

第十七章 IO流

17.1 IO流的概念

- IO就是Input和Output的简写，也就是输入和输出的含义。
- IO流就是指读写数据时像流水一样从一端流到另外一端，因此得名为“流”。

17.2 基本分类

- 按照读写数据的基本单位不同，分为 字节流 和 字符流。
其中字节流主要指以字节为单位进行数据读写的流，可以读写任意类型的文件。
其中字符流主要指以字符(2个字节)为单位进行数据读写的流，只能读写文本文件。
- 按照读写数据的方向不同，分为 输入流 和 输出流（站在程序的角度）。
其中输入流主要指从文件中读取数据内容输入到程序中，也就是读文件。
其中输出流主要指将程序中的数据内容输出到文件中，也就是写文件。
- 按照流的角色不同分为节点流和处理流。
其中节点流主要指直接和输入输出源对接的流。
其中处理流主要指需要建立在节点流的基础之上的流。

17.3 体系结构

分类	字节输入流	字节输出流	字符输入流	字符输出流
抽象基类	InputStream	OutputStream	Reader	Writer
访问文件	FileInputStream	FileOutputStream	FileReader	FileWriter
访问数组	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
访问管道	PipedInputStream	PipedOutputStream	PipedReader	PipedWriter
访问字符串	--	--	StringReader	StringWriter
缓冲流	BufferedInputStream	BufferedOutputStream	BufferedReader	BufferedWriter
转换流	--	--	InputStreamReader	OutputStreamWriter
对象流	ObjectInputStream	ObjectOutputStream	--	--
	FilterInputStream	FilterOutputStream	FilterReader	FilterWriter
打印流	--	PrintStream	--	PrintWriter
推回输入流	PushbackInputStream	--	PushbackReader	--
特殊流	DataInputStream	DataOutputStream	--	--

17.4 相关流的详解

17.4.1 FileWriter类（重点）

（1）基本概念

- java.io.FileWriter类主要用于将文本内容写入到文本文件。

（2）常用的方法

方法声明	功能介绍
FileWriter(String fileName)	根据参数指定的文件名构造对象
FileWriter(String fileName, boolean append)	以追加的方式根据参数指定的文件名来构造对象
void write(int c)	写入单个字符
void write(char[] cbuf, int off, int len)	将指定字符数组中从偏移量off开始的len个字符写入此文件输出流
void write(char[] cbuf)	将cbuf.length个字符从指定字符数组写入此文件输出流中
void flush()	刷新流
void close()	关闭流对象并释放有关的资源

17.4.2 FileReader类（重点）

（1）基本概念

- java.io.FileReader类主要用于从文本文件读取文本数据内容。

（2）常用的方法

方法声明	功能介绍
FileReader(String fileName)	根据参数指定的文件名构造对象
int read()	读取单个字符的数据并返回，返回-1表示读取到末尾
int read(char[] cbuf, int offset, int length)	从输入流中将最多len个字符的数据读入一个字符数组中，返回读取到的字符个数，返回-1表示读取到末尾
int read(char[] cbuf)	从此输入流中将最多 cbuf.length 个字符的数据读入字符数组中，返回读取到的字符个数，返回-1表示读取到末尾
void close()	关闭流对象并释放有关的资源

17.4.3 FileOutputStream类（重点）

（1）基本概念

- java.io.FileOutputStream类主要用于将图像数据之类的原始字节流写入到输出流中。

（2）常用的方法

方法声明	功能介绍
FileOutputStream(String name)	根据参数指定的文件名来构造对象
FileOutputStream(String name, boolean append)	以追加的方式根据参数指定的文件名来构造对象
void write(int b)	将指定字节写入此文件输出流
void write(byte[] b, int off, int len)	将指定字节数组中从偏移量off开始的len个字节写入此文件输出流
void write(byte[] b)	将 b.length 个字节从指定字节数组写入此文件输出流中
void flush()	刷新此输出流并强制写出任何缓冲的输出字节
void close()	关闭流对象并释放有关的资源

17.4.4 FileInputStream类（重点）

（1）基本概念

- java.io.FileInputStream类主要用于从输入流中以字节流的方式读取图像数据等。

（2）常用的方法

方法声明	功能介绍
FileInputStream(String name)	根据参数指定的文件路径名来构造对象
int read()	从输入流中读取单个字节的数据并返回，返回-1表示读取到末尾
int read(byte[] b, int off, int len)	从此输入流中将最多len个字节的数据读入字节数组中，返回读取到的字节个数，返回-1表示读取到末尾
int read(byte[] b)	从此输入流中将最多 b.length 个字节的数据读入字节数组中，返回读取到的字节个数，返回-1表示读取到末尾
void close()	关闭流对象并释放有关的资源
int available()	获取输入流所关联文件的大小

- 案例题目
编程实现两个文件之间的拷贝功能。

17.4.5 BufferedOutputStream类（重点）

（1）基本概念

- java.io.BufferedOutputStream类主要用于描述缓冲输出流，此时不用为写入的每个字节调用底层系统。

（2）常用的方法

方法声明	功能介绍
BufferedOutputStream(OutputStream out)	根据参数指定的引用来构造对象
BufferedOutputStream(OutputStream out, int size)	根据参数指定的引用和缓冲区大小来构造对象
void write(int b)	写入单个字节
void write(byte[] b, int off, int len)	写入字节数组中的一部分数据
void write(byte[] b)	写入参数指定的整个字节数组
void flush()	刷新流
void close()	关闭流对象并释放有关的资源

17.4.6 BufferedInputStream类（重点）

（1）基本概念

- java.io.BufferedInputStream类主要用于描述缓冲输入流。

（2）常用的方法

方法声明	功能介绍
BufferedInputStream(InputStream in)	根据参数指定的引用构造对象
BufferedInputStream(InputStream in, int size)	根据参数指定的引用和缓冲区大小构造对象
int read()	读取单个字节
int read(byte[] b, int off, int len)	读取len个字节
int read(byte[] b)	读取b.length个字节
void close()	关闭流对象并释放有关的资源

17.4.7 BufferedWriter类（重点）

（1）基本概念

- java.io.BufferedWriter类主要用于写入单个字符、字符数组以及字符串到输出流中。

（2）常用的方法

方法声明	功能介绍
BufferedWriter(Writer out)	根据参数指定的引用来构造对象
BufferedWriter(Writer out, int sz)	根据参数指定的引用和缓冲区大小来构造对象
void write(int c)	写入单个字符到输出流中
void write(char[] cbuf, int off, int len)	将字符数组cbuf中从下标off开始的len个字符写入输出流中
void write(char[] cbuf)	将字符串数组cbuf中所有内容写入输出流中
void write(String s, int off, int len)	将参数s中下标从off开始的len个字符写入输出流中
void write(String str)	将参数指定的字符串内容写入输出流中
void newLine()	用于写入行分隔符到输出流中
void flush()	刷新流
void close()	关闭流对象并释放有关的资源

17.4.8 BufferedReader类（重点）

（1）基本概念

- java.io.BufferedReader类用于从输入流中读取单个字符、字符数组以及字符串。

（2）常用的方法

方法声明	功能介绍
BufferedReader(Reader in)	根据参数指定的引用来构造对象
BufferedReader(Reader in, int sz)	根据参数指定的引用和缓冲区大小来构造对象
int read()	从输入流读取单个字符，读取到末尾则返回-1，否则返回实际读取到的字符内容
int read(char[] cbuf, int off, int len)	从输入流中读取len个字符放入数组cbuf中下标从off开始的位置上，若读取到末尾则返回-1，否则返回实际读取到的字符个数
int read(char[] cbuf)	从输入流中读满整个数组cbuf
String readLine()	读取一行字符串并返回，返回null表示读取到末尾
void close()	关闭流对象并释放有关的资源

17.4.9 PrintStream类

（1）基本概念

- java.io.PrintStream类主要用于更加方便地打印各种数据内容。

（2）常用的方法

方法声明	功能介绍
PrintStream(OutputStream out)	根据参数指定的引用来构造对象
void print(String s)	用于将参数指定的字符串内容打印出来
void println(String x)	用于打印字符串后并终止该行
void flush()	刷新流
void close()	用于关闭输出流并释放有关的资源

17.4.10 PrintWriter类

(1) 基本概念

- java.io.PrintWriter类主要用于将对象的格式化形式打印到文本输出流。

(2) 常用的方法

方法声明	功能介绍
PrintWriter(Writer out)	根据参数指定的引用来构造对象
void print(String s)	将参数指定的字符串内容打印出来
void println(String x)	打印字符串后并终止该行
void flush()	刷新流
void close()	关闭流对象并释放有关的资源

• 案例题目

不断地提示用户输入要发送的内容，若发送的内容是"bye"则聊天结束，否则将用户输入的内容写入到文件d:/a.txt中。

要求使用BufferedReader类来读取键盘的输入 System.in代表键盘输入

要求使用PrintStream类负责将数据写入文件

17.4.11 OutputStreamWriter类

(1) 基本概念

- java.io.OutputStreamWriter类主要用于实现从字符流到字节流的转换。

(2) 常用的方法

方法声明	功能介绍
OutputStreamWriter(OutputStream out)	根据参数指定的引用来构造对象
OutputStreamWriter(OutputStream out, String charsetName)	根据参数指定的引用和编码构造对象
void write(String str)	将参数指定的字符串写入
void flush()	刷新流
void close()	用于关闭输出流并释放有关的资源

17.4.12 InputStreamReader类

(1) 基本概念

- java.io.InputStreamReader类主要用于实现从字节流到字符流的转换。

(2) 常用的方法

方法声明	功能介绍
InputStreamReader(InputStream in)	根据参数指定的引用来构造对象
InputStreamReader(InputStream in, String charsetName)	根据参数指定的引用和编码来构造对象
int read(char[] cbuf)	读取字符数据到参数指定的数组
void close()	用于关闭输出流并释放有关的资源

17.4.13 字符编码

(1) 编码表的由来

- 计算机只能识别二进制数据，早期就是电信号。为了方便计算机可以识别各个国家的文字，就需要将各个国家的文字采用数字编号的方式进行描述并建立对应的关系表，该表就叫做编码表。

(2) 常见的编码表

- ASCII：美国标准信息交换码，使用一个字节的低7位二进制进行表示。
- ISO8859-1：拉丁码表，欧洲码表，使用一个字节的8位二进制进行表示。
- GB2312：中国的中文编码表，最多使用两个字节16位二进制为进行表示。
- GBK：中国的中文编码表升级，融合了更多的中文文字符号，最多使用两个字节16位二进制位表示。
- Unicode：国际标准码，融合了目前人类使用的所有字符，为每个字符分配唯一的字符码。所有的文字都用两个字节16位二进制位来表示。

(3) 编码的发展

- 面向传输的众多 UTF (UCS Transfer Format) 标准出现了，UTF-8就是每次8个位传输数据，而 UTF-16就是每次16个位。这是为传输而设计的编码并使编码无国界，这样就可以显示全世界上所有文化的字符了。

- Unicode只是定义了一个庞大的、全球通用的字符集，并为每个字符规定了唯一确定的编号，具体存储成什么样的字节流，取决于字符编码方案。推荐的Unicode编码是UTF-8和UTF-16。
- UTF-8：变长的编码方式，可用1-4个字节来表示一个字符。

17.4.14 DataOutputStream类（了解）

（1）基本概念

- java.io.DataOutputStream类主要用于以适当的方式将基本数据类型写入输出流中。

（2）常用的方法

方法声明	功能介绍
DataOutputStream(OutputStream out)	根据参数指定的引用构造对象 OutputStream类是个抽象类，实参需要传递子类对象
void writeInt(int v)	用于将参数指定的整数一次性写入输出流，优先写入高字节
void close()	用于关闭文件输出流并释放有关的资源

17.4.15 DataInputStream类（了解）

（1）基本概念

- java.io.DataInputStream类主要用于从输入流中读取基本数据类型的数据。

（2）常用的方法

方法声明	功能介绍
DataInputStream(InputStream in)	根据参数指定的引用来构造对象 InputStream类是抽象类，实参需要传递子类对象
int readInt()	用于从输入流中一次性读取一个整数数据并返回
void close()	用于关闭文件输出流并释放有关的资源

17.4.16 ObjectOutputStream类（重点）

（1）基本概念

- java.io.ObjectOutputStream类主要用于将一个对象的所有内容整体写入到输出流中。
- 只能将支持 java.io.Serializable 接口的对象写入流中。
- 类通过实现 java.io.Serializable 接口以启用其序列化功能。
- 所谓序列化主要指将一个对象需要存储的相关信息有效组织成字节序列的转化过程。

（2）常用的方法

方法声明	功能介绍
ObjectOutputStream(OutputStream out)	根据参数指定的引用来构造对象
void writeObject(Object obj)	用于将参数指定的对象整体写入到输出流中
void close()	用于关闭输出流并释放有关的资源

17.4.17 ObjectOutputStream类（重点）

（1）基本概念

- java.io.ObjectOutputStream类主要用于从输入流中一次性将对象整体读取出来。
- 所谓反序列化主要指将有效组织的字节序列恢复为一个对象及相关信息的转化过程。

（2）常用的方法

方法声明	功能介绍
ObjectInputStream(InputStream in)	根据参数指定的引用来构造对象
Object readObject()	主要用于从输入流中读取一个对象并返回 无法通过返回值来判断是否读取到文件的末尾
void close()	用于关闭输入流并释放有关的资源

（3）序列化版本号

- 序列化机制是通过在运行时判断类的serialVersionUID来验证版本一致性的。在进行反序列化时，JVM会把传来的字节流中的serialVersionUID与本地相应实体类的serialVersionUID进行比较，如果相同就认为是一致的，可以进行反序列化，否则就会出现序列化版本不一致的异常（InvalidCastException）。

（4）transient关键字

- transient是Java语言的关键字，用来表示一个域不是该对象串行化的一部分。当一个对象被串行化的时候，transient型变量的值不包括在串行化的表示中，然而非transient型的变量是被包括进去的。

（5）经验的分享

- 当希望将多个对象写入文件时，通常建议将多个对象放入一个集合中，然后将集合这个整体看做一个对象写入输出流中，此时只需要调用一次readObject方法就可以将整个集合的数据读取出来，从而避免了通过返回值进行是否达到文件末尾的判断。

17.4.18 RandomAccessFile类

（1）基本概念

- java.io.RandomAccessFile类主要支持对随机访问文件的读写操作。

（2）常用的方法

方法声明	功能介绍
RandomAccessFile(String name, String mode)	根据参数指定的名称和模式构造对象 r: 以只读方式打开 rw : 打开以便读取和写入 rwd:打开以便读取和写入，同步文件内容的更新 rws:打开以便读取和写入，同步文件内容和元数据的更新
int read()	读取单个字节的数据
void seek(long pos)	用于设置从此文件的开头开始测量的文件指针偏移量
void write(int b)	将参数指定的单个字节写入
void close()	用于关闭流并释放有关的资源