**Java学习day20**

**1.IO流， File概述 对应1,2,3,4**

用来将文件或者文件夹封装成对象，方便对文件与文件夹的属性信息进行操作。

并且file对象可以作为参数传递给流的构造函数。

separator 是一个跨平台的分隔符。

1.创建。

boolean createNewFile(): 在指定位置创建文件，如果该文件已经存在，则不创建，返回false。

和输出流不一样，输出对象一键立创建文件，而且文件已经存在，会覆盖。

2.删除。

boolean delete():删除失败返回false

void deleteOnExit():在程序退出的时候删除指定的文件

3.判断

学习了几个方法

boolean exists() 文件是否存在

isHidden() 隐藏文件是否存在

isDirectory()

isFile

isAbsolute() 是否是绝对路径。

makdir() 记住这个方法只能创建一级文件。

4.获取信息。

getName():

getPath():

getparent():

getAbsolutePath():

lastModified():

length():

renameTo:重命名文件的抽象路径名记。 其实现的功能就像是剪切一样

**2.IO流（File对象功能-文件列表） 对应day5,6**

了解一下File类中的list和listRoots这两个方法

static File[] listRoots()

列出可用的文件系统根

String[] list()

返回的字符串在该目录下的抽象路径名的文件和目录命名为数组。

第一个方法能够列出能够有管理权限的磁盘，第二个方法能够列出c盘中具体都有哪些文件

list()

返回的字符串在该目录下的抽象路径名的文件和目录命名为数组。

留一个练习：过滤掉抽象路径中满足过滤器的文件和目录，用到list(FilenameFilter Filter)，比如都留下了.java这些文件

练习的实现：用匿名内部类中的方法重写来实现对应的功能，要用到endsWith这个方法来实现。并且查阅accept这一个api函数

然后改进一下，用listFiles比list好，因为listFiles中已经有很多方法了。因为拿到的是File这个列表对象， 拿到对象以后可以多次操作。

**3.// IO流（列出目录下所有内容） 递归 对应07day**

/\*

这里我们用递归的方法来打印出目录中还有的目录。

递归就是调用函数自身。

1.递归要注意限定条件。

2.递归要注意递归的次数，避免栈溢出。

两个递归的练习：

练习1：1.递归运算：10进制转换成二进制。

练习2：数的1+到n的相加

1. **列出目录下所有内容 带格式 对应day8**
2. **删除原理： 在windows中，删除目录从里面往外删除， 对应day9**

既然是从里往外删除，就需要用到递归操作。

**6.练习：**

将一个指定目录下的java文件的绝对路径，存储到一个文本文件中。

建立一个java文件列表文件。

思路：

1.对指定的目录进行递归；

2.获取递归过程中的java文件的路径；

3.将这些路径存储到集合中，

4.将这些数据写到一个文件中

这个练习可以当做是一个有意思的归纳的东西。

**7.IO流（Properties存取） 对应11和12 13day**

Class Properties

该对象的特点：可以用于键值对形式的配置文件。

那么在加载数据时，需要数据有固定格式：键=值

getProperty(String key)

在这个属性列表中搜索指定的键的属性。

setProperty(String key, String value)

调用方法 put Hashtable。

stringPropertyNames()

在这个属性列表中返回一组键，其中键和它的对应值是字符串，包括在默认属性列表中的不同键，如果同一个名称的一个键没有从主要属性列表中找到。

演示：如何将流中的数据存储到集合中。

想要将info.txt中的简直数据存到集合中进行操作。

1.用一个流与info.txt文件关联。

2. 读取一行数据，将该行数据用"="进行切割。

3.等号左边作为键，右边作为值。存入到Properties集合中即可。

**8.IO流（Properties练习）**

思路：程序解读：

新建一个Properties类

为什么一定要用流来操作文件？因为放入内存有可能会消失（程序退出），而放入流文件则可以保存数据。

新建文件，如果文件不存在，则创建一个文件，并且文件的后缀名为.ini这种配置文件的格式

用文件输入流来操作文件， 并且将流中的数据加载进集合。

定义常数count，并且获取集合中的键time，getProperty这个方法能够返回一个值，如果值为null则+1，如果不为null则用Integer解析为整形数字。

将存完的数据重新写回去用setProperty这一个方法，并且定义输出流文件， 用store来将将内存结果存储到流中

记得关流。

新增的东西：就是count到了一定的次数会退出。

**9.打印流 IO流（printWriter） 对应15**

PrintWriter是字符流。 是OutputStream中的一个子类

其中最强的功能：println可以用这个方法直接将数据类型原样打印出去

构造函数可以接受的参数类型.

1. file对象。File

2. 字符串路径。String

3. 字节输出流。OutputStream

4. 字符输出流。 Writer

编程注释：

用字节流的形式读入文件，并且用缓冲区的方式提高效率。

定义一个输出流，用的是OutputStream的子类，PrintWriter

然后用readLine()方法操作读入文件，用println(可以查一下API它的优势)来操作输出文件输入

记得关流和抛异常。

**10.IO流（合并流）**

编程思路：创建一个Vector对象， 然后添加文件， 用Enumeration枚举将文件放入

定义一个SequenceInputStream 合并流，放入枚举的文件。 定义 文件输出流，

创建字节数组， 并且按照字节数组的格式读入，其中读入的东西是ASCII码，并且读入-1即为最后一行。

.write方法写入输出流文件，（字节数组名，0，对应的ASCII码）

关流。

**11.IO流（切割文件）**

编程思路：

1.定义切割函数：

文件输入流定义， 用懒汉式定义文件输出流， 定义一个byte类型的数组

定义一个文件读取的ASCII码的存储变量， 定义一个文件名的变量，

开始读取文件，懒汉式完成，并在其中写入文件。要在函数内部立即关流，一次切割完成 存储的名称为part

关流，切割完成

2.定义合成函数：

定义一个ArrayList类型的列表，记住泛型

for循环添加切割后的part进行添加。

创建一个迭代器，并且定义一个匿名内部类搞枚举的东西，这里看不懂可以暂时放一下，

获取枚举的元之后，放入SequenceInputStream合并流中，

定义输出流，定义byte类型元组，定义常量ASCII，然后进行read写入操作。

两个都关流。