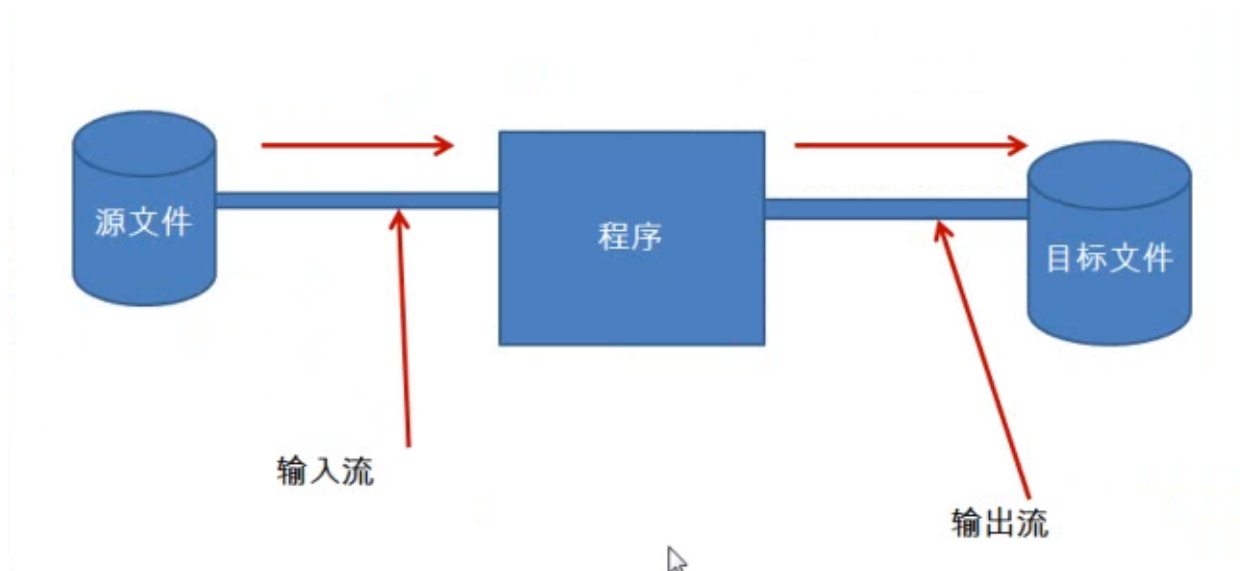


- 1, io流用来处理设备之间的数据传输
- 2, java程序中, 对于数据的输入/输出操作以“流(stream)”的方式进行
- 3, java.io包下提供了各种“流”类和接口, 用以获取不同种类的数据, 并通过标准的方法输入或输出数据

据



输入input: 读取外部数据(磁盘、光盘等存储设备的数据)到程序(内存)

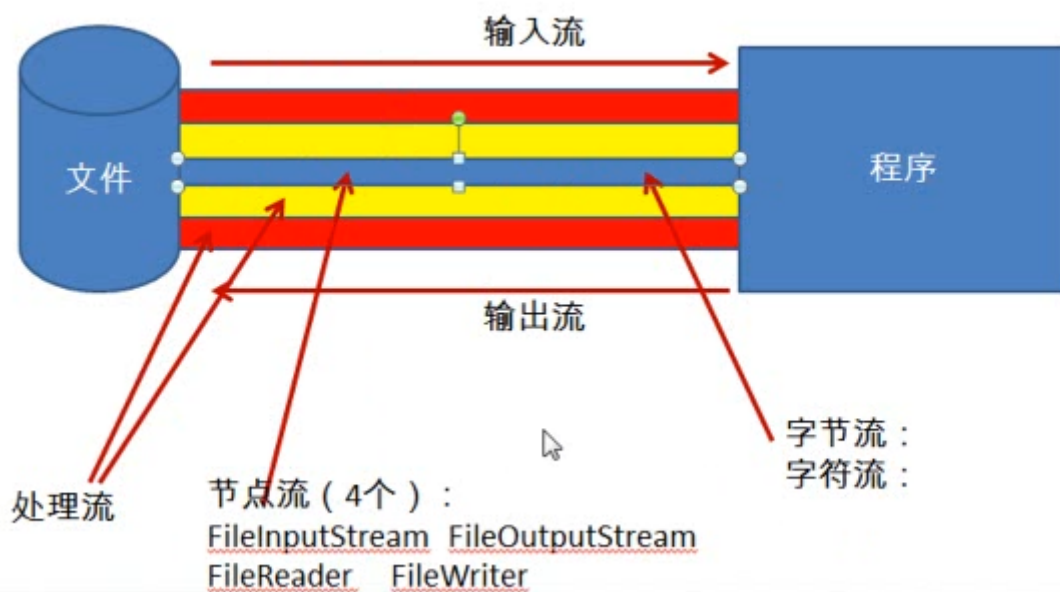
输出output: 将程序(内存)数据输出到磁盘、光盘等存储设备中

一, 分类

- 1, 按照操作数据单位不同分为: 字节流(8bit)和字符流(16bit)
- 2, 按照数据流的流向不同分为: 输入流和输出流
- 3, 按照流的角色不同分为: 字节流(直接作用于文件)和处理流

1, java的io流共涉及40多个类, 实际上非常规则, 都是从如下4个抽象基类派生的

2, 由这四个类派生出来的子类名称都是以其父类作为子类名后缀



分类	字节输入流	字节输出流	字符输入流	字符输出流
抽象基类	InputStream	OutputStream	Reader	Writer
访问文件	FileInputStream	FileOutputStream	FileReader	FileWriter
访问数组	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
访问管道	PipedInputStream	PipedOutputStream	PipedReader	PipedWriter
访问字符串			StringReader	StringWriter
缓冲流	BufferedInputStream	BufferedOutputStream	BufferedReader	BufferedWriter
转换流			InputStreamReader	OutputStreamWriter
对象流	ObjectInputStream	ObjectOutputStream		
	FilterInputStream	FilterOutputStream	FilterReader	FilterWriter
打印流		PrintStream		PrintWriter
推回输入流	PushbackInputStream		PushbackReader	
特殊流	DataInputStream	DataOutputStream		

1, IO的体系

抽象基类

InputStream

OutputStream

节点流 (文件流)

FileInputStream

FileOutputStream

Reader

FileReader

Writer

FileWriter

①（FileInputStream操作文件）使用try--catch--finally的方式进行处理更加合理，保证流的关闭操作一定可以被执行

```
//从硬盘存在的一个文件中读取内容到程序, 使用FileInputStream
//要读取的文件一定要存在，否则抛FileNotFoundException异常
FileInputStream fis=null;
int s=0;
//创建一个file类的对象
File file = new File("e://hello.doc");
try {
    //创建一个fileinputstream类的对象
    fis = new FileInputStream(file);
    int b;
    //调用fileinputstream的方法，实现file文件的读取
    while((b=fis.read())!=-1) {
        System.out.println(b);
        s++;
        if (s==30) {
            System.out.println();
            s=0;
        }
    }
} catch (Exception e) {
    // TODO: handle exception
}
finally {
    try {
        //关闭相应的文件流
        fis.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
    }
}
```

```

        e.printStackTrace();
    }
}

```

另一种形式:

```

File file = new File("e://hello.doc");
FileInputStream fis=null;
int s=0;
try {
    fis = new FileInputStream(file);
    byte[] b = new byte[1024]; //读取到的数据要写入到数组
    int len; //每次读到数组中的字节的长度
    while ((len = fis.read(b)) != -1) {
        for (int i = 0; i < len; i++) {
            System.out.print(b[i]);
            s++;
            if (s==30) {
                System.out.println();
                s=0;
            }
        }
    }
} catch (Exception e) {
    // TODO: handle exception
    e.printStackTrace();
}
finally {
    if (fis!=null) {
        try {
            fis.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

