

Java IO2

- What?Why?How?



本章概述

- File类
- IO流的原理及概念
- IO流的分类
- IO流类的体系
- 字节流和字符流
- 处理流
- 文件拷贝



上机练习

- 将D盘已存在的文件moring.txt中的内容复制到C:盘根目录
- 可以使用InputStream, OutputStream字节流的读写
- 也可以使用Reader , Writer字符流的读写

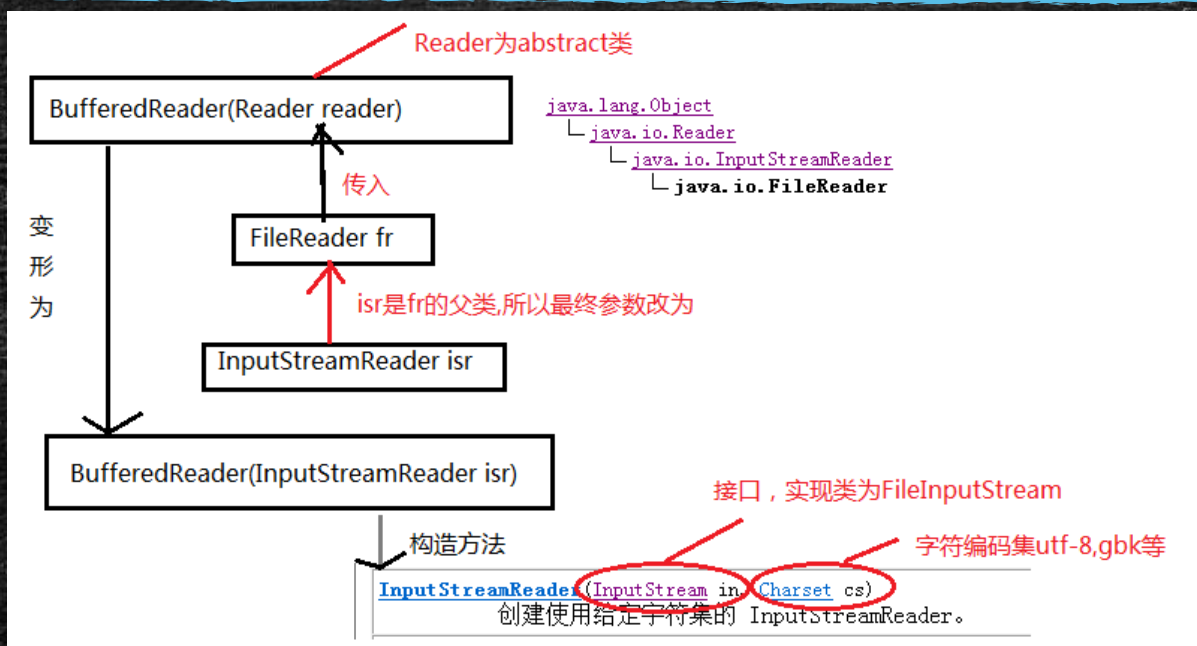


Reader与Writer的编码方式

- 获得当前开发环境的字符编码方式
- `System.out.println(System.getProperty("file.encoding"));`
- 字符流的读写根据需要设置编码方式
- 涉及到的类:
- 读: `FileReader (File file)`
- 写: `FileWriter(File file)`
- 加入缓冲区的读: `BufferedReader(Reader fr)`
- 加入缓冲区的写: `BufferedWriter(Writer bw)`



Reader与Writer的编码方式



调用顺序与编码顺序相反，
所以创建类的依次顺序为
FileInputStream
InputStreamReader
BufferedReader

```
FileInputStream fis=new FileInputStream("文件的路径");  
InputStreamReader isr=new InputStreamReader(fis,"utf-8");  
BufferedReader br=new BufferedReader(isr);
```



上机练习

- 需求：将<http://www.baidu.com>的号文件
- 以特定的编码方式读取到控制台输出
- **核心代码：**
- `URL url=new URL("http://www.baidu.com");`
- `InputStream is=url.openStream();`



标准的输入输入出流System.in, System.out

- Scanner input=new Scanner(System.in);

分析顺序 ↓

对象的创建顺序 ↑

Reader的子类InputStreamReader

BufferedReader(InputStreamReader)

子类替换父类对象 (多态)

InputStreamReader的构造方法需要一个输入流做为参数

InputStreamReader(InputStream in)

而java.lang包中的System类中的静态字段System.in就是一个标准的输入流

(1) InputStreamReader isr=new InputStreamReader(System.in);

(2) BufferedReader br=new BufferedReader(isr);

第二种方式, 匿名对象

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

java.lang

- java.lang.annotation
- java.lang.instrument
- java.lang.management
- java.lang.ref
- java.lang.reflect

Enum

Float

InheritableThreadLocal

Integer

Long

Math

Number

Object

Package

Process

ProcessBuilder

Runtime

RuntimePermission

SecurityManager

Short

StackTraceElement

StrictMath

String

StringBuilder

System

Thread

ThreadGroup

static InputStream in

static PrintStream out

方法摘要

static void	ar
	指
static String	cl
static Console	cc
static long	cu
static void	ex
static void	gc
static Map<String, String>	gc



上机练习

使用键盘录入(System.in)，将如下诗歌写入文件

《虞美人》
春花秋月何时了？
往事知多少。
小楼昨夜又东风，
故国不堪回首月明中。
雕栏玉砌应犹在，
只是朱颜改。
问君能有几多愁？
恰似一江春水向东流。

再将写入的文件读取到控制台输出System.out



总结

- 流操作的基本规律
- 1、明确源和目的
 - a)源: 输入流 `InputStream` `Reader`
 - b)目的:输出流 `OutputStream` `Writer`
- 2、操作的数据是否是纯文本
 - 是:字符流 `Reader,Writer`
 - 不是:字节流 `InputStream,OutputStream`
- 3、具体的操作设备
 - 源设备:内存, 硬盘, 键盘
 - 目的设备:内存, 硬盘, 控制台



操作对象的流

- 序列化对象: `ObjectOutputStream`
- 反序列化对象: `ObjectInputStream`
- 序列化:
 - `ObjectOutputStream oos=new ObjectOutputStream(new FileOutputStream("obj.txt"));`
 - `oos.writeObject(new Person("张三",19));` // 序列化
 - `oos.close();`
- 反序列化:
 - `ObjectInputStream ois=new ObjectInputStream(new FileInputStream("obj.txt"));`
 - `Person p=(Person)ois.readObject();` // 反序列化
 - `System.out.println(p);`



常见问题

- 1、类必须实现Serializable接口
- 2、给类加个序列化编号，给类定义一个标记，
新的修改后的类还可以操作曾经序列化的对象
- 3、静态是不能被序列化的，
序列化只能对堆中的进行序列化，不能对“方法区”中的进行序列化
- 4、不需要序列化的字段前加 transient



上机练习

【1】创建User类，包含以下属性name:String,age:int
gender enum,重写toString方法显示对象的信息

【2】序列化对象到文件

【3】反序列化对象，调用

对象的toString方法输出对象的信息

操作要求：初始化几个对象，存入集合,序列化与反序列化集合



操作基本数据的流对象 **DataStream**

写入数据

```
DataOutputStream dos=new DataOutputStream(new FileOutputStream("data.txt"));  
dos.writeInt(234);  
dos.writeBoolean(false);  
dos.writeDouble(9943.00);  
dos.writeUTF("中国");  
dos.close();
```

读取数据

```
DataInputStream dis=new DataInputStream(new FileInputStream("data.txt"));  
int num=dis.readInt();  
boolean isFind=dis.readBoolean();  
double price=dis.readDouble();  
String str=dis.readUTF();  
System.out.println(num+"\t"+isFind+"\t"+price+"\t"+str);
```



总结

文本操作: FileReader, FileWriter
字符操作: FileInputStream, FileOutputStream
基本数据类型操作: DataInputStream, DataOutputStream
操作对象: ObjectInputStream, ObjectOutputStream

