

Java 流(I/O Stream)的介绍

流的概念

“流”是 Java 中处理数据的一个抽象的概念。当程序运行时，需要输入输出大量不同的数据，比如网络，文件，各种 IO 设备等等。然而，不同情况下数据获取及输出的方式、数据传递的情形也会不同。如果分门别类地进行编写，会让代码显得复杂冗余且难以理解。但当我们把所有这些数据的流入和流出抽象成水流的概念，就会大大简化程序的编写。

可以想象我们的程序运行就像在一个厨房里干活，数据就好比需要用到的水。没有处理过的水从水龙头流入，而处理过的水则由下水道排出。我们只需在需要取水时打开水龙头(read)，需要排水时打开下水道的活塞(write)就足够了，而不必因为水的具体来源和去向而进行不同的操作。当然，水龙头流出的水可能速度缓慢，甚至偶尔停水；下水道也有可能临时堵塞，排水需要等一段时间。但当我们把这些情况都考虑到时，其它因素则可以通通被忽略了。

当然，具体的“水如何来，又如何走”的问题还是需要考虑的。因此在“流”的抽象概念基础上，会实现各种带有具体功能的不同的类，在实际需要中，则由具体的类实现特定情况下的流的功能。

流的种类

根据数据的基本单位，流可以分为字节流(Byte Stream)和字符流(Character Stream)。顾名思义，字节流以字节为单位，可以传递二进制数据；而字符流则会传递 Unicode 字符。

根据写入和写出的不同，字节流分为两个抽象类:InputStream 和 OutputStream，而字符流分为 Reader 和 Writer。这四个就是流的基本抽象类，其它类都将继承其中之一。

字节流的一些比较重要的实现类有：BufferedInput(OutputStream), DataInput(OutputStream), FileInputStream;

字符流的一些比较重要的实现类有：BufferedReader(Writer), FileReader(Writer), InputStreamReader, OutputStreamWriter.

实例

3-12 的大作业实现了从网上下载图片的功能。在给定图片 url 后，我们会进行如下的代码：

```
URLConnection mURLConnection = url.openConnection();
mURLConnection.connect();
InputStream iStream = mURLConnection.getInputStream();
```

当需要读取数据时，会进行如下操作：

```
iStream.read(buffer, offset, bufferLen);
```

可知我们获取了基本输入字节流 `InputStream iStream`，并且调用了其 `read` 函数读取数据。然而现实中，由于 url 的协议是 HTTP 或者 HTTPS，实际获取的 `URLConnection` 其实真实身份是 `HttpURLConnection`，因而我们拿到的实际是 `BufferedInputStream`。其实现方式是添加了一个 `buffer` 并可以从中读取数据，而 `buffer` 中的数据则由更深一层的 `Socket` 的 `InputStream` 填充。