

## 四大IO抽象类

### • InputStream

此抽象类是表示字节输入流的所有类的父类。不可以实例化。数据的读取需要由它的子类来实现。继承自InputStream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)。

- `ByteArrayInputStream`、`StringBufferInputStream`、`FileInputStream` 是三种基本的介质流，它们分别从Byte 数组、`StringBuffer`、和本地文件中读取数据。
- `PipedInputStream` 是从与其它线程共用的管道中读取数据
- `ObjectInputStream` 和所有`FilterInputStream` 的子类都是装饰流（装饰器模式的主角）

常用方法：

`int read()`：读取一个字节的的数据，并将字节的值作为int类型返回(0-255之间的一个值)。如果未读出字节则返回-1(返回值为-1表示读取结束)。

`void close()`：关闭输入流对象，释放相关系统资源。

### • OutputStream

此抽象类是表示字节输出流的所有类的父类。输出流接收输出字节并将这些字节发送到某个目的地。

- `ByteArrayOutputStream`、`FileOutputStream` 是两种基本的介质流，它们分别向Byte 数组、和本地文件中写入数据。
- `PipedOutputStream` 是向与其它线程共用的管道中写入数据。
- `ObjectOutputStream` 和所有`FilterOutputStream` 的子类都是装饰流。

常用方法：

`void write(int n)`：向目的地中写入一个字节。

`void close()`：关闭输出流对象，释放相关系统资源。

- **Reader**

Reader用于读取的字符流抽象类，数据单位为字符。

**常用方法：**

`int read()`：读取一个字符的数据，并将字符的值作为int类型返回(0-65535之间的一个值，即Unicode值)。如果未读出字符则返回-1(返回值为-1表示读取结束)。

`void close()`：关闭流对象，释放相关系统资源。

- **Writer**

Writer用于写入的字符流抽象类，数据单位为字符。

**常用方法：**

`void write(int n)`：向输出流中写入一个字符。

`void close()`：关闭输出流对象，释放相关系统资源。

## 常用的节点流

- 父类：`InputStream`、`OutputStream`、`Reader`、`Writer`
- 文件：`FileInputStream`、`FileOutputStream`、`FileReader`、`FileWriter` 文件进行处理的节点流
- 数组：  
`ByteArrayInputStream`、`ByteArrayOutputStream`、`CharArrayReader`、`CharArrayWriter` 对数组进行处理的节点流（对应的不再是文件，而是内存中的一个数组）
- 字符串：`StringReader`、`StringWriter` 对字符串进行处理的节点流
- 管道：`PipedInputStream`、`PipedOutputStream`、`PipedReader`、`PipedWriter` 对管道进行处理的节点流

## 常用的处理流

- 缓冲流：`BufferedInputStream`、`BufferedOutputStream`、`BufferedReader`、`BufferedWriter` 增加缓冲功能，避免频繁读写硬盘。
- 转换流：`InputStreamReader`、`OutputStreamReader` 实现字节流和字符流之间的转换。
- 数据流：`DataInputStream`、`DataOutputStream` 等-提供将基础数据类型写入到文件中，或者读取出来。

## 转换流

`InputStreamReader`、`OutputStreamWriter` 要 `InputStream` 或 `OutputStream` 作为参数，实现从字节流到字符流的转换。

构造函数：

`InputStreamReader(InputStream);` //通过构造函数初始化，使用的是本系统默认的编码表GBK。

`InputStreamWriter(InputStream, String charSet);` //通过该构造函数初始化，可以指定编码表。

`OutputStreamWriter(OutputStream);` //通过该构造函数初始化，使用的是本系统默认的编码表GBK。

`OutputStreamWriter(OutputStream, String charSet);` //通过该构造函数初始化，可以指定编码表。

### 1. `InputStream/OutputStream`

字节流的抽象类。

### 2. `Reader/Writer`

字符流的抽象类。

### 3. `FileInputStream/FileOutputStream`

节点流：以字节为单位直接操作“文件”。

### 4. `ByteArrayInputStream/ByteArrayOutputStream`

节点流：以字节为单位直接操作“字节数组对象”。

5. ObjectInputStream/ObjectOutputStream

处理流：以字节为单位直接操作“对象”。

6. DataInputStream/DataOutputStream

处理流：以字节为单位直接操作“基本数据类型与字符串类型”。

7. FileReader/FileWriter

节点流：以字符为单位直接操作“文本文件”（注意：只能读写文本文件）。

8. BufferedReader/BufferedWriter

处理流：将Reader/Writer对象进行包装，增加缓存功能，提高读写效率。

9. BufferedInputStream/BufferedOutputStream

处理流：将InputStream/OutputStream对象进行包装，增加缓存功能，提高读写效率。

10. InputStreamReader/OutputStreamWriter

处理流：将字节流对象转化成字符流对象。

11. PrintStream

处理流：将OutputStream进行包装，可以方便地输出字符，更加灵活。