2 字节流

2.1 IO流概述和分类

概述

- IO：输入 | 输出 input | output
- 流：一种抽象概念，是对数据传输的总称。就是说数据在设备间的传输成为留，其本质是数据传输
- IO流就是用来处理设备间数据传输问题的
  - 文件复制，文件上传，文件下载

分类

- 按照数据的流向：输入流 | 输出流
- 按照数据类型： 字节流：字节输入流；字节输出流 | 字符流：字符输入流；字符输出流

一般来说，我们说IO流的分类时按照数据类型来分类的

什么情况下使用呢?

- 如果数据通过Windows自带的记事本打开，我们还可以读懂里面的内容，就是用字符流，否则使用字节流，如果你不知道用哪种，就用字节流

2.2 字节流写数据

字节流抽象基类

- `InputStream` 这个抽象类时所有字节输入流类的超类
- `OutputStream` 这个抽象类时表示字节输出流的所有类的超类
- 子类名特点：都是以其父类名作为子类名后缀

FileOutputStream 文件输出流用于写数据写入File

- `FileOutputStream(String name)` 创建文件输出流以指定的名称写入文件

字节输出流写数据的步骤

- 创建字节输出流对象（调用系统功能创建了文件，创建字节输出流对象，让字节输出流指向文件）
- 调用字节输出流对象的写数据方法
- 释放资源（关闭此文件输出流并释放与此流相关联的任何系统资源）

2.3 字节流写数据的三种方式

方法名	说明
void write(int b)	将指定的字节写入此文件输出流 一次写一个字节数据
void write(byte[] b)	将 b.length字节从指定的字节数组写入此文件输出流 一次写一个字节数组数据
void write(byte[] b, int off, int len)	将 len字节从指定的字节数组开始，从偏移量off开始写入此文件输出流 一次写一个字节数组的部分数据

2.4 字节流写数据的两个小问题

如何换行?

- 加换行符
  - Win `\r\n`
  - linux `\n`
  - mac `\r`

如何实现追加写入而非重头写

- `public FileOutputStream(String name,boolean append)`
- 创建输出流以指定的名称写入文件，如果第二个参数`true`，则写入文件的末尾而不是开头

2.5 字节流写数据加异常处理

finally 在异常处理时候提供finally来执行所有的清楚操作，比如释放IO流资源

特点：finally语句一定会执行，除非JVM退出

```
try{
    可能出现异常的代码;
}catch(异常类名 变量名){
    异常的处理代码;
}finally{
    执行所有清除操作;
}
```

2.6 字节流读数据（一次读一个字节 | 一个字节数组）

> FileInputStream 从文件系统中获取输入字节

- `FileInputStream(String name)`通过打开与实际文件的连接来创建一个FileInputStream，该文件由文件兄同种的路径名name命名

使用字节输入流读数据的步骤

- 创建字节输入流对象
- 调用字节输入流对象的读数据方法
- 释放资源

2.7 字节缓冲流

- BufferOutputStream 该类实现缓冲输出流，通过设置这样一个输出流，应用程序可以向底层输出流写入字节，而不必为写入的每个字节导致底层系统的调用
- BufferInputStream 创建BufferInputStream 将创建一个内部缓冲区数组，当从流中读取或跳过字节时，内部缓冲区将根据需要从所包含的输入流中重新填充，一次很多字节

构造方法

- 字节缓冲输出流 `BufferOutputStream(OutputStream out)`
- 字节缓冲输入流 `BufferInputStream(InputStream in)`

为什么构造方法需要的时字节流而非文件or路径

- 因为缓冲流仅仅提供缓冲区域，最终真正读写数据的还是基本的字节流对象