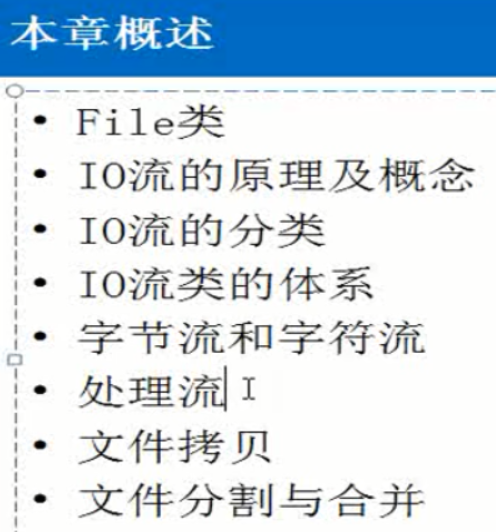
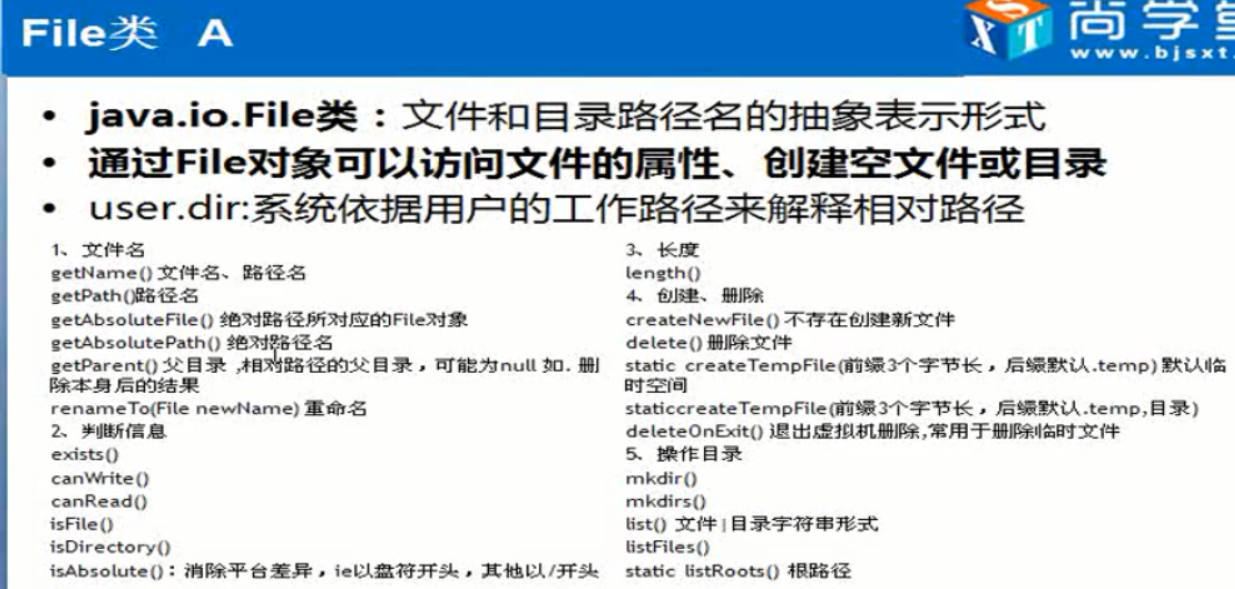
P166 概述;File类



P167 File类常用方法



P168 文件夹操作;命令模式查找

P169 IO流;原理;分类;标准步骤

1. 概念

流:流动,流向,从一端流向另一端. 源头与目的地

程序与文件|数组|网络连接 |数据库,以程序为中心

1. IO流分类
2. 流向:输入流与输出流
3. 数据:字节流:二进制,可以处理一切文件,包括纯文本,doc,音频,视频

字符流:文本文件,只能处理纯文本

1. 功能:节点:包裹源头

处理:增强功能,提高性能

1. 字符流与字节流(重点)
2. 字节流

输入流:InputStream

[read](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html#read(byte[]))(byte[] b)

[read](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html#read(byte[], int, int))(byte[] b, int off, int len) +[close](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html#close())()

FileInputStream()

输出流:OutputStream

[write](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/OutputStream.html#write(byte[]))(byte[] b)

[write](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/OutputStream.html#write(byte[], int, int))(byte[] b, int off, int len) +[flush](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/OutputStream.html#flush())()+[close](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html#close())()

FileOutputStream()

1. 字符流

输入流:Reader

[read](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Reader.html#read(char[]))(char[] cbuf)

[read](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Reader.html#read(char[], int, int))(char[] cbuf, int off, int len) +[close](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html#close())()

FileReader()

输出流:Writer

[write](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Writer.html#write(char[]))(char[] cbuf)

[write](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Writer.html#write(char[], int, int))(char[] cbuf, int off, int len) +[flush](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/OutputStream.html#flush())() +[close](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/InputStream.html#close())()

[write](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/Writer.html#write(java.lang.String, int, int))([String](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/lang/String.html) str, int off, int len)

FileWriter()

1. 操作
2. 举例:搬家 🡪读取文件

1)关联房子 🡪建立与文件的联系

2)选择搬家 🡪选择对应流

3)搬家 🡪读取|写出

a)卡车大小 🡪数组大小

b)运输 🡪读取,写出

4)打发over 🡪释放资源

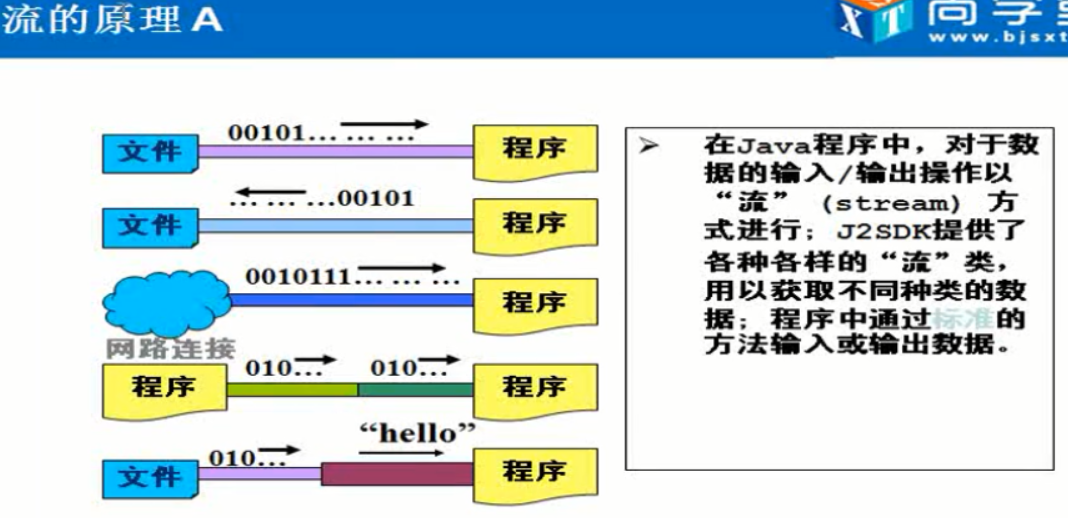
2.操作

1)建立联系

2)选择流

3)操作 数组大小+read,write

4)释放资源



P170 字节流;节点流;文件读取,写出;追加文件

字节流:可以处理一切文件,包括二进制,音频,视频,doc等

1. 读取文件

1.建立联系 File对象 源头

2.选择流 文件输入流 InputStream FileInputStream

3.操作 :byte[] car = new byte[1024]; +read+读取大小

输出

4.释放资源:关闭

1. 写出文件

1.建立联系 File对象 目的地

2.选择流 文件输出流 OutputStream FileOutputStream

3.操作 :write() +flush

4.释放资源:关闭

P171 文件的拷贝

三.文件拷贝 程序为桥梁

1.建立联系 File对象 源头 目的地

2.选择流 文件输入流 InputStream FileInputStream

文件输出流 OutputStream FileOutputStream

3.操作:拷贝

Byte[] car = new byte[1024];

int len = 0;

While(-1!=(len=输入流.read(flush))){

输出流.write(flush,0,len)

}

输出流.flush

4.释放资源:关闭两个流

P172 文件夹拷贝;工作制作;文件后缀与软件的关系

四.文件夹拷贝

1. 递归查找子孙级文件|文件夹
2. 文件 复制(IO流复制)

文件夹 创建

P174 字符流;节点流;Reader;FileReader

字符流:只能处理纯文本,全部为可见字符 .txt .html

节点流: Reader FileReader

Writer FileWriter

1. 文件的读取
2. 建立联系
3. 选择流 Reader FileReader
4. 读取 char[] flush = new char[1024]
5. 关闭
6. 纯文本写出

1.建立联系

2.选择流 Writer FileWriter

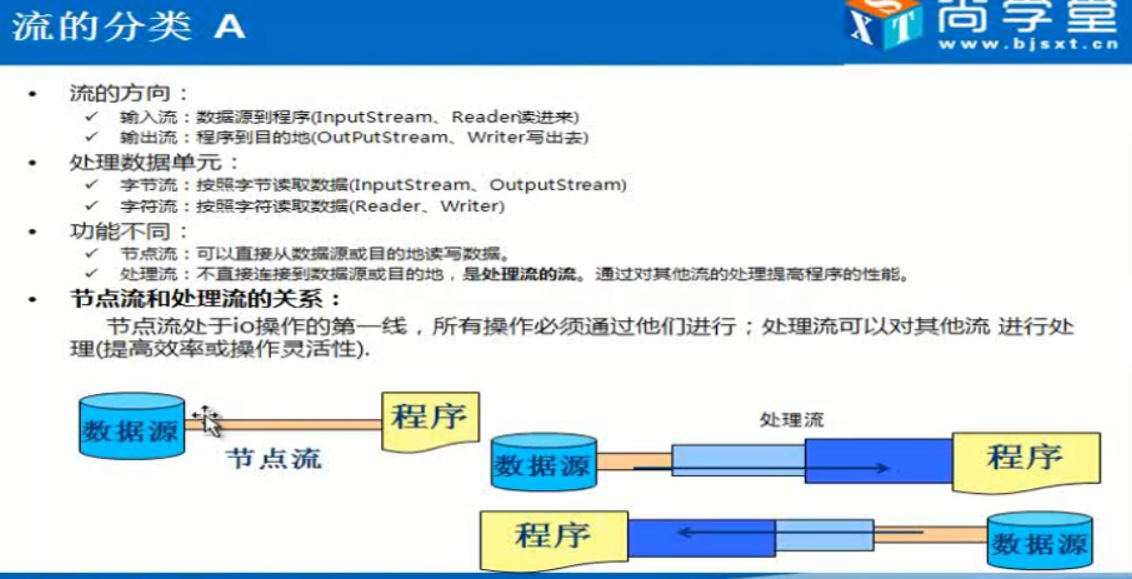
3.读取 write(字符数组,0,长度) +flush

write(字符串)

append(字符|字符串)

4.关闭

P175 缓冲流;BufferedInputStream; BufferedOutputStream; BufferedReader; BufferedWriter;



处理流:增强功能,提高性能,节点流之上

一.缓冲流

1)字节缓冲流

BufferedInputStream

BufferedOutputStream

2)字符缓冲流

BufferedReader [**readLine**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/BufferedReader.html#readLine())()

BufferedWriter [**newLine**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/BufferedWriter.html#newLine())()

P176 转换流;字节转换为字符;乱码分析

二.转换流:字节流转为字符流 处理乱码(编码集,解码集)

1.编码与解码概念

编码: 字符 🡪编码字符集🡪 二进制

解码: 二进制 🡪解码字符集🡪 字符

2.乱码

1. 编码与解码的字符集不统一
2. 字节缺少,长度丢失
3. 文件乱码

P177 字节转换为字符;

输出流:InputStreamReader

输入流:OutputStreamWriter

P178 重点流;总结

P179 其他流;字节数组流

1. 节点流
2. 字节数组

输入流:ByteArrayInputStream [read](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/ByteArrayInputStream.html#read(byte[], int, int))(byte[] b, int off, int len) +close()

输出流: [ByteArrayOutputStream](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/ByteArrayOutputStream.html#ByteArrayOutputStream()) [write](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/ByteArrayOutputStream.html#write(byte[], int, int))(byte[] b, int off, int len)+ [**toByteArray**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/ByteArrayOutputStream.html#toByteArray())() 不要使用多态

P181

1. 处理流
2. 基本数据类型+String 保留数据+类型

输入流: DataInputStream readXxx

输出流:DataOutputStream writeXxx

P182

1. 应用类型(对象)+String 保留数据+类型

反序列化 输入流:ObjectInputStream **[readObject](mk:@MSITStore:C:\\Users\\Hasee\\Desktop\\JAVA_API_1.7中文.chm::/java/io/ObjectInputStream.html" \l "readObject())**()

序列化 输出流:ObjectOutputStream [**writeObject**](mk:@MSITStore:C:\Users\Hasee\Desktop\JAVA_API_1.7中文.chm::/java/io/ObjectInputStream.html#readObject())()

注意:

1. 先序列化后反序列化, 反序列化顺序必须与序列化一致
2. 不是所有对象都可以序列化, java.io.Serializable

不是所有属性都需要序列化,transient

P183 关闭流方法

P184 关闭流方法

P185 装饰设计模式

类与类之间的关系

1. 依赖:形参|局部变量
2. 关联:属性

聚合:属性整体与部分 不一致的生命周期 人与手

组合: 属性整体与部分 一致的生命周期 人与大脑

1. 继承:父子类关系
2. 实现:接口与实现类关系

P186 拷贝文件夹bug修复;如何删除超长文件夹

P187 文件分割与合并 RamdonAccessFile

文件分隔思路:

1. 分割的块数
2. 每一块的大小

P188初始化各项参数

P189分割

P190 文件合并 SequenceInputStream

P191 IO流总结