I/O流

# 第一章 IO流概述

将数据持久化：文件、数据库、网络；

IO ： input、output : 输入输出；

IO流：数据就像水流一样，可以实现从一个节点流向另外一个节点；其中，一个节点是程序，另外一个节点可以是文件或网络或内存等；

也就是说，IO流就是指一种用于传输数据的技术；

## 1.1 IO流的分类

1. 按照流向不同分类：

输入流：从其他节点流向程序（节点可以是：文件或网络或内存或键盘）；

输出流：从程序流向其他节点（节点可以是：文件或网络或内存或显示器）；

1. 按照传输单位不同分类：

字节流：按一个字节一个字节的传输，效率较低而且容易出现乱码，但可以读写任意类型的数据；

字符类：按一个字符一个字符的传输，效率较高而且底层使用默认的字符编码格式进行了包装，所以不太容易出现中文乱码问题，但只能读写纯文本的数据；

1. 按照功能不同分类：

节点流(基础流)：用于读写数据；

处理流(包装流)：可以在节点流的基础上增强一些其他功能，比如提高效率等；

## 1.2 文件流体系图

1）字节输入流InputStream:

2）字节输出流OutputStream :

3）字符输入流Reader :

4）字符输出流Writer:

# 第二章 IO流API

## 2.1 FileInputStream :

使用步骤：

1. 创建流对象打开一个连接指向一个具体的文件：

注意：该文件必须存在，如果不存在，则报“找不到文件异常”；

1. 调用读取方法：
2. 关闭资源（因为连接着文件，所以垃圾器不会回收）：

## 2.2 FileOutputStream :

使用步骤：

1. 创建流对象打开一个连接指向一个具体的文件：

注意：该文件不存在，则创建并指向；如果文件存在，则直接指向；

1. 调用读取方法：
2. 关闭资源（因为连接着文件，所以垃圾器不会回收）：

## 2.3 练习1：

## 2.4 FileReader：

## 2.5 FileWriter：

## 2.6 练习2：

## 2.7 BufferedInputStream、BufferedOutputStream：

## 2.8 BufferedReader:

## 2.9 BufferedWriter:

## 2.10 练习3：

## 2.11 ObjectInputStream、ObjectOutputStream：

序列化：将二进制文件持久化到本地或网络

反序列化：将存到本地或网络的数据还原到程序中

作用：当把一个对象保存到文件或者数据库时使用；

## 2.12 PrintStream、PrintWriter：

## 2.13 InputStreamReader、OutputStreamWriter：

## 2.14 练习4：

## 2.15 Properties:

Properties属于集合中Hashtable的子类，遵循hashtable的特点；

特点：1、保存键值对，键不能重复，值可以重复；

1. 不允许null键null值；

功能：用于读写配置文件；

配置文件的特点：键=值 ， 默认类型都是String;

## 2.16 FileUtils：



## 2.17 File类:

File类代表了一组对象，用于描述磁盘中的文件或者目录的对象；

功能：用于操作文件或目录的，比如创建、删除、重命名、获取信息等等，但不能读写文件的内容（文件流）；

### 练习：