输入输出IO

1. 输入输出概念：
2. 优点：使用输入输出流在外设上写入读取数据，外设上的数据断电不会丢失。
3. 概念：以内存为参照。数据从内存到外设称为输出；数据从外设到内存称为输入。
4. 输入输出体系结构：

五个类：File OutputStream InputStream Writer Reader

一个接口：Serializable

序列化：将对象的状态信息转换为可以存储或传输的形式的过程

1. File类

该类封装了文件和目录对象。对文件和目录对象提供了大量的操作。File类对象可以表示文件，也可以表示目录。在程序中一个File对象可以代表一个文件或目录。

数据成员：

separator：\ 如：D:\work\lesson

pathSeparator：; 如：D:\work\lesson；D:\work\lesson；D:\work\lesson

构造方法：

File(String pathname)

相对路径：不是从盘符开始的路径。相对于工程目录。

绝对路径：从盘符开始的路径。

方法成员：

static File[] listRoots()：判断是否插入或拔出U盘。

createNewFile()：创建普通文件。

mkdir()：创建文件夹。

delete()：删除文件或文件夹

exists()：判断是否存在对应的文件或目录。

getName()：返回文件名

getPath()：返回带路径的文件名

getAbsolutePath ()：返回绝对路径

isDirectory()：判断是否是目录

isFile()：判断是否是普通文件

length()：文件内容字节个数。一个回车对应两个字符：换行 10 回车 13

list()：必须是目录文件。返回文件夹下文件名。

list(FilenameFilter filter)

listFiles()：返回文件夹下文件对应的File对象。

listFiles(FilenameFilter filter)

练习：

判断是否插入一个U盘，插入则输出提示信息。

1. 流的分类

数据传输的方向：

输入流：InputStream、Reader

输出流：OutputStream、Writer

数据传输的单位：

字节流：InputStream、OutputStream

字符流：Reader、Writer

创建流对象传入的参数：

节点流：String文件名、File对象。 灰色

处理流：其他的流对象 白色

1. 节点流：

FileOutputStream：

FileOutputStream(String name)

FileOutputStream(String name, boolean append)

write(byte[] b)

close()

ByteArrayOutputStream

ByteArrayOutputStream()：构造方法中会自动创建缓冲区。

toByteArray()：获取流中的缓冲区。

write(byte[] b, int off, int len)：将b数组中的数据写到缓冲区。

FileInputStream：

FileInputStream(String name)

close()

int read(byte[] b)：返回值是实际读取到的字节数。

skip(long n)：设置文件指针的偏移量。

ByteArrayInputStream

ByteArrayInputStream(byte[] buf)

int read(byte[] b, int off, int len)

skip(long n)

FileWriter：

FileWriter(String fileName)

FileWriter(String fileName, boolean append)

write(String str, int off, int len)

void write(char[] cbuf, int off, int len) throws IOException

write(String str)

FileReader

FileReader(String fileName)

ready()

int read(char[] cbuf, int offset, int length)

int read(char[] cbuf)

文本文件是指以ASCII码方式(也称文本方式)存储的文件，更确切地说，英文、数字等字符存储的是ASCII码，而汉字存储的是机内码。文本文件中除了存储文件有效字符信息（包括能用ASCII码字符表示的回车、换行等信息）外，不能存储其他任何信息。

操作非文本文件时一定要用字节流。

操作文本文件时可以使用字符流和字节流，推荐使用字符流。

1. 处理流
2. IO编程步骤总结
3. PrintStream
4. 输入输出重定向

转移字符：\n \t \” \\ \’