

Day09

字节流 操作byte

字符流 操作char String

输入流 读

输出流 写

字节输出流写出数据到文件：

//1. 创建对象

```
FileOutputStream fos = new FileOutputStream( "文件路径");
```

//2. 写数据

```
fos.write(字节/字节数组);
```

//释放资源

```
fos.close();
```

字节输入流读取数据到程序

// 1. 创建流对象

```
FileInputStream fis = new FileInputStream("文件路径");
```

// 2. 读数据

```
int by (数据) = fis.read(); // 一次读一个字节
```

```
int len (个数) = fis.read(数组); // 一次读一个字节数组
```

// 3. 释放资源

```
fis.close();
```

read(byte[])方法的原理

```
byte[] bytes = new byte[2]; ABCDE
```

```
read(bytes); // AB 2个字节
```

```
read(bytes); // CD 2个字节
```

```
read(bytes); // ED 1个字节
```

字节流完成文件的复制

边读边写

// 1创建2个流对象 输入流 输出流

```
FileInputStream fis = ...; // 数据源
```

```

FileOutputStream fos = ..;    // 目的地
// 循环读写
// 一次读写一个字节
int by;
while ((by = fis.read()) != -1) {
    fos.write(by);
}
// 一次读写一个字节数组
byte[] bytes = new byte[1024];
int len;
while ((len = fis.read(bytes)) != -1) {
    fos.write(bytes, 0, len);
}
// 释放资源
close();

```

FileWriter中关闭和刷新方法的区别

close(): 刷新缓冲区 并释放资源. 不能操作流对象

flush(): 刷新缓冲区. 能操作流对象

使用FileWriter写数据的5个方法

char

char[]

String

char[]/String的一部分

使用FileWriter写数据实现换行和追加写

windows: "\r\n"

追加写: boolean append构造 传true

FileReader读数据

1. 创建流对象

2. read()

char

char[]

3. 释放资源

使用FileReader读数据一次一个字符数组

```
char[] chs = new char[1024];
int len;
while ((len = fr.read(chs)) != -1) {
}
```

使用Properties的load方法加载文件中配置信息

1. 创建Properties对象
2. 用Properties对象调用load(字符流对象)
3. 获取键值对

Day10

使用字节缓冲流读取数据到程序

```
BufferedInputStream bis = new BufferedInputStream(基本流对象FileInputStream);
bis.read();
bis.close();
```

使用字节缓冲流写出数据到文件

```
BufferedOutputStream bos = new BufferedOutputStream(基本流对象FileOutputStream);
bos.write();
bos.close();
```

字符串缓冲流的作用和基本用法

BufferedReader/BufferedWriter

提高基本流的读写效率

```
BufferedReader br = new BufferedReader(基本流FileReader);
br.read();
br.close();
```

使用缓冲流的特殊功能

BufferedReader

String readLine(): 不会返回换行符

BufferedWriter

`void newLine()`: 可以根据系统自动确定换行符

阐述编码表的意义

编码表:存储 字符 和 数字 的对应关系

使用转换流读取指定编码的文本文件

InputStreamReader

`new InputStreamReader(基本流, "编码")`

使用转换流写入指定编码的文本文件

OutputStreamWriter

`new OutputStreamWriter(基本流, "编码")`

打印流的特点

1. 只有输出, 没有输入.
2. 不会抛IOException
3. 有特有的方法 `print()` `println()` 原样输出

Day11

辨别UDP和TCP协议特点

UDP

1. 无连接的不可靠协议.
2. 以包发送, 64K一个包.
3. 速度快效率高, 容易丢包.

TCP

1. 有连接的可靠协议.
2. 传输数据没有大小限制
3. 速度慢, 效率低, 安全性高

TCP协议下两个常用类全名

客户端:Socket

服务端:ServerSocket

TCP协议下字符串数据传输程序

你好服务器 收到谢谢

客户端Socket获取流发送和接收

服务端等待客户端连接, 通过客户端获取流,接收发送

TCP协议下文件上传案例

上传图片

客户端:

读取磁盘图片

通过网络流写出

通过网络流读取服务端的响应

服务端:

通过网络流读取上传的数据

写入磁盘保存图片

通过网络流发送响应

理解TCP写一下B/S案例

客户端: 浏览器

服务端:

通过网络流读取客户端发来的数据(处理)

读取磁盘的html文件

通过网络流发到客户端