#### IO流

**概述**：

Java中I/O操作主要是指使用java.io包下的内容，进行输入（读取数据），输出操作（写出数据）。

分类：

流向：输入流（把数据从其他设备上读取到内存上）和输出流（把数据从内存上写出到其他设备上）

格局数据：字节流和字符流

##### 字节流：

一切文本数据（图片，文本，视频）在存储时，都是以二进制数字的形式进行保存；不建议使用中文否则会出现乱码

###### 字节输出流【OutputStream】

Java.io.OutputStream抽象类是表示字节输出流的所有类的父类；将指定的字节信息写出到目的地；

public void close()：关闭此输出流并释放与此流相关联的任何系统资源；

public void flush()：刷新此输出流并强制任何缓存的输出字节被写出；

public void write(byte[] b)：将b.length字节从指定的字节数组写入此输出流。

如果写入的第一个字节是正数，那么显示的时候会查询ASCII码表

如果写入的第一个字节是负数，那么第一个字节和第二个字节会组成一个中文显示查询系统默认码表（GBK）；

public void write(byte[] b，int off，int len)：从指定的字节数组写入len字节，从偏移量off开始输出到此输出流；

public abstract void write(int b)：将指定的字节输出流

**扩展**：当完成流操作时，必须调close方法，释放系统资源；write将字节流从内存写入到硬盘中

###### 字节输出流续写

public FileOutputStream(File file，boolean append)：创建文件输入流以写入由指定的File对象表示的文件

public FileOutputStream(String name，boolean append)：创建文件输出流已指定的名称写入数据

其中boolean类型中true表示创建对象，不会覆盖原文件继续写入，false表示创建新文件，覆盖原有文件；换行打印加 “\r\n“

###### 字节输入流【InputStream】

Java.io.InputStream抽象类是表示字节输入所有类的父类，可以读取到自己信息到内存中。它定义了字节输入流的基本共性功能方法。

Public void close()：关闭输入流并释放与此流相关的任何系统资源；

Public abstract int read()：从输入流读取数据的下一个字节

Public int read(byte [] b)：从输入流中读取一些字节数，并将它们存储到字节数组b中；

FileInputStream文件字节输入流，作用：把硬盘文件中的数据，读取到内存中使用；

读取时使用构造方法；

**文件复制的原理**：

文件的读写；

#### 字符流：

当使用字节流读取文件，如果遇到中文字符，可能不会显示完整的字符，因为一个中文字符会可能占用多个字节存储；因此java提供了一些字符流类，以字符为单位读写数据，专门用于处理文本文件；

##### 字符输入流【Reader】

**子类fileReader**

Java.io.Reader抽象类是用于读取字符流的所有类的父类，可以读取字符信息到内存中。定义了字符数如流的基本共性功能方法；

public void close()：关闭此流并释放此流相关联的任何系统资源

public void read()：从输入流读取下一个字符。

public void read(char[] cbuf)：从输入流中读取一些字符，并将它们存储到字符cbuf中

##### 字符输出流【Writer】

Java.io.Writer抽象类是表示用于写出字符流的所有类的父类，将指定的字符信息写出到目的地。它定义了字符输出流的基本共性功能方法

void write(int c)：写入单个字符

void write(char[] cbuf)：写入字符数组

abstract void write(char[] cbuf，offset，len)：写入字符数组的某一部分，off数组的开始索引字符的个数；

void write(String str)：写入字符串

void write(String str，int off，int len)：写入字符串的某一部分，off字符串的开始索引，len写的字符的个数

void close()：关闭此流

void flush():关闭此流，但要先刷新它；

**子类fileWriter**

flush方法和close方法的区别：

在字符输出流中，flush刷新缓冲区，流对象可以继续使用，而close方法先刷新缓冲区，然后通知系统释放资源，流对象不可以在使用了；

##### 字符输出流的续写

与字节输出流的追加续写相似；