**io小结**

Java的io非常非常重要！它是每个程序的必须组成部分，也是和外界交流的主要功能。 在电脑程序来说，文件是所有的基础，Java中，使用File类进行文件和文件夹的访问和修改。

Java的io使用“流”的概念，一般来说

按照数据的方向，分为输入和输出流，以用户内存（运行时程序所在的内存）为着眼点，向内存输入的是输入流，由内存输出的是输出流。

按照流的角色，分为节点流和处理流。直接向io设备对接的是节点流，对节点流进行包装的是处理流，也称为高级流。

按照操作的数据单元，分为字节流和字符流。字节流是万能的，字符流处理文本对象。

以下是一个读取文件的示例代码：

**package** ioDemo;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

/\*\*

\* 使用io将文件中的内容显示。

\* **@author** Administrator

\*

\*/

**public** **class** **Demo1** {

**public** **static** **void** **main**(**String**[] args) **throws** **IOException** {

//对于程序内存来说，文件中的内容向程序内存输入，所以使用输入流

**FileInputStream** **fis** = **new** FileInputStream("i:/003.txt");

/\*

\* 显示的过程类似于蚂蚁搬家，每只蚂蚁可以承载的重量是buff这个字节数组

\* 使用hasRead标记是否已经存入内存

\* 然后循环输出

\* 最后关闭输入流，节省资源

\*/

**byte**[] **buff** = **new** **byte**[1024];

**int** **hasRead** = 0;

**while**((hasRead=fis.read(buff))>0)

{

**System**.***out***.println(**new** String(buff,0,hasRead));

}

fis.close();

}

}

一般来说，使用处理流比较方便，因为处理流已经屏蔽了各个不同的节点之间的差异，可以更加方便，而且处理流的效率更高。

处理流的语法是（以PrintStream为例）PrintStream(节点流对象)。