视频看到210集。要写论文了。

BufferedReader:带有缓冲区的字符输入流。

BufferedWriter:带有缓冲区的字符输出流。

构造方法：BufferedReader(Reader r),这个是一个包装类，里面的的类是它的节点类，关闭流时只需要关闭外面的包装类即可。

BufferedReader中的方法：String readLine();可以读取一行数据，不带换行符。

BufferedWriter同理也是一个包装类，它有的方法newLine():写入一个换行符。

InputStreamReader(InputStream in)这是个转换流，可以把字节流转换成字符流。

题外话，System.in是一个标准的字节输入流。

装饰者模式：

1. 装饰者模式中要求：装饰者中含有被装饰者的引用。
2. 装饰者模式中要求：装饰者和被装饰者应该实现同一个类型。

DataOutputStream(OutputStream out): 数据字节输出流。输出的数据带类型。写入的是二进制文件，打开看不懂，要读此类文件要用到DataInputStream(InputStream in)。注意：要使用该流读取数据，必须提前知道该文件中数据的存储格式，顺序。读的顺序必须和写入的数据相同。

PrintStream；标准的输出流，默认打印到控制台，以字节方式

System.out返回的就是一个PrintStream

改变输出方向PrintStream(OutputStream out)

System.setOut(PrintStream p)

PrintWriter：以字符的方式

ObjectInputStream:将硬盘中的数据反序列化到jvm内存。（deserial）

Object readObject()

ObjectOutputStream：序列化java对象到硬盘。（serial）

创建序列化流：new ObjectOutputStream(OutputStream out)

写：writeObject(Serializable s);（ps：Serializable是一个没有任何方法的接口，是一个标识接口，像这样的接口还有Cloneable）

标识接口的作用，起到标识作用，jvm如果看到该对象实现了某个标识接口，会对它特殊待遇。Serializable的实现类，jvm会给类添加一个属性：static final long serialVersionUID。也可以自己定义这个版本号。如果不想让属性参加序列化，需要使用transient关键字修饰。

File:

1. File类和流无关，不能通过该类完成文件的读和写。
2. File是文件和目录路径名的抽象表示形式。

File代表的是硬盘上的Directory和file

boolean exists()；判断此文件或者路径名是否存在。

mkdir();创建目录

mkdirs();创建多重目录

createNewFile();创建文件

String getAbsolutePath();获取绝对路径

String getName();获取文件名

String getParent();获取父路径

long lastModified();文件最后修改时间

length();获取文件的长度（字节数）

File[] listFiles();获取文件下的子文件

多线程：线程和线程共享“堆内存和方法区内存”，栈内存是独立的，一个内存一个栈。

java程序的执行原理：

java命令会启动java虚拟机，启动jvm等于启动了一个应用程序，表示启动了一个进程。该进程会自动启动一个“主线程”，然后主线程去调用某个类的main方法，所以main方法运行在主线程中，在此之前的所有程序都是单线程的。

在java语言中实现多线程的第一种方式：

第一步：继承java.lang.Thread;

第二步：重写run方法；

启动线程start();

有了多线程之后，main方法结束只是主线程栈中没有方法栈帧了。

但是其它线程或者其它栈中还有栈帧。

main方法结束，程序可能还在运行。

java中实现线程的第二种方式：

第一步：写一个类实现java.lang.Runnable;接口

第二步：实现run方法。

创建线程：Thread t = new Thread(Runnable r);

线程的声明周期：

