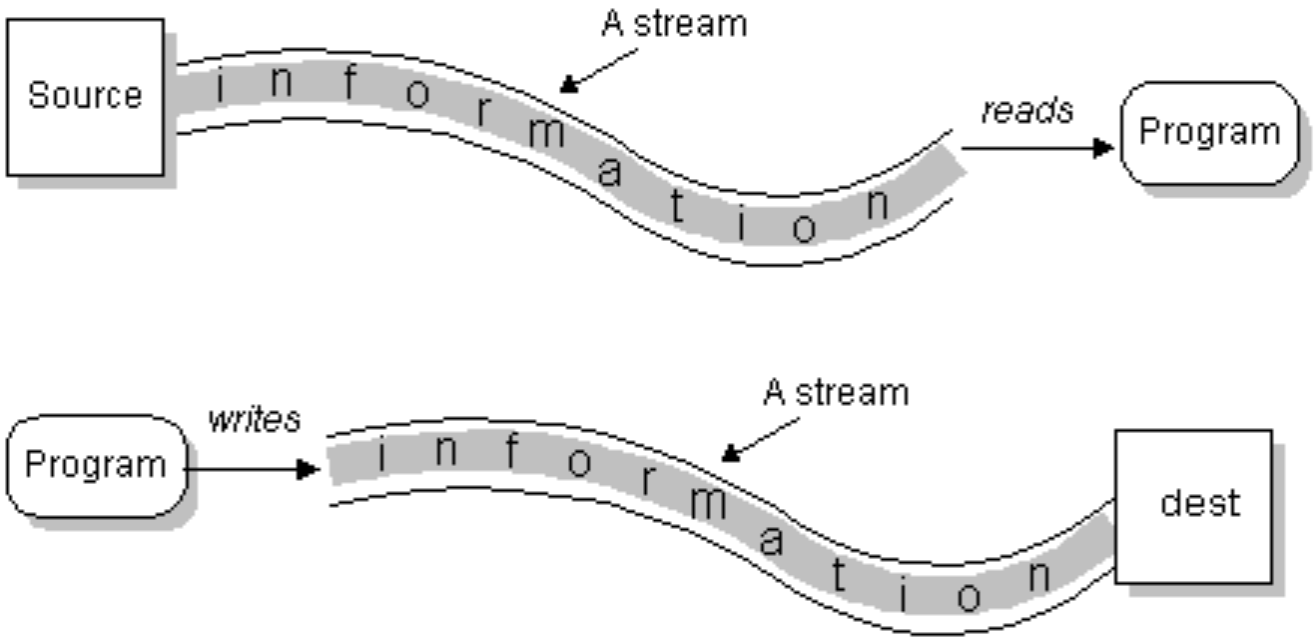
- Java 流式输入/输出原理
- Java流类的分类
- 输入/输出流类
- 常见的节点流和处理流

Java流式输入/输出原理



- 在Java程序中，对于数据的输入/输出操作以“流”(stream)方式进行；J2SDK提供了各种各样的“流”类，用以获取不同种类的数据；程序中通过标准的方法输入或输出数据。

流是一个很形象的概念


Source
A stream
i n f o r m a t i o n
reads
Program
Program
writes
A stream
i n f o r m a t i o n
dest

输入/输出流的分类

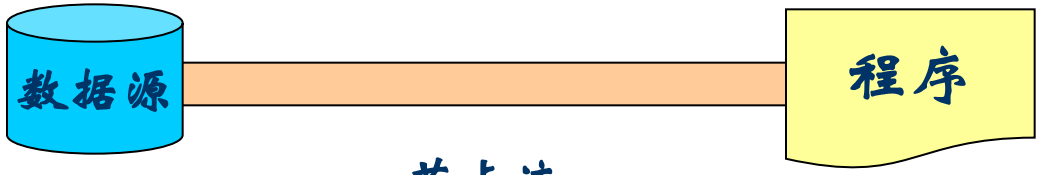
- java.io 包中定义了多个流类型(类或抽象类)来实
- 现输入/输出功能；可以从不同的角度对其进行分类：
  - 按数据流的方向不同可以分为输入流和输出流。(以程序的角度来考虑)
  - 按处理数据单位不同可以分为字节流和字符流。
  - 按照功能不同可以分为节点流和处理流。

- J2SDK 所提供的所有流类型位于包java.io内都分别继承自以下四种抽象流类型。

	字节流	字符流
输入流	InputStream	Reader
输出流	OutputStream	Writer

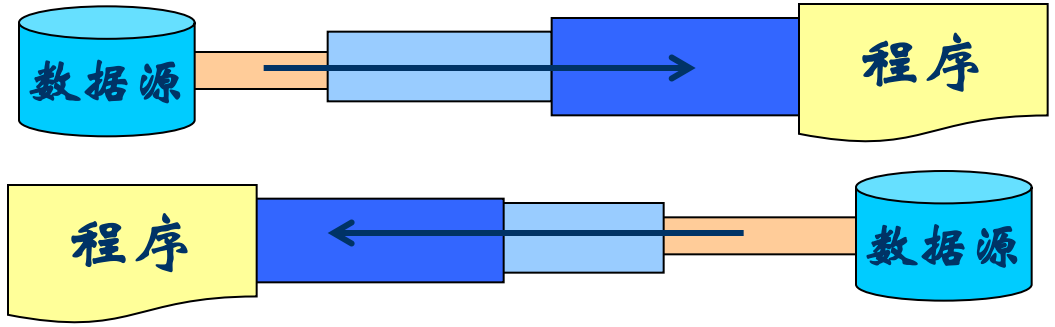
节点流和处理流

- 节点流为可以从一个特定的数据源（节点）读写数据（如：文件，内存）



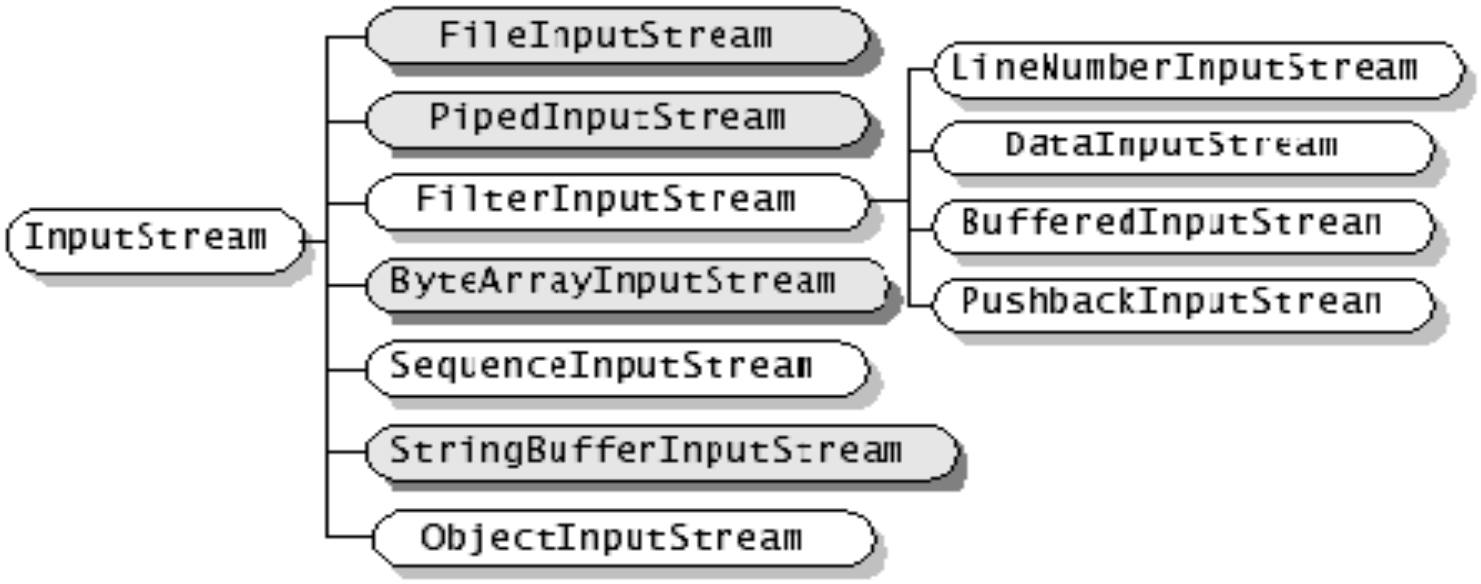
节点流

- 处理流是“连接”在已存在的流（节点流或处理流）之上，通过对数据的处理为程序提供更为强大的读写功能。



InputStream

- 继承自InputSteam的流都是用于向程序中输入数据，且数据的单位为字节（8 bit）；下图中深色为节点流，浅色为处理流。

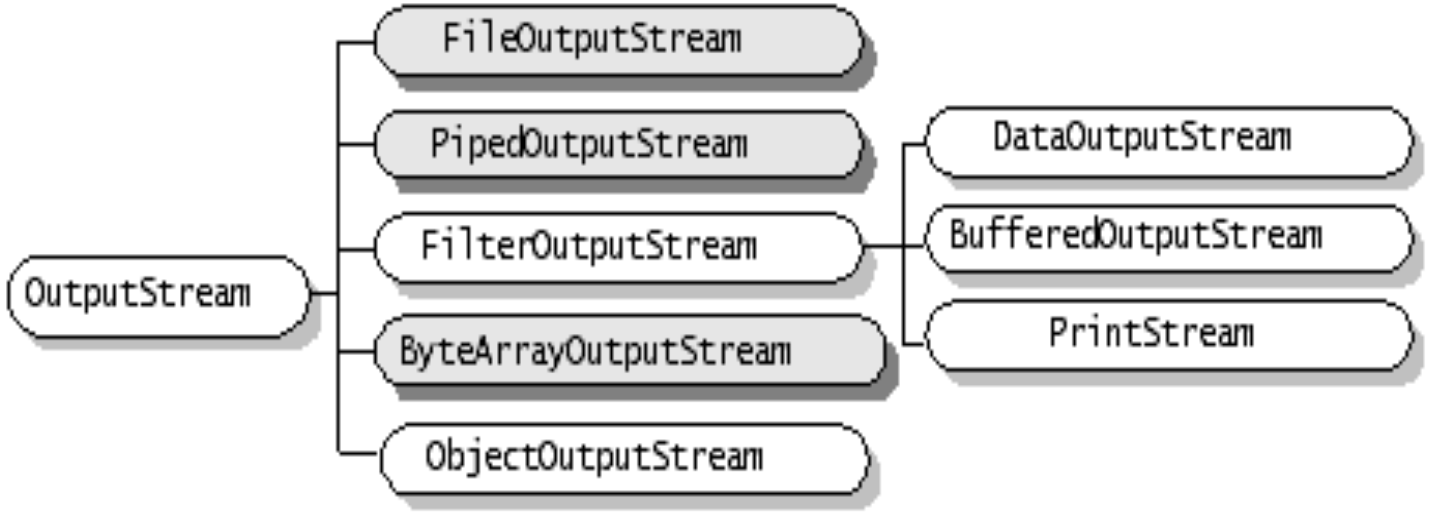


InputStream的基本方法

```
//读取一个字节并以整数的形式返回(0~255),
//如果返回-1已到输入流的末尾。
int read() throws IOException

//读取一系列字节并存储到一个数组buffer,
//返回实际读取的字节数，如果读取前已到输入流的末尾返回-1
int read(byte[] buffer) throws IOException

//读取length个字节
//并存储到一个字节数组buffer，从off位置开始存,最多len
//返回实际读取的字节数，如果读取前以到输入流的末尾返回-1
int read(byte[] buffer, int off, int len)
                        throws IOException
//关闭流释放内存资源
void close() throws IOException
```

OutputStream

- 继承自OutputSteam的流是用于程序中输入数据，且数据的单位为字节（8 bit）；下图中深色为节点流，浅色为处理流。



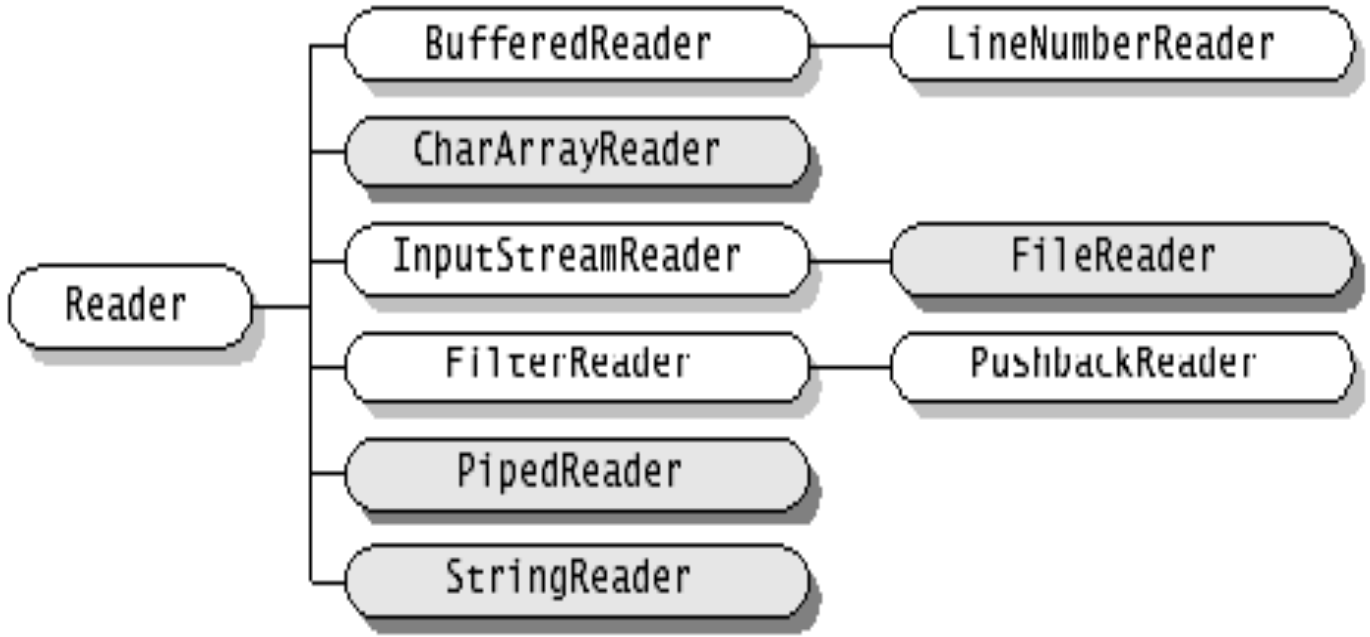
OutputStream的基本方法

```
//向输出流中写入一个字节数据,该字节数据为参数b的低8位
void write(int b) throws IOException

//将一个字节类型的数组中的数据写入输出流
void write(byte[] b) throws IOException

//将一个字节类型的数组中的从指定位置（off）开始的
//len个字节写入到输出流
void write(byte[] b, int off, int len)
                throws IOException
//关闭流释放内存资源
void close() throws IOException

//将输出流中缓冲的数据全部写出到目的地
void flush() throws IOException
```

良好的编程习惯→,先flush(),再close()

Reader

- 继承自Reader的流都是用于向程序中输入数据，且数据的单位为字符（16 bit）；下图中深色为节点流，浅色的为处理流。

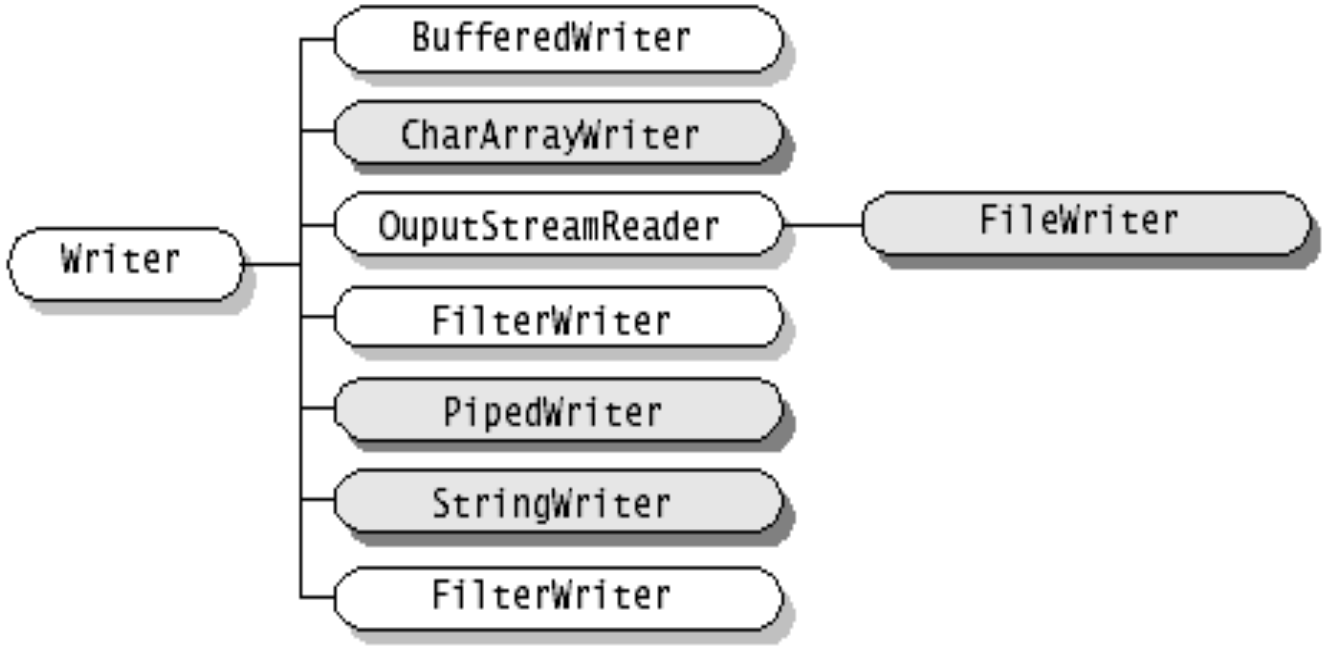


Reader 的基本方法

```
//读取一个字符并以整数的形式返回(0~255),
//如果返回-1已到输入流的末尾。
int read() throws IOException

//读取一系列字符并存储到一个数组buffer,
//返回实际读取的字符数，如果读取前已到输入流的末尾返回-1
int read(char[] cbuf) throws IOException

//读取length个字符
//并存储到一个数组buffer，从off位置开始存,最多读取len
//返回实际读取的字符数，如果读取前以到输入流的末尾返回-1
int read(char[] cbuf, int off, int len)
                    throws IOException
//关闭流释放内存资源
void close() throws IOException
```

Writer

- 继承自Writer的流都是用于程序中输入数据，且数据的单位为字符（16 bit）；下图中深色为节点流，浅色为处理流。



Writer 的基本方法

```
//向输出流中写入一个字符数据,该字节数据为参数b的低16位
void write(int c) throws IOException
//将一个字符类型的数组中的数据写入输出流,
void write(char[] cbuf) throws IOException
//将一个字符类型的数组中的从指定位置（offset）开始的
//length个字符写入到输出流
void write(char[] cbuf, int offset, int length)
                throws IOException
//将一个字符串中的字符写入到输出流
void write(String string) throws IOException
//将一个字符串从offset开始的length个字符写入到输出流
void write(String string, int offset, int length)
                throws IOException
//关闭流释放内存资源
void close() throws IOException
//将输出流中缓冲的数据全部写出到目的地
void flush() throws IOException
```

节点流类型

类　型	字 符 流	字 节 流
File（文件）	FileReader FileWriter	FileInputStream FileOutputStream
Memory Array	CharArrayReader CharArrayWriter	ByteArrayInputStream ByteArrayOutputStream
Memory String	StringReader StringWriter	—
Pipe（管道）	PipedReader PipedWriter	PipedInputStream PipedOutputStream

访问文件

- FileInputStream和FileOutputStream分别继承自InputStream和OutputStream用于向文件中输入和输出字节。
- FileInputStream和FileOutputStream的常用构造方法：
- FileInputStream(String name) throws FileNotFoundException
- FileInputStream(File file) throws FileNotFoundException
- FileOutputStream(String name)throws FileNotFoundException
- FileOutputStream(File file) throws FileNotFoundException
- FileOutputStream(File file, boolean append)
- throws FileNotFoundException

- FileInputSteam 和 FileOutputStream 类支持其父类InputStream 和 OutputStream 所提供的数据读写方法。
- 注意：
- 在实例化FileInputStream和FileOutputSteam流时要用try – catch语句以处理其可能抛出的FileNotFoundException。
- 在读写数据时也要用try – catch语句以处理可能抛出的 IOException。
- FileNotFoundException是IOException的子类

例：TestFileInputStream.java / TestFileOutputStream.java

访问文件

- FileReader 和 FileWriter 分别继承自Reader和Writer，FileInputSteam与FileOutputStream类似，所不同的时FileReader和FileWriter向文件输入和输出的数据单位为字符。
- FileReader和FileWriter的常用构造方法：

```
public FileWriter(File file) throws IOException
public FileWriter(File file, boolean append)
                          throws IOException
public FileWriter(String fileName)throws IOException
public FileWriter(String fileName,boolean append)
                          throws IOException
public FileReader(String fileName)
                          throws FileNotFoundException
public FileReader(File file)
                          throws FileNotFoundException
```

例：TestFileWriter.java / TestFileReader.java

处理流类型

处理类型	字 符 流	字 节 流
Buffering	BufferedReader BufferedWriter	BufferedInputStream BufferedOutputStream
Filtering	FilterReader FilterWriter	FilterInputStream FilterOutputStream
Converting between bytes and character	InputStreamReader OutputStreamWriter	
Object Serialization	—	ObjectInputStream ObjectOutputStream
Data conversion	—	DataInputStream DataOutputStream
Counting	LineNumberReader	LineNumberInputStream
Peeking ahead	PusbackReader	PushbackInputStream
Printing	PrintWriter	PrintStream

缓冲流

- 缓冲流要“套接”在相应的节点流之上，对读写的数据提供了缓冲的功能，提高了读写的效率，同时增加了一些新的方法。
- J2SDK提供了四种缓存流，其常用的构造方法为：

```
BufferedReader(Reader in)
BufferedReader(Reader in,int sz) //sz 为自定义缓存区的大小
BufferedWriter(Writer out)
BufferedWriter(Writer out,int sz)
BufferedInputStream(InputStream in)
BufferedInputStream(InputStream in,int size)
BufferedOutputStream(OutputStream out)
BufferedOutputStream(OutputStream out,int size)
```

- BufferedReader提供了readLine方法用于读取一行字符串（以\r或\n分隔）。
- BufferedWriter提供了newLine用于写入一个行分隔符。
- 对于输出的缓冲流，写出的数据会先在内存中缓存，使用flush方法将会使内存中的数据立刻写出。

例：TestBufferStream1 / 2.java

转换流

- InputStreamReader和OutputStreamWriter用与字节数据到字符数据之间的转换。
- InputStreamReader 需要和 InputStream “套接” 。
- OutpStreamWriter 需要和 OutputStream “套接” 。
- 转换流在构造时可以指定其编码集合，例如：

```
InputStream isr = new InputStreamReader
                          (System.in, “ISO8859_1”)
```

nio：异步IO,非阻塞

例：TestTransform1 / 2.java

Print 流

- PrintWriter和PrintStream 都属于输出流，分别针对与字符和字节。
- PrintWriter和PrintStream提供了重载的print
- Println方法用于多种数据类型的输出。
- PrintWriter和PrintStream的输出操作不会抛出异常，用户通过检测错误状态获取错误信息。
- PrintWriter和PrintStream有自动flush功能。

```
PrintWriter(Writer out)
PrintWriter(Writer out,boolean autoFlush)
PrintWriter(OutputStream out)
PrintWriter(OutputStream out,boolean autoFlush)
PrintStream(OutputStream out)
PrintStream(OutputStream out,booleanautoFlush)
```

例：TestPrintStream1 / 2 / 3.java

Object流

- 直接将Object写入或读出
  - transient关键字
  - serializable接口
  - Externalizable 接口
    - void **writeExternal**(ObjectOutput out) throws IOException
    - void **readExternal**(ObjectInput in) throws IOException, ClassNotFoundException

TestObjectIO.java

总结

- InputStream/OutputStream //最基础的抽象类，字节流
- Reader/Writer //最基础的抽象类，字符流
- FileInputStream / FileOutputStream //File打头的用来操作文件，字节流
- FileReader / FileWriter //File打头的用来操作文件，字符流
- BufferedInputStream / BufferedOutputStream //带缓冲的，字节流
- BufferedReader / BufferedWriter //带缓冲的，字符流
- ByteArrayInputStream / ByteArrayOutputStream //读写内存中的字符数组
- InputStreamReader / OutputStreamWriter //字节转字符，转化流
- DataInputStream / DataOutputStream //读写基本数据类型
- PrintStream / PrintWriter //都是输出流，不抛出异常，自动flush
- ObjectInputStream / ObjectOutputStream //读写Object
  - Serializable接口→标记性接口
  - Externalizable →自己控制序列化
  - Transient→