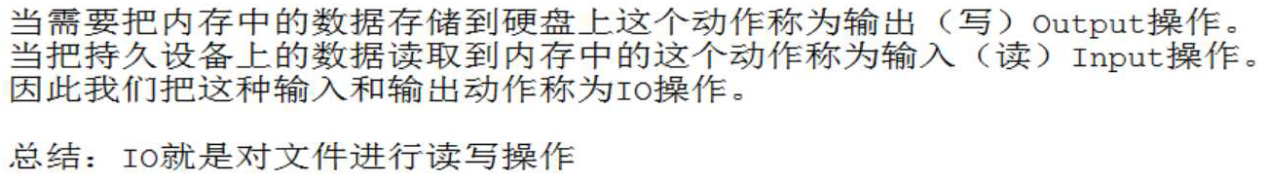
1 **delete()可以删除文件和没有文件的文件夹。**Exists()也可以用于文件和文件夹。

2 输入输出都是相对于内存的。

3 汉诺塔最底下一个盘子为什么没在程序中操作？？？？？？？？？？？？？？

4

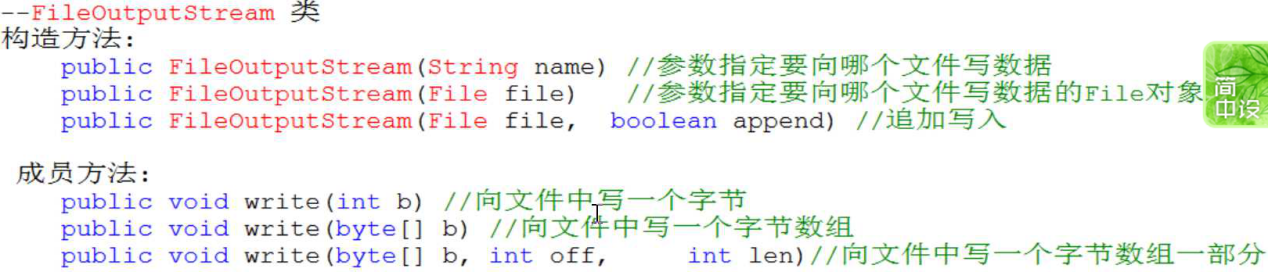


5 （**字节流和字符流都有缓冲流**）

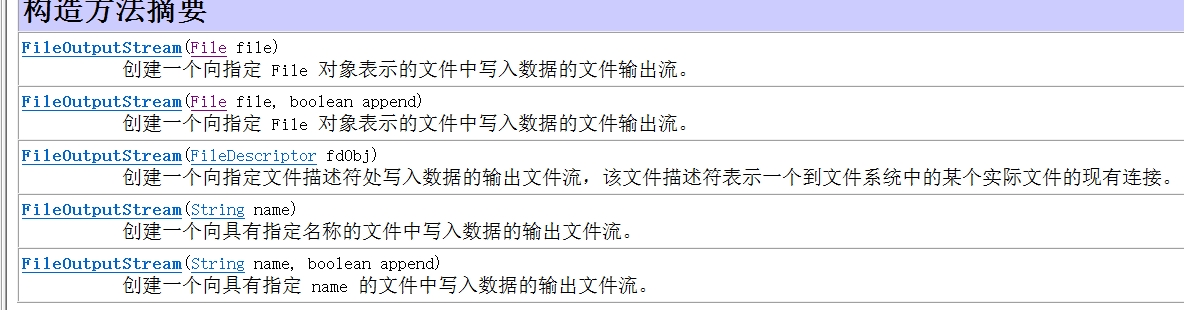


6 字符流只能操作**文本文件**。**字节流可以操作任何类型的文件**。计算机底层只有字节流。

7



**四种构造方法：**



8 创建输入输出流会抛出异常

9



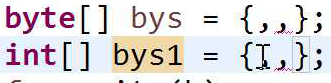
10 当获取的文件是文件对象而不是文件名时：



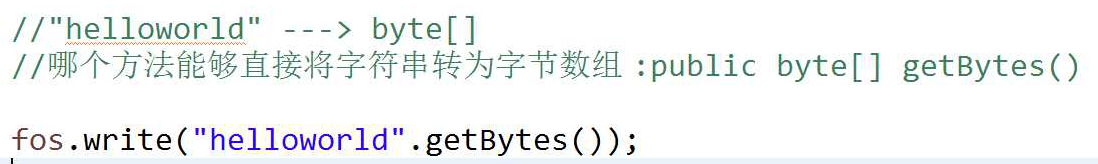
11 所有的字节流都不需要刷新（缓冲流需要刷新），字符流才需要刷新。字节流写入97就是写入a。

**因为记事本显示的不是文件本身，而是文件查ascii码表后的结果。**记事本不是每种ascii码都能显示出来，也不是每个数字都对应ascii码表。**符号显示不出来就显示问号。例如记事本不能显示笑脸。如果查不到的话就不会显示问号，而是显示乱码。**

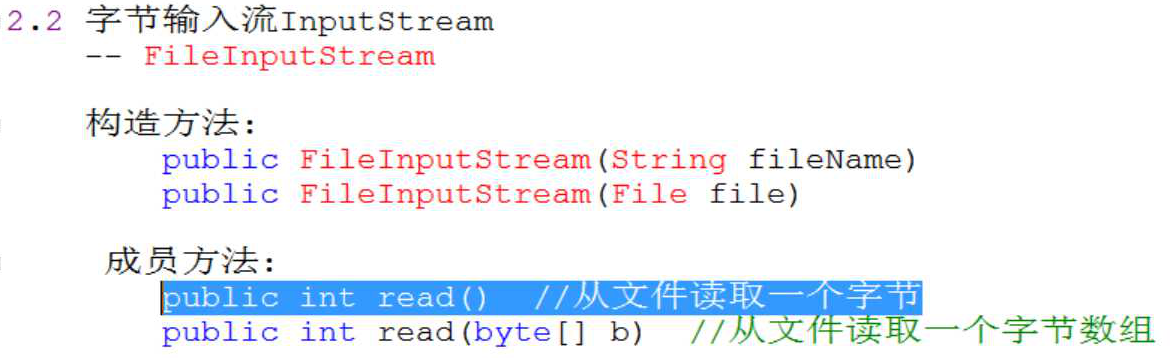
12 字节数组byte和int数组的区别是前者是-128~127.



13 （区分转化为字符数组和转化为字节数组（一个字符有两个字节），char[],byte[]，**getBytes(),toCharArray()**）



14

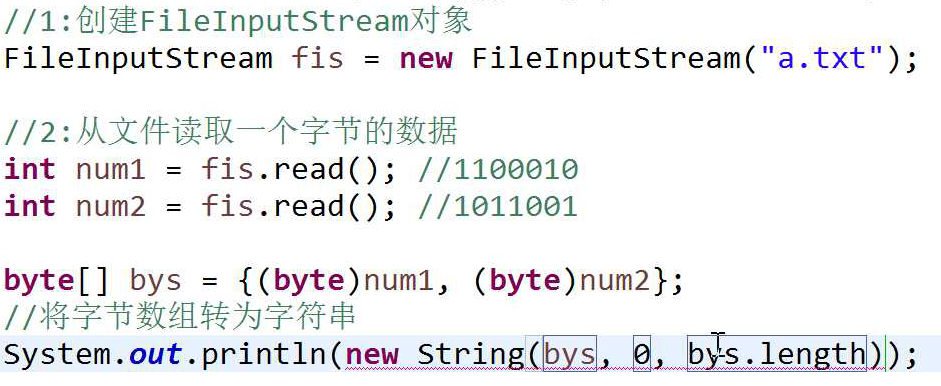


顺便将字节转换成字符：(char)强转



15 **在内存中所有字符都是两个字节，而文件中中文字符两个字节，英文字符一个字节,所以上面可以读到英文字符**。

16 **将字节数组转换成字符串：（记住字节数组和字符串的相互转换）**

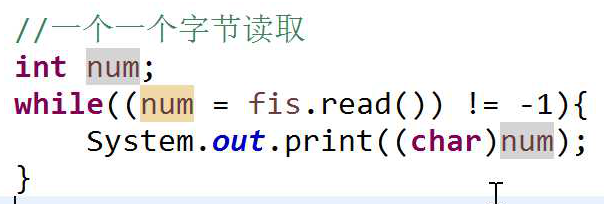


17 一行的最大长度是1024个字节，因此字符数组一般设置为1024.

18 读取字节数组（**读取数组相当于一起装到车里，缓冲流相当于装满车才能走。只有数组比只有缓冲流速度快得多。数组大于缓冲流的内置数量的话就没什么效果了，因为车的大小是固定的。因此数组不用太大。**）



19 **下面的方法不能读中文：**



**当操作纯文本时，不要用字节流，要用字符流。**

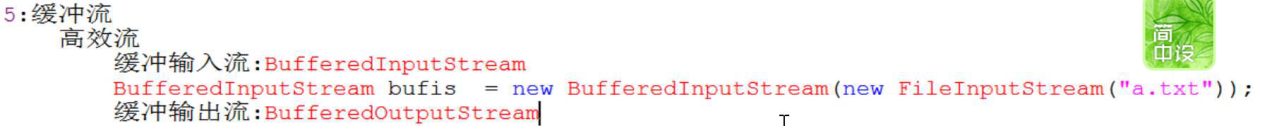
如果要用字节流（**一个字节一个字节的**）操作文本文件：不需要刷新，但仍然要关闭流。



20 10.20-10.50重新看？？？？？？？？？？？

21 字节复制和字节数组复制速度相差巨大，**原因就是循环次数的问题**。

22 缓冲流即高效流。



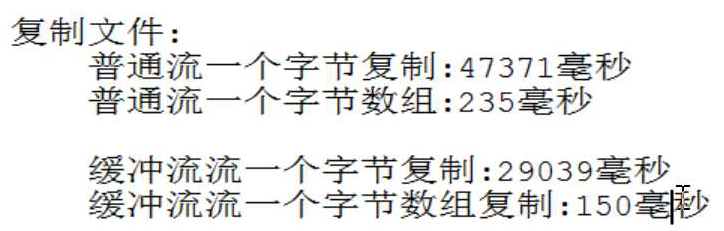
缓冲流是装饰模式，是功能增强类。

23 **字节缓冲流中没有一个特有方法，都是继承的。字符缓冲流有两个特有方法readLine()和newLine()。前者是输入流后者是输出流。**

24 带缓冲的字节流必须刷新，不带缓冲的字节流不用刷新。当然刷新也不算错。（**不刷新也没事，只不过数据只有在close()之后才能显示**）

25 所有的字符流都需要刷新。

26 速度比较举例：



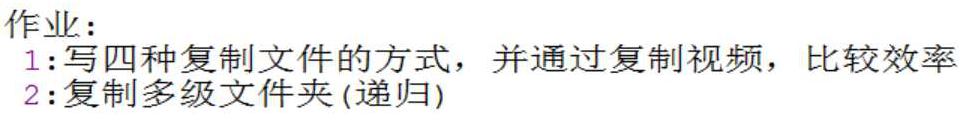
一个字节的和一个字节的表示，字节数组的和字节数组的表示。

27 复制文件夹案例：（**注意新文件绝对路径的拼接**）

getName()：此抽象路径名表示的**文件或目录**的名称；如果路径名的名称序列为空，则返回**空字符串**



28 作业：



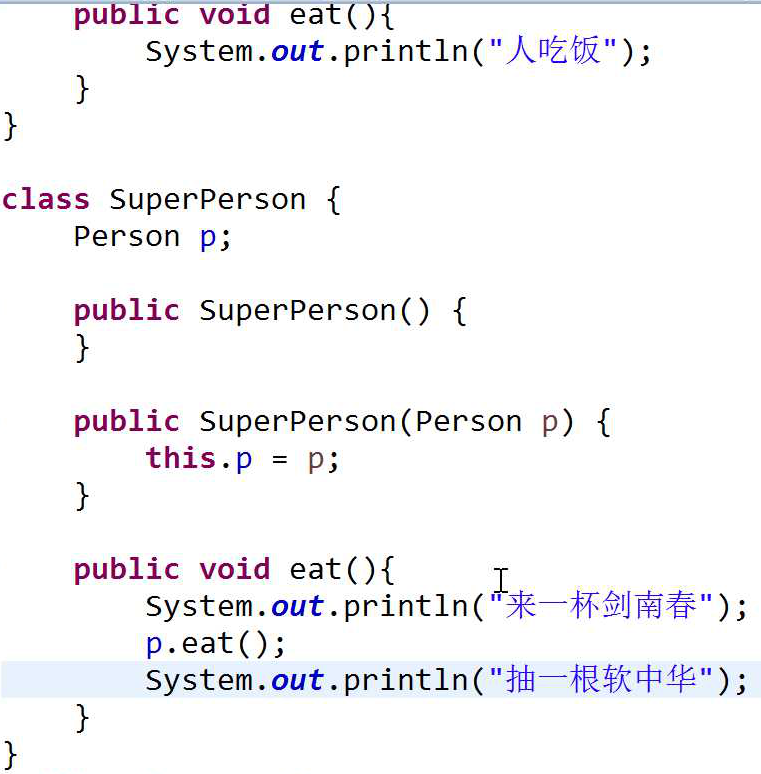
29 字节流写的是ascii码值。

30 字符流的底层是字节流。

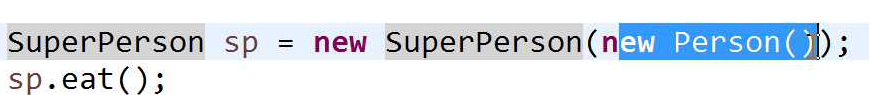
31 **listFiles()方法应用于a.txt的话数组为null。**

32 ListFiles()返回所有的子文件和文件夹。不只是文件夹或文件。

33 装饰模式：将要增强的对象当做成员变量传入新的对象：



使用方法：



不够好，可以改装成传入抽象类做参数：

**看第十七个视频？？？？？？？？？？？？？？？？**

**Pack05 Testdemo.java**

34 不带数组的缓冲流和不带数组的字节流相比是否效果明显？？？？？？？？？？？

35 File fii=**new** File(duqu);

File[] listFiles = fii.listFiles();

**for** (File file : listFiles) {

**if**(file.isFile()){//是文件的话

System.***out***.println(file.getPath());

**getPath()返回的路径是绝对路径，因为duqu传入的就是绝对路径。**

36 readLine()方法判断是否是等于null，而不是是否是等于-1.

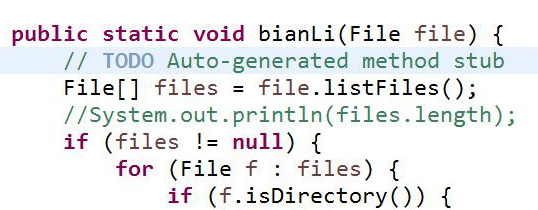
37 不能在同一目录下创建同名文件和文件夹。应该指的是没有后缀的文件。或者是下面这样也不行：

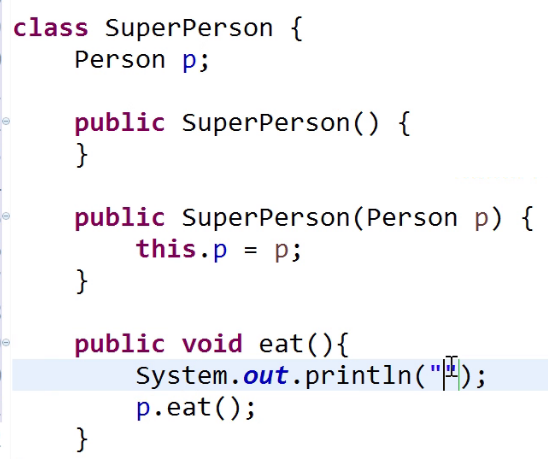


但是linux系统可以。

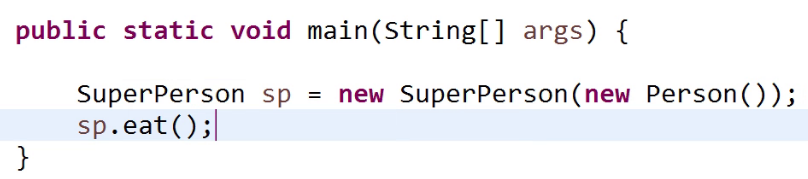
38 在盘符下面遍历时用listFiles().length可能会报错，因为有隐藏文件，而权限不够无法访问，所以会报错。其他的文件夹即使是空文件夹listFiles().length也不会返回null,而是报错。

解决办法：加上！=null：（也仅仅是不报错而已。）



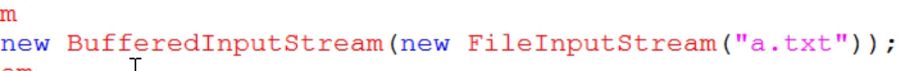
39 装饰设计模式：将被装饰的类作为新的装饰器类的一个成员变量：  


..........................................................................................................................................

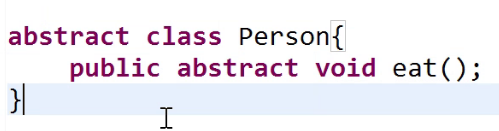


但是不够好。改造：

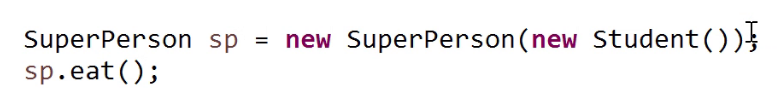
应该改成传入抽象类作为参数，以使某个抽象类的子类都可以传进去做参数：



使得是InputStream的子类都可以传入。抽象类如下：

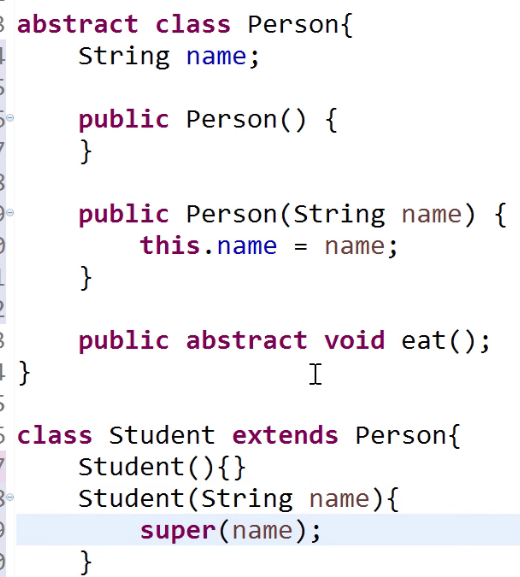


然后将抽象类的子类传入做参数：



总之，装饰设计模式一个是将原来的类作为新类的成员变量，另一个是将原来的类改成原来的类的抽象父类，使得该抽象父类的子类都可以传入新类做参数。（**即一个装饰器可以装饰多个类**）

40 抽象类虽然不可以创建对象，但可以有构造函数：



41 模仿BufferedWriter()给装饰器中作为参数的对象传入参数：

