

Java语言基础

【Day15】

10流

IO流的概念和分类

- 流 - 输入或输出时，读写数据象流水一样，不间断，得名为流。
- 流的分类：
 - 1、字节流和字符流

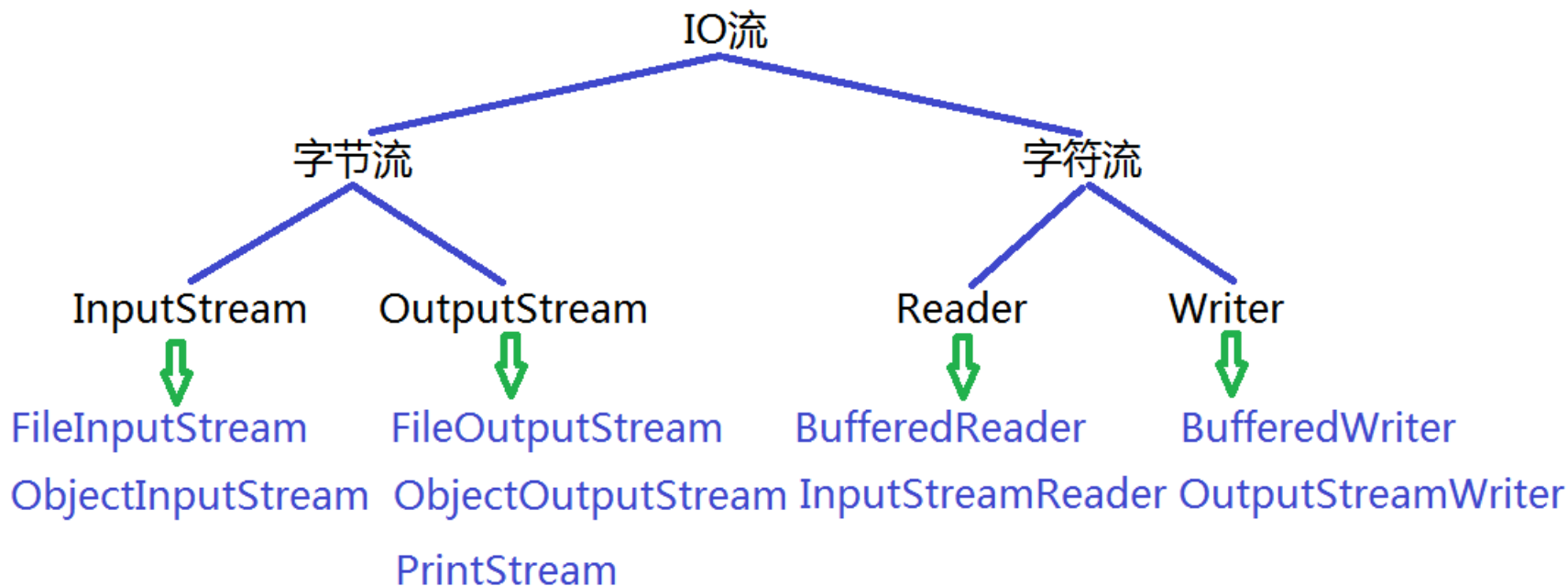
字节流 - 读写的基本单位是按 字节 进行，各种类型的文件都可以读写。

字符流 - 读写的基本单位是按 字符(双字节) 进行，只能读写文本文件。
 - 2、输入流和输出流

站在程序的角度出发的，读 就是 输入，写 就是 输出。
 - 3、节点流和包装流

IO流的框架结构

- IO流的框架结构如下：



FOS和FIS

FileOutputStream类的概述

- 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。
- 常用的构造方法如下：

`FileOutputStream(String name)`

- 根据参数指定的文件名来构造对象。

`FileOutputStream(String name, boolean append)`

- 表示以追加的方式根据参数指定的文件名来构造对象。

FileOutputStream类的方法

- FileOutputStream类的常用成员方法如下：

<code>void write(int b)</code>	将指定字节写入此文件输出流。
<code>void write(byte[] b, int off, int len)</code>	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。
<code>void write(byte[] b)</code>	将 b.length 个字节从指定字节数组写入此文件输出流中
<code>void close()</code>	用于关闭文件输出流并释放有关的资源

项目案例

- 编程实现以字节的方式向文件中写入数据。

FileInputStream类的概述

- 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
- 常用的构造方法如下：

FileInputStream(String name)

- 根据参数指定的文件路径名来构造对象。

FileInputStream类的方法

- FileInputStream类的常用成员方法如下：

<code>int read()</code>	读取一个byte 无符号填充到int低八位,-1是EOF
<code>int read(byte[] b)</code>	从此输入流中将最多 <code>b.length</code> 个字节的数据读入一个字节数组中，返回读取的字节的个数
<code>int read(byte[] b, int off, int len)</code>	从此输入流中将最多 <code>len</code> 个字节的数据读入一个字节数组中
<code>void close()</code>	用于关闭文件输出流并释放有关的资源

项目案例

- 编程实现以字节的方式从文件中读取数据。

项目案例

- 编程使用字节输出流和字节输入流实现文件的拷贝功能。

PS和BR

PrintStream类

PrintStream类的概述

- java.io.PrintStream类主要用于更加方便地打印各种数据内容。
- 常用的构造方法如下：
PrintStream(OutputStream out) - 根据参数指定的引用来构造对象。
 - 其中OutputStream类是个抽象类，实参需要传递子类的对象。

PrintStream类的方法

- PrintStream类的常用成员方法如下：

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 close()	用于关闭文件输出流并释放有关的资源

项目案例

- 编程实现以字符串的方式向文件中写入数据。

BufferedReader类的概述

- 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。
- 常用的构造方法如下：

BufferedReader(Reader in) - 根据参数指定的引用来构造对象。

- 其中Reader类是个抽象类，实参需要传递子类的对象。

BufferedReader类的方法

- BufferedReader类的常用成员方法如下：

String readLine()	用于读取一行字符串并返回
void close()	用于关闭文件输出流并释放有关的资源

项目案例

- 编程实现以字符的方式从文件中读取数据。

OIS和OOS

ObjectOutputStream类的概述

- 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。

ObjectOutputStream类的方法

- ObjectOutputStream类的常用方法如下：

ObjectOutputStream(OutputStream out)	根据参数指定引用来构造对象 其中OutputStream类是个抽象类，因此实参需要传递子类对象
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()	用于关闭文件输出流并释放有关的资源

项目案例

- 编程实现以对象的方式向文件中写入数据。

ObjectInputStream类的概述

- java.io.ObjectInputStream类主要用于从输入流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。

ObjectInputStream类的方法

- ObjectInputStream类的常用方法如下：

ObjectInputStream(InputStream in)	根据参数指定的引用来构造对象 其中InputStream类是个抽象类，实参需要传递子类的对象
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()	用于关闭文件输出流并释放有关的资源

项目案例

- 编程实现以对象的方式从文件中读取数据。

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。

总结与答疑

技术资料





IT兄弟连
ITXDL.CN

变态严管 让学习成为一种习惯