Day 9

IO 流有两种分类：

1. 输入输出流

   输入: 从硬盘(文件)读取到内存(Java 程序)

   输出: 从内存(Java 程序)写入到硬盘(文件)

2. 字节流和字符流

   字节流 (读写字节: byte, 可以读写所有类型的文件, 包括视频, 图片, 音频, 文本等)

   字节输入流: java.io.InputStream 抽象类

   字节输出流: java.io.OutputStream 抽象类

   字符流 (读写字符: char, String, 只能读写文本文件)

   字符输入流: java.io.Reader 抽象类

   字符输出流: java.io.Writer 抽象类

计算机中一切都是字节存储的

java.io.OutputStream 抽象类: 字节输出流 (顶层类)

// 成员方法

void close() ：释放资源

void flush() ：刷新缓冲区(对于字节流来说没有作用)

// 写字节的成员方法

abstract void write(int b): 一次写一个字节 (参数 int 便于传递 byte 的整数不用强转)

void write(byte[] b): 一次写一个字节数组

void write(byte[] b, int offset, int len): 一次写一个字节数组的一部分

java.io.FileOutputStream 类: 文件字节输出流 (向文件写数据)

// 构造方法

FileOutputStream(String name): 通过文件路径创建文件字节输出流

FileOutputStream(File file): 通过 File 对象创建文件字节输出流

构造方法的作用:

1. 创建一个 FileOutputStream 对象

2. 根据构造方法传递的路径, 在磁盘上创建一个空文件("如果文件存在则会清空数据")

3. 将创建的 FileOutputStream 对象指向这个磁盘上的文件

文件存储原理和记事本打开文件的原理

向文件中写入字节数据时, 十进制的数字会被转换为"二进制"的数字写入文件

文本编辑器打开文本文件时, 会先查询编码表, 将二进制数字转换为对应的字符进行显示

字符输出流: 一次写入多字节

```
public static void main(String[] args) throws IOException {
    FileOutputStream f = new
FileOutputStream("jiuday09-code\\testOutputStream1.txt");
    f.write(97);
    f.write(98);
    f.write(99);
```

```
    f.close();
    FileOutputStream f1 = new
FileOutputStream("jiuday09-code\\testOutputStream1.txt",true)
;
    f1.write("你好".getBytes());
    f1.write("\r\n".getBytes());
    f1.write("你好".getBytes());
    f1.write("\r\n".getBytes());
    f1.write("你好".getBytes());
    f1.write("\r\n".getBytes());
    f1.close();
    FileInputStream f3 = new
FileInputStream("jiuday09-code\\testOutputStream1.txt");
    int s;
    while((s=f3.read())!=-1){
        System.out.print((char) s);
    }
    f3.close();
```

字节输出流的续写：在构造方法的参数中在加一个参数 true

换行符:

Windows 系统: "\r\n"

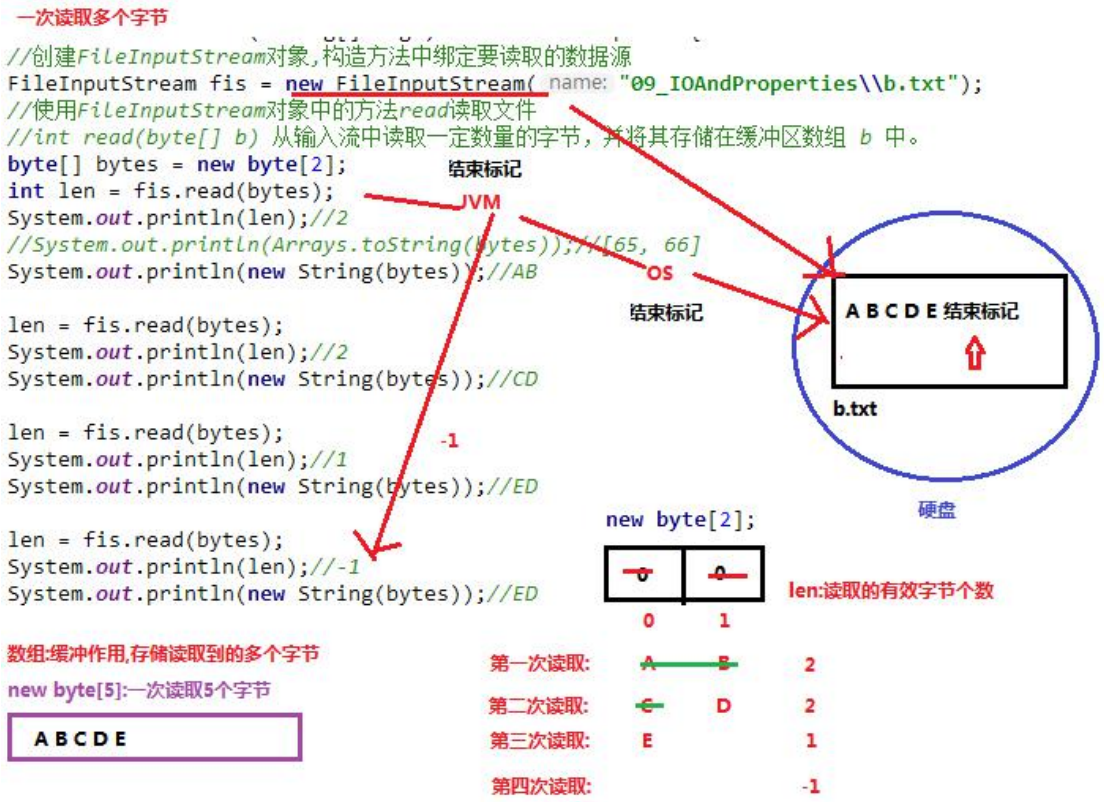
Linux 系统: "\n"

MacOS 系统: "\r"

Java 程序从文件读取数据的原理:

Java 程序 -> JVM 虚拟机 -> OS(操作系统) -> OS 调用读取的方法 -> 读取磁盘文件数据

字节流一次读取多个字节原理



如果不够取最后一次会有重复内容，所以要取有效的内容

复制图片案例：

因为电脑中所有都是由字节组成的，所以图片也是由字节组成的，读取和写入图片都必须用字节流。

```
public static void main(String[] args)  {
    FileInputStream f1 = null;
    FileOutputStream f2 = null;
    try{

        f1 = new FileInputStream("D:\\小岳岳.jpg");

        f2 = new FileOutputStream("e:\\小岳岳.jpg");

        byte[] bytes = new byte[1024];
        int s = 0;
        while((s=f1.read(bytes))!=-1){
            f2.write(bytes,0,s);
        }
    }catch (IOException e){
```

```
            System.out.println(e);
        }finally {
            if (f2 != null){
                try {
                    f2.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            if (f1 != null){
                try {
                    f1.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
}
```

字符流读取文件和字节流读取文件的原理是相同的，可以一次读取一个字符或一次读取一个字符数组两种方式。

字符输出流

注意: write()方法只是将数据写到内存缓冲区，最后必须调用 flush()或 close()才能将数据真正写入磁盘

flush 和 close 的区别:

flush(): 刷新缓冲区 (将数据从内存中写入到磁盘)

close(): 刷新缓冲区，并释放资源. 关闭流后不能再用同一个流对象操作

flush()可以省略，只用 close()来刷新并释放资源

write 的参数可以是字符串，字符，字符数组和写字符数组和字符串的一部分

字符输出流的续写换行和字节输出流的续写原理相同。

IO 异常处理

通用的方法是直接 try catch

```
FileInputStream fis = null;
try{
fis = new FileInputStream(s1);
}catch (IOException e){
    e.printStackTrace();
}finally {
    if (fis!=null){
        try {
            fis.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

JDK 7 中可以省略 finally，可自动释放资源

```
{
    try(FileWriter fw = new
FileWriter("jiuday09-code\\src\\prop.properties")){
        Properties p  = new Properties();

        p.setProperty("照例音","168");

        p.setProperty("迪丽热巴","165");

        p.setProperty("古力娜扎","160");

        p.store(fw,"save date");
    }catch (IOException e){
        System.out.println(e);
    }
}
```

JDK 9 中流对象的声明和创建可以放在括号外面，流对象要求是有效的 final 的，不修改流对象的值

Properties 双列集合:

键和值都是 String 类型

常用方法有：

Object setProperty(String key, String value): 保存/替换键值对

String getProperty(String key): 通过键获取值. 键不存在返回 null

Set<String> stringPropertyNames(): 返回键的集合

void store(OutputStream out, String comments): 将集合用字节流写入文件(不能中文),并写入注释

void store(Writer writer, String comments): 将集合用字符流写入文件(可以中文),并写入注释

void load(InputStream inStream): 从配置文件中通过字节流加载数据到 Properties 集合(不能读中文)

void load(Reader reader): 从配置文件中通过字符流加载数据到 Properties 集合(可以读中文)

总结：

在读文件时，会自动识别文件中的=和空格，用来分隔键和值，如果没有，就会把内容存到一个键中，值为空

今日错题：

ABC

下面代码执行之后

```
public static void main(String[] args) throws Exception {
    FileReader fr = new FileReader("测试文件.txt");
    int temp;
    while((temp=fr.read())!=-1){
        System.out.print(temp);
    }
    fr.close();
}
```

请问控制台显示结果是?( )

- A： ABC
- B： abc
- C： 656667
- D： 979899

我的答案： A

参考答案： C

因为输出的是 temp 的值，temp 只是存储读取文件的有效值，是 int 类型，所以会输出 ASCII 值，如果要输出 ABC 则要把 temp 强转成 char 类型

12、
在“测试文件.txt”中，有如下数据：

abc

请分析，以下代码执行后，文件中的数据为：（ ）

```
public static void main(String[] args) throws Exception {
    FileInputStream fis = new FileInputStream("测试文件.txt");
    FileOutputStream fos = new FileOutputStream("测试文件.txt");
    int temp;
    while((temp = fis.read()) != -1){
        fos.write(temp);
    }
    fis.close();
    fos.close();
}
```

A：abcabc
B：abc
C：没有数据
D：979899

我的答案：B

参考答案：C

在写文件时，会把文件中的内容全部覆盖，所以文件就是一个空文件了，从里面读不到任何内容，所以什么内容也不会被输出