

非贷款，0元入学，不1万就业不给1分钱学费，我们已千四年了！

笔记总链接：<http://bbs.itheima.com/thread-200600-1-1.html>

8、IO

8.2 IO流

8.2.4 IO流常用基类-字节流

读取一个键盘录入的数据，并打印在控制台上。

键盘本身就是一个标准的输入设备，对于java而言，对于这种输入设备都有对应的对象。

示例3：

```
01. import java.io.IOException;
02. import java.io.InputStream;
03.
04. public class ReadKey{
05.     public static void main(String[] args) throws IOException {
06.         readKey();
07.     }
08.
09.     public static void readKey() throws IOException {
10.         InputStream in = System.in;
11.
12.         int ch = in.read(); //阻塞式方法
13.         System.out.println(ch);
14.
15.         ch = in.read(); //阻塞式方法
16.         System.out.println(ch);
17.
18.         ch = in.read(); //阻塞式方法
19.         System.out.println(ch);
20.
21.         in.close();
22.     }
23. }
```

复制代码

运行结果：

```
D:\code\day21>javac ReadKey.java
D:\code\day21>java ReadKey
abc
99
99
99
```

P.S.

1. 获取键盘录入数据，然后将数据流向显示器，那么显示器就是目的地。

通过System类的setIn，setOut方法可以对默认设备进行改变。

System.setIn(new FileInputStream("1.txt")); //将源改成文件1.txt.

System.setOut(new PrintStream("2.txt")); //将目的改成文件2.txt

因为字节流处理的是文本数据，可以转换成字符流，操作更方便。

BufferedReader bufr =
new BufferedReader(new InputStreamReader(System.in));

BufferedWriter bufw =
new BufferedWriter(new OutputStreamWriter(System.out));

2. 默认的输入和输出系统不需要关，它会随着系统的结束而消失。

练习：

获取用户键盘录入的数据并将数据变成大写显示在控制台上，如果用户输入的是over，结束键盘录

入。

思路：

1. 因为键盘录入只读取一个字节，要判断是否是over，需要将读取到的字节拼成字符串。

2. 那就需要一个容器：StringBuilder。

3. 在用户回车之前将录入的数据变成字符串判断即可。

代码：

```
01. import java.io.IOException;
02. import java.io.InputStream;
03.
04. public class ReadKey{
05.     public static void main(String[] args) throws IOException {
06.         readKey();
07.     }
08.
09.     public static void readKey() throws IOException {
10.
11.         //1. 创建容器
12.         StringBuilder sb = new StringBuilder();
13.
14.         //2. 获取键盘读取数据
15.         InputStream in = System.in;
16.
17.         //3. 定义变量记录读取到的字节，并循环获取
18.         int ch = 0;
19.
20.         while((ch = in.read()) != -1){
21.             //在存储之前需要判断是否是换行标记，因为换行标记不存储
22.             if(ch == '\n' )
23.                 continue;
24.             if(ch == '\n' ){
25.                 String temp = sb.toString();
26.                 if("over" .equals(temp))
27.                     break;
28.                 System.out.println(temp.toUpperCase());
29.                 sb.delete(0,sb.length());
30.             } else{
31.                 //将读取到的字节存储到StringBuilder中
32.                 sb.append(( char)ch);
33.             }
34.         }
35.     }
36. }
```

复制代码

运行结果：

```
D:\code\day21>javac ReadKey.java
D:\code\day21>java ReadKey
abcde
ABCD
123456
123456
over
```

8.2.5 转换流

转换流的由来：

字符流与字节流之间的桥梁

方便了字符流与字节流之间的操作

转换流的应用：

字节流中的数据都是字节时，转换成字符流操作更高效。

转换流：

InputStreamReader：字节到字符的桥梁，解码。

OutputStreamWriter：字符到字节的桥梁，编码。

InputStreamReader是字节流通向字符流的桥梁。

示例1：

```
01. import java.io.BufferedReader;
02. import java.io.IOException;
03. import java.io.InputStream;
04. import java.io.InputStreamReader;
05.
06. public class TransStreamDemo{
07.
08.     public static void main(String[] args) throws IOException {
09.         //字节流
10.         InputStream in = System.in;
11.
12.         //将字节转换成字符的桥梁，转换流
13.         InputStreamReader isr = new InputStreamReader(in);
14.
15.         //对字节流进行高效装饰，缓冲区
16.         BufferedReader bufr = new BufferedReader(isr);
17.
18.         String line = null;
19.
20.         //读取到了字符串数据
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.             System.out.println(line.toUpperCase());
25.         }
26.     }
27. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

P.S.

使用字节流读取一个中文字符需要读取两次，因为一个中文字符由两个字节组成，而使用字符流只需读取一次。

System.out的类型是PrintStream，属于OutputStream类别。

OutputStreamReader是字节流通向字符流的桥梁。

示例2：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.IOException;
04. import java.io.InputStream;
05. import java.io.InputStreamReader;
06. import java.io.OutputStream;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         InputStream in = System.in;
14.
15.         InputStreamReader isr = new InputStreamReader(in);
16.
17.         //字符流
18.         BufferedReader bufr = new BufferedReader(isr);
19.
20.         //字节流，字节数据
21.         OutputStream out = System.out;
22.
23.         //数据到了osw，目的地控制台
24.         OutputStreamWriter osw = new OutputStreamWriter(out);
25.
26.         BufferedWriter bufw = new BufferedWriter(osw);
27.
28.         String line = null;
29.
30.         while((line = bufr.readLine()) != null){
31.             if("over" .equals(line))
32.                 break;
33.
34.             //将字符数据用缓冲区对象将数据写入缓冲区，目的是osw
35.             bufw.write(line.toUpperCase());
36.             bufw.newLine();
37.             //osw.write(line.toUpperCase()) + "\n\n"可以替代上面两行代码
38.             bufw.flush();
39.         }
40.     }
41. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

以上代码可以简化为：

示例3：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.IOException;
04. import java.io.InputStream;
05. import java.io.InputStreamReader;
06. import java.io.OutputStreamWriter;
07.
08. public class TransStreamDemo{
09.
10.     public static void main(String[] args) throws IOException {
11.
12.         BufferedReader bufr = new BufferedReader(new InputStreamReader(System.in));
13.
14.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
15.             FileOutputStream("a.txt")));
16.
17.         String line = null;
18.
19.         while((line = bufr.readLine()) != null){
20.             if("over" .equals(line))
21.                 break;
22.
23.             bufw.write(line.toUpperCase());
24.             bufw.newLine();
25.             bufw.flush();
26.         }
27.     }
28. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将键盘录入的数据写入到一个文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.InputStreamReader;
05. import java.io.OutputStreamWriter;
06. import java.io.IOException;
07.
08. public class TransStreamDemo{
09.
10.     public static void main(String[] args) throws IOException {
11.
12.         BufferedReader bufr = new BufferedReader(new InputStreamReader(System.in));
13.
14.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
15.             FileOutputStream("a.txt")));
16.
17.         String line = null;
18.
19.         while((line = bufr.readLine()) != null){
20.             if("over" .equals(line))
21.                 break;
22.
23.             bufw.write(line.toUpperCase());
24.             bufw.newLine();
25.             bufw.flush();
26.         }
27.     }
28. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将一个文本文件中的内容复制到另一个文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04. import java.io.FileOutputStream;
05. import java.io.IOException;
06. import java.io.InputStreamReader;
07. import java.io.OutputStreamWriter;
08.
09. public class TransStreamDemo{
10.
11.     public static void main(String[] args) throws IOException {
12.
13.         BufferedReader bufr = new BufferedReader(new InputStreamReader(new
14.             FileInputStream("a.txt")));
15.
16.         BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
17.             FileOutputStream("b.txt")));
18.
19.         String line = null;
20.
21.         while((line = bufr.readLine()) != null){
22.             if("over" .equals(line))
23.                 break;
24.
25.             bufw.write(line.toUpperCase());
26.             bufw.newLine();
27.             bufw.flush();
28.         }
29.     }
30. }
```

复制代码

运行结果：

```
D:\code\day21>javac TransStreamDemo.java
D:\code\day21>java TransStreamDemo
abcde
ABCD
123456
123456
over
```

需求：将a.txt文件中的内容复制到b.txt文件中。

代码：

```
01. import java.io.BufferedReader;
02. import java.io.BufferedWriter;
03. import java.io.FileInputStream;
04.
```


是，就加上buffer

需求1：复制一个文本文件

1、明确源和目的。

源：InputStream Reader

目的：OutputStream Writer

2、是否是纯文本？

是！

源：Reader

目的：Writer

3、明确具体设备。

源：

硬盘：File

目的：

硬盘：File

```
FileReader fr = new FileReader("a.txt");
FileWriter fw = new FileWriter("b.txt");
```

4、需要额外功能吗？

需要，高效

```
BufferedReader bufr = new BufferedReader(new FileReader("a.txt"));
BufferedWriter bufw = new BufferedWriter(new FileWriter("b.txt"));
=====
```

需求2：读取键盘录入信息，并写入到一个文件中

1、明确源和目的。

源：InputStream Reader

目的：OutputStream Writer

2、是否是纯文本？

是！

源：Reader

目的：Writer

3、明确具体设备。

源：

键盘：System.in

目的：

硬盘：File

```
FileReader fr = new FileReader("a.txt");
FileWriter fw = new FileWriter("b.txt");
```

这样就可以完成，但是麻烦。将读取的字节数据转换成字符串，再由字节流操作。

4、需要额外功能吗？

需要，转换。将字节流转换成字符串，因为明确的源是Reader，这样操作文本数据更快捷。

```
InputStreamReader isr = new InputStreamReader(System.in);
FileWriter fw = new FileWriter("b.txt");
```

还需要功能吗？

需要，高效。

```
BufferedReader bufr = new BufferedReader(new InputStreamReader(System.in));
BufferedWriter bufw = new BufferedWriter(new FileWriter("b.txt"));
=====
```

需求3：将一个文本文件数据 displayed 在控制台上

1、明确源和目的。

源：InputStream Reader

目的：OutputStream Writer

2、是否是纯文本？

是！

源：Reader

目的：Writer

3、明确具体设备。

源：

硬盘：File

目的：

硬盘：System.out

```
FileReader fr = new FileReader("a.txt");
FileWriter fw = System.out//PrintStream
```

4、需要额外功能吗？

需要，转换。

```
FileReader fr = new FileReader("a.txt");
OutputStreamWriter osw = new OutputStreamWriter(System.out);
```

需要，高效。

```
BufferedReader bufr = new BufferedReader(new FileReader("a.txt"));
BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(System.out));
=====
```

需求4：读取键盘录入数据，显示在控制台上

1、明确源和目的。

源：InputStream Reader

目的：OutputStream Writer

2、是否是纯文本？

是！

源：Reader

目的：Writer

3、明确具体设备。

源：

键盘：System.in

目的：

控制台：System.out

```
InputStream in = System.in;
OutputStream out = System.out;
```

4、需要额外功能？

需要，转换。因为都是字节流，但是操作的却是文本数据。所以使用字节流操作起来更为便捷。

```
InputStreamReader isr = new InputStreamReader(System.in);
OutputStreamWriter osw = new OutputStreamWriter(System.out);
```

需要，高效。

```
BufferedReader bufr = new BufferedReader(new InputStreamReader(System.in));
BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(System.out));
```

练习：将一个中文字符串数据按照指定的编码表写入到一个文本文件中。

1、目的：OutputStream，Writer

2、是纯文本：Writer

3、设备：硬盘File

```
FileWriter fw = new FileWriter("a.txt");
fw.write("你好");
```

P.S.

FileWriter是用来写入字节文件的便捷类，此类的方法假定默认字符编码和默认字节缓冲区大小都是可接受的。

代码：

```
01. import java.io.FileWriter;
02. import java.io.IOException;
03.
04. public class TransStreamDemo {
05.     public static void main(String[] args) throws IOException {
06.         writeText();
07.     }
08.
09.     public static void writeText() throws IOException {
10.         FileWriter fw = new FileWriter("c.txt");
11.         fw.write("你好");
12.         fw.close();
13.     }
14. }
复制代码
```

运行结果：



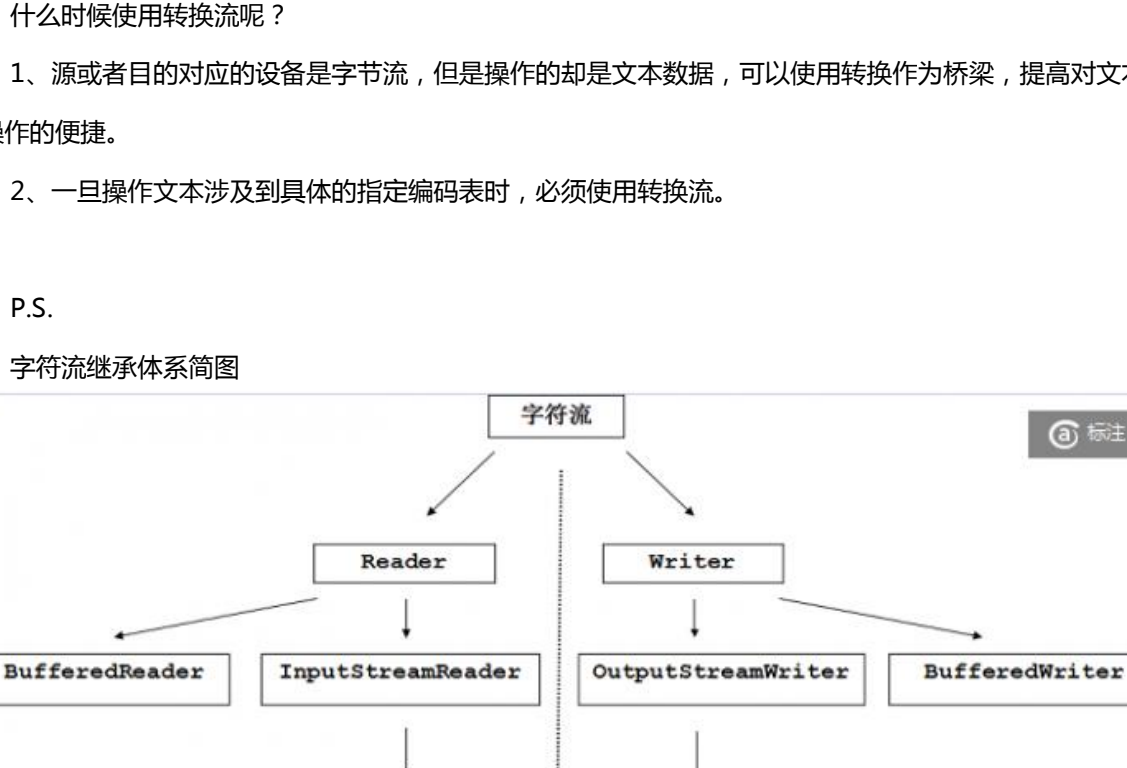
任何Java识别的字符数据使用的都是Unicode码表，但是FileWriter写入本地文件使用的是本地编码，也就是GBK码表。

而OutputStreamWriter可使用指定的编码将要写入的字符编码成字节。

代码：

```
01. import java.io.FileOutputStream;
02. import java.io.IOException;
03. import java.io.OutputStreamWriter;
04.
05. public class TransStreamDemo {
06.     public static void main(String[] args) throws IOException {
07.         writeText();
08.     }
09.
10.     public static void writeText() throws IOException {
11.         //下面这句代码等同于FileWriter fw = new FileWriter("b.txt");
12.         //FileWriter其实就是转换流指定了本机默认码表的体现，而且这个转换流的
13.         //对象，可以方便操作文本文件。
14.         //简单说：操作文件的字节流+本机默认的编码表。
15.         //这是按照默认码表来操作文件的便捷类
16.         //OutputStreamWriter osw = new OutputStreamWriter(new
17.         //FileOutputStream("d.txt"),"GBK");
18.         //如果操作文本文件需要明确具体的码表，FileWriter就不行了，必须用转换
19.         //流。
20.         OutputStreamWriter osw = new OutputStreamWriter(new
21.         FileOutputStream("d.txt"), "UTF-8");
22.         osw.write("您好");
23.         osw.close();
24.     }
25. }
复制代码
```

运行结果：



P.S.

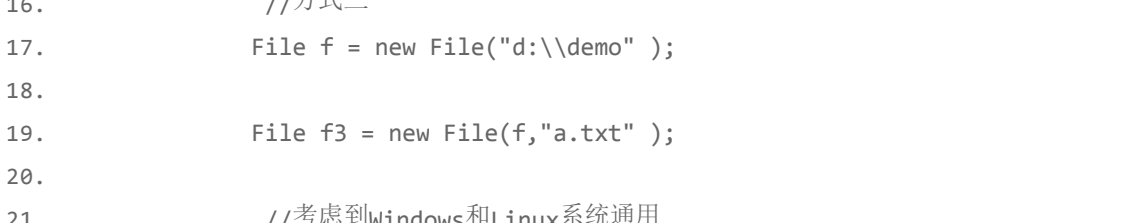
UTF-8编码，一个中文三个字节。

需求：打印d.txt文件中的内容至控制台显示。

代码：

```
01. import java.io.FileReader;
02. import java.io.IOException;
03.
04. public class TransStreamDemo {
05.     public static void main(String[] args) throws IOException {
06.         readText();
07.     }
08.
09.     public static void readText() throws IOException {
10.         FileReader fr = new FileReader("d.txt");
11.
12.         char[] buf = new char[10];
13.         int len = fr.read(buf);
14.         String str = new String(buf,0,len);
15.         System.out.println(str);
16.
17.         fr.close();
18.     }
19. }
复制代码
```

运行结果：



原因分析：由于d.txt文件中是UTF-8编码的“您好”，6个字节。

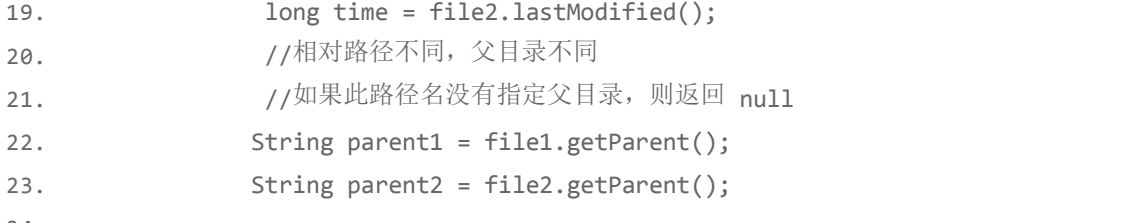
使用FileReader类是用GBK编码进行读取，6个字节代表3个字符，并且按照GBK进行解码，因此才出现如上结果。

需求：打印c.txt文件中的内容至控制台显示。

代码：

```
01. import java.io.FileInputStream;
02. import java.io.IOException;
03. import java.io.InputStreamReader;
04.
05. public class TransStreamDemo {
06.     public static void main(String[] args) throws IOException {
07.         writeText();
08.     }
09.
10.     public static void writeText() throws IOException {
11.         InputStreamReader isr = new InputStreamReader(new
12.         FileInputStream("c.txt"),"UTF-8");
13.
14.         char[] buf = new char[10];
15.         int len = isr.read(buf);
16.         String str = new String(buf,0,len);
17.         System.out.println(str);
18.
19.         isr.close();
20.     }
21. }
复制代码
```

运行结果：



原因分析：由于c.txt文件中是GBK编码的“您好”，4个字节。

使用InputStreamReader类是用UTF-8编码进行读取，由于GBK编码的字节使用UTF-8没有对应的字符，因此使用“？”进行代替。

需求：按照指定编码写字符到指定文件中。

如果需求中已经明确指定了编码表，那就不能使用FileWriter，因为FileWriter内部使用的默认的本地的码表。只能使用其父类，OutputStreamWriter。

OutputStreamWriter接收一个字节输出流对象，既然是操作文件，那么该对象应该是FileOutputStream。

```
OutputStreamWriter osw = new OutputStreamWriter(new
FileOutputStream("a.txt"),charsetName);
```

并且需要高效：

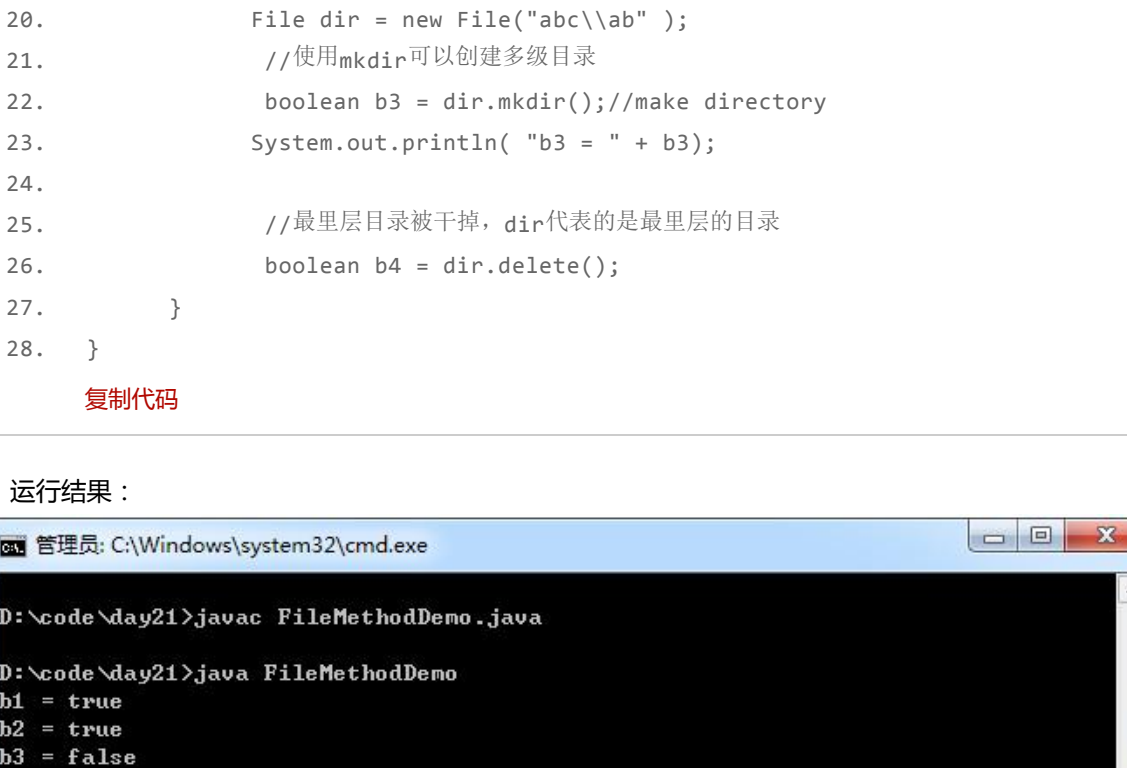
```
BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream("a.txt"),charsetName));
```

什么时候使用转换流呢？

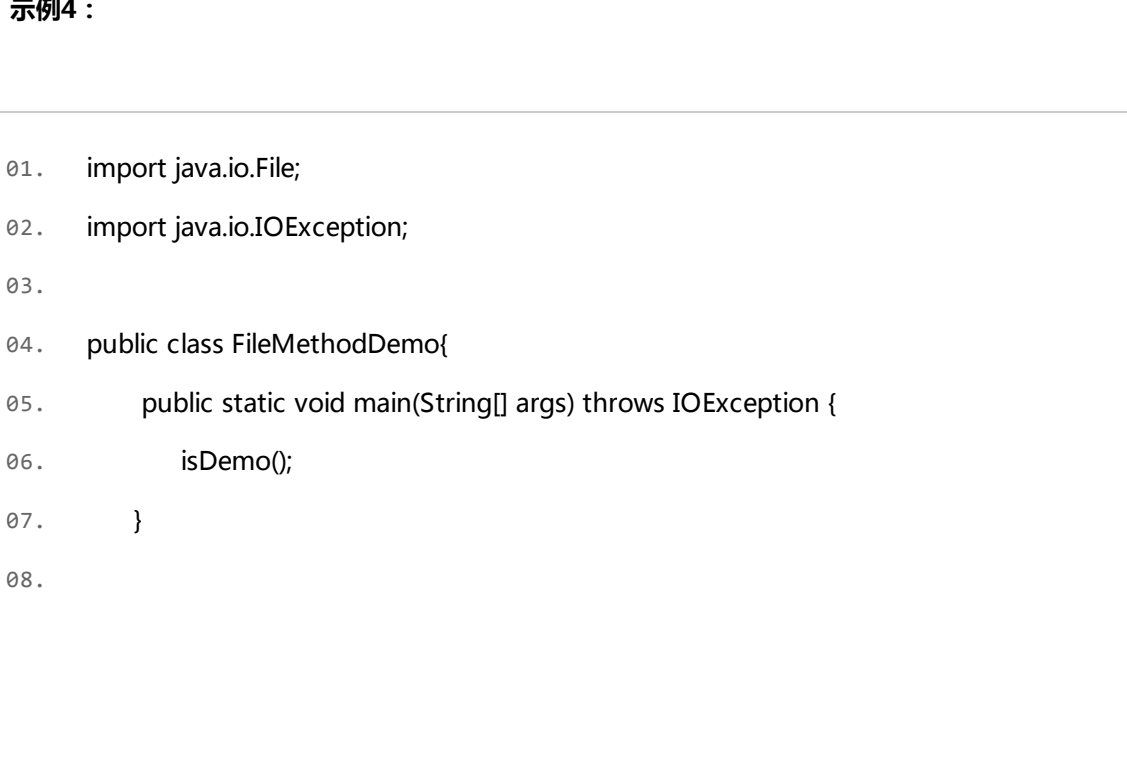
- 1、源或者目的对应的设备是字节流，但是操作的却是文本数据，可以使用转换作为桥梁，提高对文本操作的便捷。
- 2、一旦操作文本涉及到具体的指定编码表时，必须使用转换流。

P.S.

字节流体系架构图



字节流继承体系图



8.2.7 File类

File类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。

File对象可以作为参数传递给流的构造函数。

P.S.

流只能操作数据，不能操作文件。

示例1：

```
01. import java.io.File;
02.
03. public class FileDemo{
04.     public static void main(String[] args){
05.         constructorDemo();
06.     }
07.
08.     public static void constructorDemo(){
09.         //可以将一个已存在的、或者不存在的文件或者目录封装成File对象
10.         //方式一
11.         File f1 = new File("demo\\a.txt");
12.
13.         //方式二
14.         File f2 = new File("d:\\demo","a.txt");
15.
16.         //方式三
17.         File f = new File("d:\\demo");
18.
19.         File f3 = new File(f,"a.txt");
20.
21.         //考虑到Windows和Linux系统通用
22.         File f4 = new File("d:" + File.separator + "demo" +
23.         File.separator + "a.txt");
24.     }
25. }
复制代码
```

运行结果：

P.S.

File.separator是与系统有关的默认名称分隔符。在UNIX系统上，此字段的值为“/”；在Microsoft Windows系统上，它为“\\”。

File对象的常用方法：

- 1、获取
- 获取文件名称
- 获取文件路径
- 获取文件大小
- 获取文件修改时间

示例2：

```
01. import java.io.File;
02. import java.text.DateFormat;
03. import java.util.Date;
04.
05. public class FileMethodDemo{
06.     public static void main(String[] args){
07.         getDemo();
08.     }
09.
10.     public static void getDemo(){
11.         File file1 = new File("name.txt");
12.         File file2 = new File("d:\\demo\\a.txt");
13.
14.         String name = file2.getName();
15.         String absPath = file2.getAbsolutePath(); //绝对路径
16.         String path1 = file2.getPath();
17.         String path2 = file2.getParent();
18.         long len = file2.length();
19.         long time = file2.lastModified();
20.         //相对路径不同，父目录不同
21.         //如果路径名没有指定父目录，则返回 null
22.         String parent1 = file2.getParent();
23.         String parent2 = file2.getParent();
24.
25.         Date date = new Date(time);
26.         DateFormat df =
27.         DateFormat.getDateInstance(DateFormat.LONG,DateFormat.LONG);
28.         String str_time = df.format(date);
29.
30.         System.out.println("name:" + name);
31.         System.out.println("absPath:" + absPath);
32.         System.out.println("path1:" + path1);
33.         System.out.println("path2:" + path2);
34.         System.out.println("len:" + len);
35.         System.out.println("time:" + str_time);
36.         System.out.println("parent1:" + parent1);
37.         System.out.println("parent2:" + parent2);
38.     }
39. }
复制代码
```

运行结果：

2、创建和删除

示例3：

```
01. import java.io.File;
02. import java.io.IOException;
03.
04. public class FileMethodDemo{
05.     public static void main(String[] args) throws IOException {
06.         createAndDeleteDemo();
07.     }
08.
09.     public static void createAndDeleteDemo() throws IOException {
10.         File file = new File("file.txt");
11.
12.         //和输出流不一样，如果文件不存在，则创建，如果文件存在，则不创建
13.         boolean b1 = file.createNewFile();
14.         System.out.println("b1 = " + b1);
15.
16.         //使用delete方法删除文件类的时候，如果文件夹中有文件，则会删除失败
17.         boolean b2 = file.delete();
18.         System.out.println("b2 = " + b2);
19.
20.         File dir = new File("abc\\ab");
21.         //使用mkdir可以创建多级目录
22.         boolean b3 = dir.mkdir(); //make directory
23.         System.out.println("b3 = " + b3);
24.
25.         //顶层目录被干掉，dir代表的是最顶层的目录
26.         boolean b4 = dir.delete();
27.     }
28. }
复制代码
```

运行结果：

3、判断

示例4：

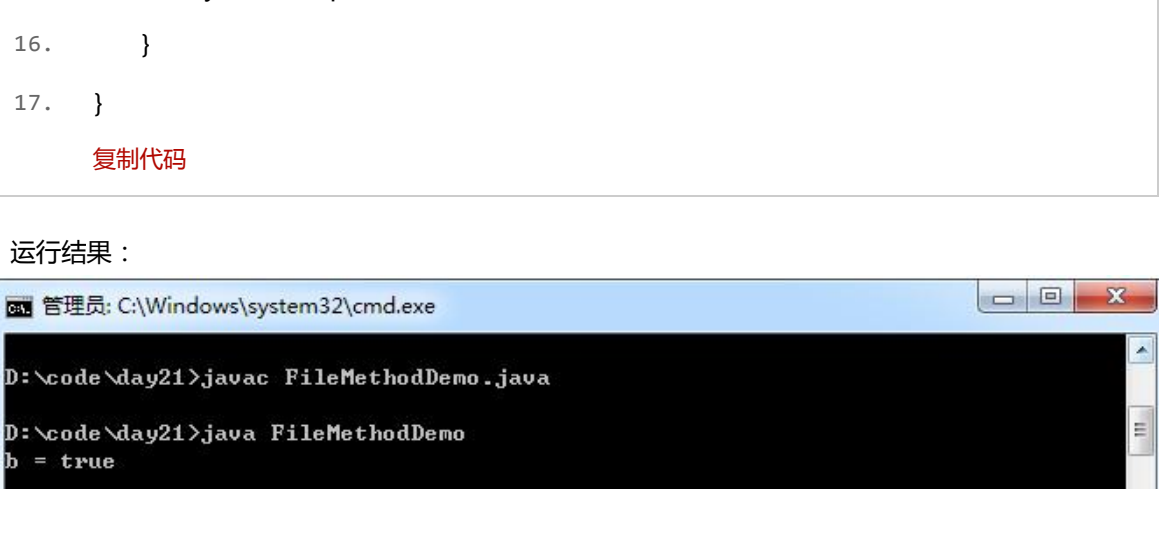
```
01. import java.io.File;
02. import java.io.IOException;
03.
04. public class FileMethodDemo{
05.     public static void main(String[] args) throws IOException {
06.         isDemo();
07.     }
08. }
复制代码
```



```
09.     public static void isDemo() throws IOException {
10.         File f = new File("aaa.txt" );
11.
12.         boolean b = f.exists();
13.         System.out.println( "b = " + b);
14.
15.         if(!f.exists()){
16.             f.createNewFile();
17.         }
18.
19.         //最好先判断是否存在
20.         if(f.exists()){
21.             System.out.println(f.isFile());
22.             System.out.println(f.isDirectory());
23.         }
24.
25.         f = new File("aa\\bb" );
26.
27.         f.mkdirs();
28.         if(f.exists()){
29.             System.out.println(f.isFile());
30.             System.out.println(f.isDirectory());
31.         }
32.     }
33. }
```

复制代码

运行结果：



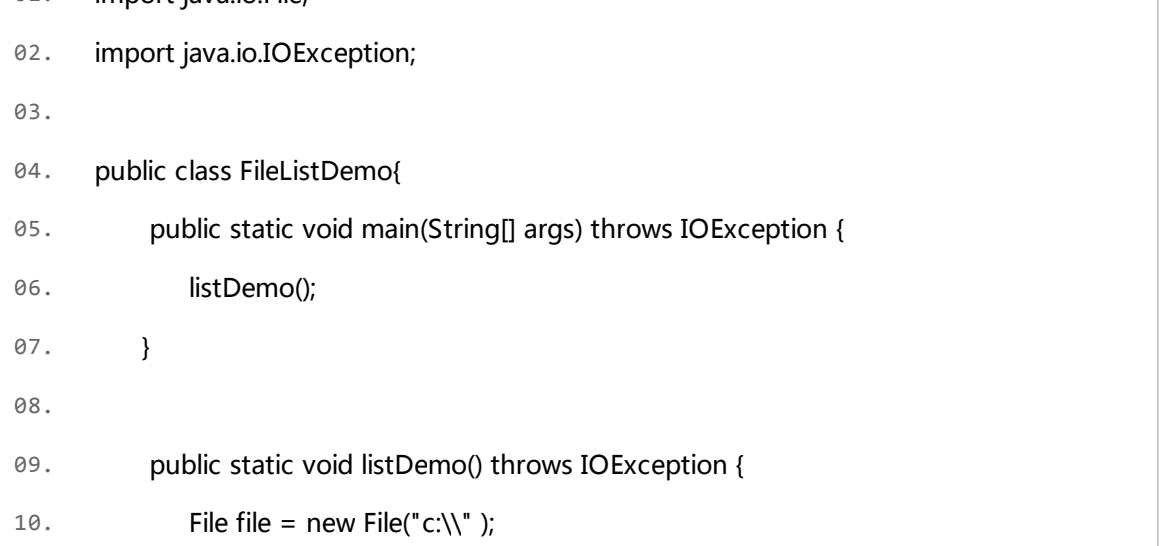
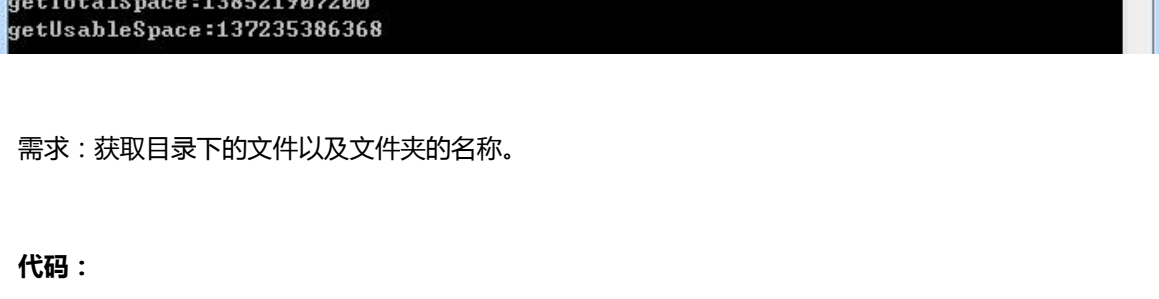
4、重命名

示例5：

```
01.     import java.io.File;
02.     import java.io.IOException;
03.
04.     public class FileMethodDemo{
05.         public static void main(String[] args) throws IOException {
06.             renameToDemo();
07.         }
08.
09.         public static void renameToDemo() throws IOException {
10.             File f1 = new File("d:\code\day21\0.mp3" );
11.             File f2 = new File("d:\code\day21\1.mp3" );
12.
13.             boolean b = f1.renameTo(f2);
14.
15.             System.out.println( "b = " + b);
16.         }
17.     }
```

复制代码

运行结果：



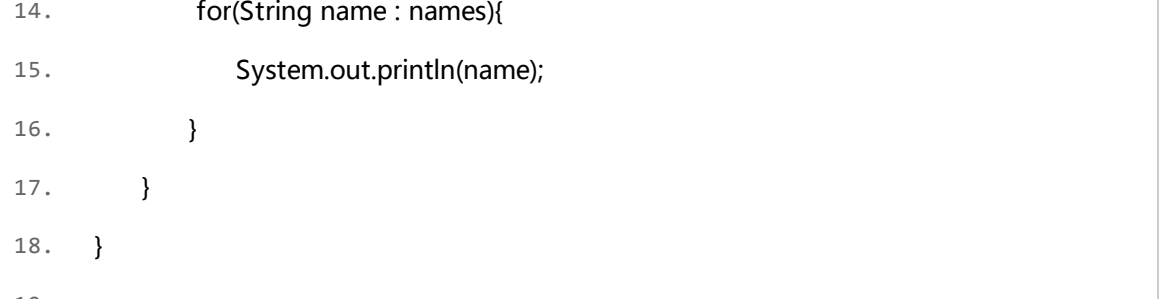
5、系统根目录和容量获取

示例6：

```
01.     import java.io.File;
02.     import java.io.IOException;
03.
04.     public class FileMethodDemo{
05.         public static void main(String[] args) throws IOException {
06.             listRootsDemo();
07.         }
08.
09.         public static void listRootsDemo() throws IOException {
10.             File[] files = File.listRoots();
11.
12.             for(File file : files){
13.                 System.out.println(file);
14.             }
15.
16.             File file = new File("d:\\");
17.
18.             System.out.println( "getFreeSpace" + file.getFreeSpace());
19.             System.out.println( "getTotalSpace" + file.getTotalSpace());
20.             System.out.println( "getUsableSpace" + file.getUsableSpace());
21.         }
22.     }
```

复制代码

运行结果：



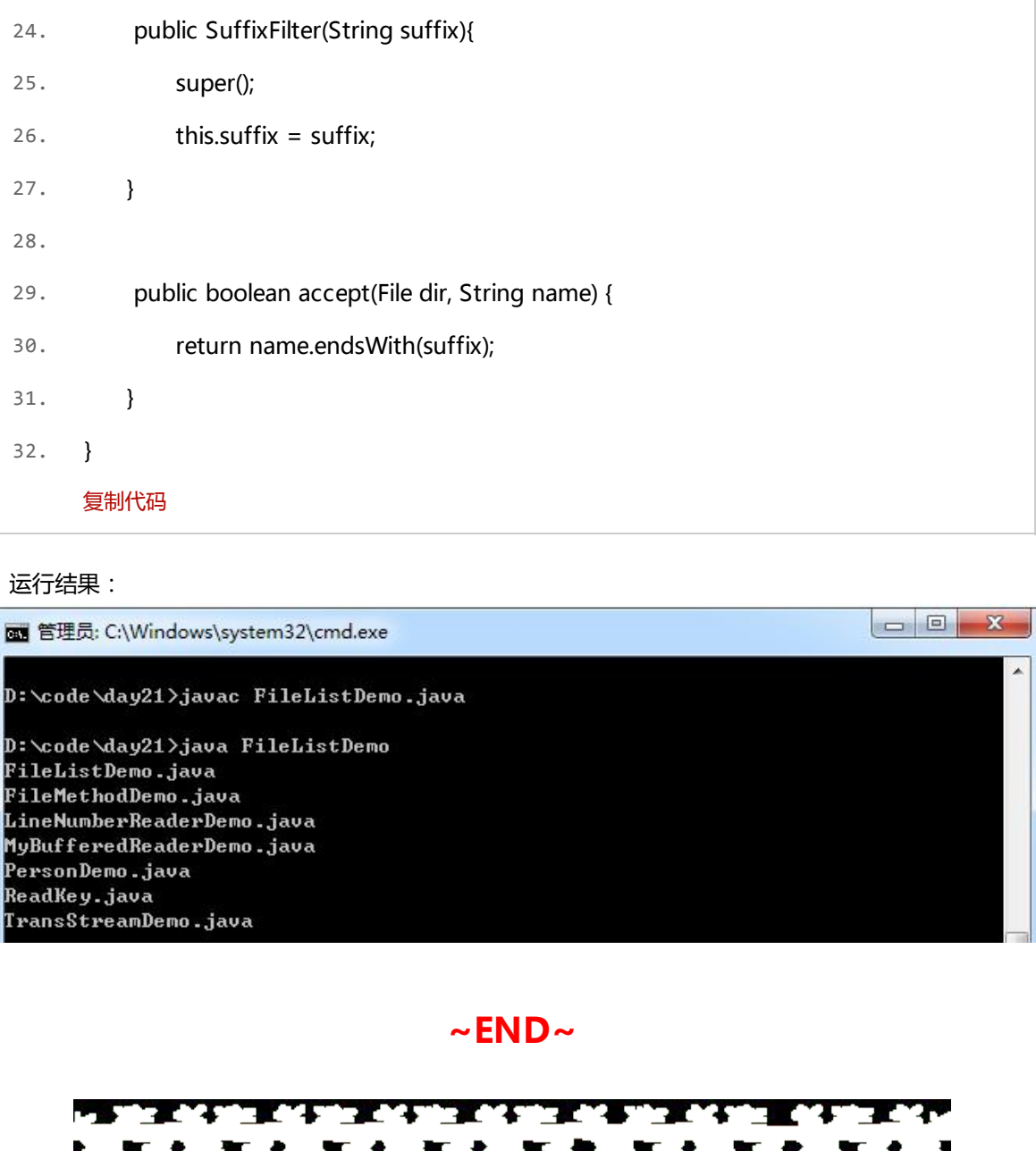
需求：获取目录下的文件以及文件夹的名称。

代码：

```
01.     import java.io.File;
02.     import java.io.IOException;
03.
04.     public class FileListDemo{
05.         public static void main(String[] args) throws IOException {
06.             listDemo();
07.         }
08.
09.         public static void listDemo() throws IOException {
10.             File file = new File("c:\\");
11.
12.             //获取目录下的文件以及文件夹的名称，包含隐藏文件
13.             //调用list方法的File对象中封装的必须是目录，否则会产生NullPointerException
14.             //如果访问的是系统级目录也会发生空指针异常
15.             //如果目录存在但是没有内容，会返回一个数组，但是长度为0
16.             String[] names = file.list();
17.
18.             for(String name : names){
19.                 System.out.println(name);
20.             }
21.         }
22.     }
```

复制代码

运行结果：



需求：获取d盘demo目录下后缀名为java的文件。

代码：

```
01.     import java.io.File;
02.     import java.io.FileNameFilter;
03.
04.     public class FileListDemo{
05.         public static void main(String[] args){
06.             listDemo();
07.         }
08.
09.         public static void listDemo(){
10.             File dir = new File("d:\code\day21" );
11.
12.             String[] names = dir.list( new FilterByJava());
13.
14.             for(String name : names){
15.                 System.out.println(name);
16.             }
17.         }
18.     }
19.
20.     class FilterByJava implements FileNameFilter{
21.         public boolean accept(File dir,String name){
22.             return name.endsWith("java" );
23.         }
24.     }
```

复制代码

运行结果：



~END~



~爱上海，爱黑马~

