

```
非贷款,0元入学,不1万就业不给1分钱学费,我们已干四年了!
```

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。

8, IO

8.2 IO流

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

示例3: import java.io.IOException; 01.

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。

02. import java.io.InputStream; 04. public class ReadKey{

public static void main(String[] args) throws IOException { 05. readKey(); 07. }

08. public static void readKey() throws IOException { 09. InputStream in = System.in; 10. int ch = in.read();//阻塞式方法 12. System.out.println(ch); 13. 14. ch = in.read(); //阻塞式方法 15. 16. System.out.println(ch); 17. ch = in.read(); //阻塞式方法 18. 19. System.out.println(ch); 20.

21. in.close(); 22. } 23. 复制代码 运行结果: - - X 画 管理员: C:\Windows\system32\cmd.exe D:\code\day21>javac ReadKey.java D:∖code∖day21>java ReadKey abc 97 98

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

因为是字节流处理的是文本数据,可以转换成字符流,操作更方便。 BfferedReader 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: import java.io.IOException; 01. import java.io.InputStream; 02.

03. public class ReadKey{ public static void main(String[] args) throws IOException { 05. 06. readKey(); 07. } 08. 09. public static void readKey() throws IOException { 10. //1、创建容器 11. 12. StringBuilder sb = new StringBuilder(); 13. //2、获取键盘读取流 14. InputStream in = System.in; 15.

//3、定义变量记录读取到的字节,并循环获取

//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

while((ch = in.read()) != -1){

continue; if(ch == '\n' ){

if(ch == '\r' )

int ch = 0;

16.

17. 18.

19. 20.

21.

22. 23.

24.

转换流的应用:

转换流:

示例1:

01.

02.

19.

21.

22.

23.

24.

25.

26.

27.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

38.

39.

40.

41.

运行结果:

abcde ABCDE fgklm FGKLM over

示例3:

01.

02.

03.

04.

05. 06.

07. 08. 09.

10.

07.

08.

09.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

public class TransStreamDemo{

FileOutputStream("a.txt")));

String line = null;

break;

bufw.newLine();

bufw.flush();

}

■ 管理员: C:\Windows\system32\cmd.exe

D:∖code∖day21>java TransStreamDemo

查看(V)

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

需求:将一个文本文件内容显示在控制台上。

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

public class TransStreamDemo{

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

public class TransStreamDemo{

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

import java.io.IOException;

01.

02.

03.

04.

05.

06.

07.

08.

09.

10.

11.

12.

▶ 下载

🧊 b.txt - 记事本

ABCDEFG

8.2.6 流的操作规律

3、明确具体的设备

源设备:

硬盘:File

内存:数组

目的设备:

硬盘:File

内存:数组

网络:Socket流

4、是否需要其他额外功能

是否需要高效(缓冲区):

键盘:System.in

网络:Socket流

a.txt

b.txt

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

到这里,就可以明确需求中具体要使用哪个体系。

import java.io.IOException;

包含到库中 ▼

D:∖code∖day21>javac TransStreamDemo.java

▶ 计算机 ▶ 本地磁盘 (D:) ▶ code ▶ day21

新建文件夹

工具(T) 帮助(H)

共享▼

名称

a.txt

}

复制代码

编辑(E)

}

运行结果:

abcdefg over

组织▼

☆ 收藏夹

🎒 a.txt - 记事本

**ABCDEFG** 

代码:

01.

02.

03.

04.

05.

06.

07.

08.

下载

while((line = bufr.readLine()) != null){

bufw.write(line.toUpperCase());

if("over" .equals(line))

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

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

00

++

类型

文本文档

0

修改日期

2015/6/7 16:05

搜索 day21

BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new

//数据到了osw,目的地out控制台

while((line = bufr.readLine()) != null){

bufw.write(line.toUpperCase());

if("over" .equals(line))

break;

bufw.newLine();

bufw.flush();

}

画 管理员: C:\Windows\system32\cmd.exe

D:\code\day21>java TransStreamDemo

D:∖code∖day21>javac TransStreamDemo.java

import java.io.BufferedReader; import java.io.BufferedWriter;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.io.IOException;

public class TransStreamDemo{

}

复制代码

以上代码可以简化为:

String line = null;

OutputStreamWriter osw = new OutputStreamWriter(out);

//将字符数据用缓冲区对象将数据写入缓冲区,目的地是osw

//osw.write(line.toUpperCase() + "\r\n");可以替代上面两行代码

\_ D X

BufferedWriter bufw = new BufferedWriter(osw);

字节流中的数据都是字符时,转成字符流操作更高效。

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。

import java.io.BufferedReader;

//读取到了字符串数据

break;

}

■ 管理员: C:\Windows\system32\cmd.exe

D:∖code∖day21>javac TransStreamDemo.java

}

复制代码

}

运行结果:

while((line = bufr.readLine()) != null){

System.out.println(line.toUpperCase());

- - X

if("over" .equals(line))

import java.io.IOException;

25. String temp = sb.toString(); if("over" .equals(temp)) 26. 27. break; 28. System.out.println(temp.toUpperCase()); sb.delete(0,sb.length()); 29. } else{ 30. //将读取到的字节存储到StringBuilder中 31. 32. sb.append(( char)ch); 33. } 34. } 35. } } 36. 复制代码 运行结果: \_ 0 X ■ 管理员: C:\Windows\system32\cmd.exe D:\code\day21>javac ReadKey.java D:\code\day21>java ReadKey abcd ABCD djk1s356 DJKLS356 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

03. import java.io.InputStream; import java.io.InputStreamReader; 04. 05. public class TransStreamDemo{ 06. 07. 08. public static void main(String[] args) throws IOException { 09. //字节流 InputStream in = System.in; 10. 11. 12. //将字节转成字符的桥梁,转换流 13. InputStreamReader isr = new InputStreamReader(in); 14. 15. //对字符流进行高效装饰,缓冲区 16. BufferedReader bufr = new BufferedReader(isr); 17. String line = null; 18.

D:∖code∖day21>java TransStreamDemo abcde ABCDE fgklm FGKLM ver 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是字符流通向字节流的桥梁。 示例2: import java.io.BufferedReader; 01. import java.io.BufferedWriter; 02. import java.io.IOException; 03. import java.io.InputStream; 04. 05. import java.io.InputStreamReader; import java.io.OutputStream; 06. 07. import java.io.OutputStreamWriter; 08. 09. public class TransStreamDemo{ 10. 11. public static void main(String[] args) throws IOException { 12. InputStream in = System.in; 13. 15. InputStreamReader isr = new InputStreamReader(in); 16. 17. //字符流 BufferedReader bufr = new BufferedReader(isr); 18. 19. 20. //字节流,字节数据 OutputStream out = System.out; 21.

BufferedReader bufr = new BufferedReader(new 11. InputStreamReader(System.in)); 12. BufferedWriter bufw = new BufferedWriter(new 13. OutputStreamWriter(System.out)); 14. 15. String line = null; 16. while((line = bufr.readLine()) != null){ 17. if("over" .equals(line)) 18. 19. break; 20. 21. bufw.write(line.toUpperCase()); bufw.newLine(); 22. 23. bufw.flush(); 24. 25. } 26. 复制代码 运行结果: - - X ■ 管理员: C:\Windows\system32\cmd.exe D:∖code∖day21>javac TransStreamDemo.java D:\code\day21>java TransStreamDemo abcde ABCDE fgklm FGKLM over 需求:将键盘录入的数据写入到一个文件中。 代码: import java.io.BufferedReader; 01. import java.io.BufferedWriter; 02. import java.io.FileOutputStream; 03. 04. import java.io.InputStreamReader; 05. import java.io.OutputStreamWriter; import java.io.IOException; 06.

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

09. public static void main(String[] args) throws IOException { 10. 11. BufferedReader bufr = new BufferedReader(new InputStreamReader(new 12. FileInputStream("a.txt"))); 13. BufferedWriter bufw = new BufferedWriter(new 14. OutputStreamWriter(System.out)); 15. String line = null; 16. 17. while((line = bufr.readLine()) != null){ 18. if("over" .equals(line)) 19. break; 20. 21. bufw.write(line.toUpperCase()); 22. bufw.newLine(); 23. bufw.flush(); 24. 25. } 26. } 27. 复制代码 运行结果: \_ \_ \_ X 画 管理员: C:\Windows\system32\cmd.exe D:∖code∖day21>javac TransStreamDemo.java D:∖code∖day21>java TransStreamDemo ABCDEFG 需求:将一个文本文件中的内容复制到另一个文件中。 代码:

13. BufferedReader bufr = new BufferedReader(new InputStreamReader(new FileInputStream("a.txt"))); 14. 15. BufferedWriter bufw = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("b.txt"))); 16. 17. String line = null; 18. while((line = bufr.readLine()) != null){ 19. 20. if("over" .equals(line)) break; 21. 22. bufw.write(line.toUpperCase()); 23. bufw.newLine(); 24. bufw.flush(); 25. 26. } 27. } 28. 复制代码 运行结果: ▼ 4 搜索 day2 编辑(E) 查看(V) 工具(T) 帮助(H) 劃打开▼ 新建文件夹 名称 类型 修改日期 ☆ 收藏夹

2015/6/7 16:05

2015/6/7 16:09

文本文档

文本文档

之所以要弄清楚这个规律,是因为流对象太多,开发时不知道用哪个对象合适。 想要知道对象的开发时用到哪些对象,只要通过四个明确即可。 1、明确源和目的 源:InputStream Reader 目的: OutputStream Writer 2、明确数据是否是纯文本数据 源:是纯文本:Reader 否: InputStream 目的:是纯文本:Writer 否: OutputStream

控制台: System.out

```
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 System.in;
    FileWriter fw = new FileWriter("b.txt");
  这样做可以完成,但是麻烦。将读取的字节数据转成字符串,再由字符流操作。
  4、需要额外功能吗?
    需要,转换。将字节流转成字符流,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:
      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、明确额外功能?
    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。
    InputSreamReader 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是用来写入字符文件的便捷类,此类的构造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;
       import java.io.IOException;
 02.
 03.
 04.
       public class TransStreamDemo {
          public static void main(String[] args) throws IOException {
 05.
             writeText();
 06.
 07.
          }
 08.
 09.
          public static void writeText() throws IOException {
 10.
             FileWriter fw = new FileWriter("c.txt");
             fw.write( "你好");
 11.
             fw.close();
 12.
 13.
          }
  14.
       复制代码
 运算结果:
                                                                  _ O X
 🔤 管理员: C:\Windows\system32\cmd.exe
 D:∖code∖day21>javac TransStreamDemo.java
 D:∖code∖day21>java TransStreamDemo
          🎳 ▶ 计算机 ▶ 本地磁盘 (D:) ▶ code ▶ day21
                                                                  - 44
                                                                         搜索 day
  文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
           打开▼
   组织▼
                      打印
                            新建文件夹
                      名称
                                                     修改日期
                                                                   类型
   ☆ 收藏夹
                      __ c.txt
     ▶ 下载
                                                     2015/6/7 16:18
                                                                   文本文档
            修改日期: 2015/6/7 16:18
        c.txt
                              创建日期: 2015/6/7 16:18
        文本文档
               大小: 4 字节
                                                                  🧻 c.txt - 记事本
 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
 你好
  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流中的字符编码成字节。
  代码:
       import java.io.FileOutputStream;
 01.
       import java.io.IOException;
 02.
       import java.io.OutputStreamWriter;
 03.
 05.
       public class TransStreamDemo {
             public static void main(String[] args) throws IOException {
 06.
                  writeText();
 07.
            }
 08.
 09.
             public static void writeText() throws IOException {
 10.
                   //下面这句代码等同于FileWriter fw = new FileWriter("b.txt");
 11.
                   //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
       子类对象,可以方便操作文本文件。
                   //简单说:操作文件的字节流+本机默认的编码表。
 13.
                   //这是按照默认码表来操作文件的便捷类
 14.
                   //OutputStreamWriter osw = new OutputStreamWriter(new
 15.
       FileOutputStream("d.txt"),"GBK");
 16.
                   //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
 17.
       流。
                  OutputStreamWriter osw = new OutputStreamWriter(new
 18.
       FileOutputStream("d.txt" ),"UTF-8" );
 19.
                  osw.write("您好");
 20.
                  osw.close();
 21.
 22.
            }
 23.
       复制代码
 运行结果:
                                                                  _ _ _ X
 画 管理员: C:\Windows\system32\cmd.exe
 D:∖code∖day21>javac TransStreamDemo.java
 D:\code\day21>java TransStreamDemo
        → → 计算机 → 本地磁盘 (D:) → code → day21
                                                                  ▼ 4 搜索 day
  文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
   组织▼ 打开
                    新建文件夹
                      名称
                                                     修改日期
                                                                   类型
   ☆ 收藏夹
     ▶ 下载
                      d.txt
                                                     2015/6/7 16:21
                                                                   文本文档
                              创建日期: 2015/6/7 16:21
            修改日期: 2015/6/7 16:21
        d.txt
               大小:6字节
 🧻 d.txt - 记事本
 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
 您好
  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;
       import java.io.IOException;
 02.
 03.
 04.
       public class TransStreamDemo {
             public static void main(String[] args) throws IOException {
 05.
                  readText();
 07.
            }
 08.
             public static void readText() throws IOException {
 09.
                  FileReader fr = new FileReader("d.txt" );
 10.
 11.
                   char[] buf = new char[10];
 12.
 13.
                   int len = fr.read(buf);
                  String str = new String(buf,0,len);
  14.
                  System.out.println(str);
 15.
 17.
                  fr.close();
            }
 18.
 19.
       复制代码
 运行结果:
                                                                  - - X
 ■ 管理员: C:\Windows\system32\cmd.exe
 D:∖code∖day21>javac TransStreamDemo.java
 D:\code\day21>java TransStreamDemo
  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;
     import java.io.IOException;
 02.
      import java.io.InputStreamReader;
 03.
 04.
 05.
     public class TransStreamDemo {
             public static void main(String[] args) throws IOException {
                  writeText();
 07.
 08.
            }
 09.
             public static void writeText() throws IOException {
 10.
                  InputStreamReader isr = new InputStreamReader(new
 11.
       FileInputStream("c.txt" ),"UTF-8" );
 12.
                  char[] buf = new char[10];
 13.
                  int len = isr.read(buf);
 14.
                  String str = new String(buf,0,len);
 15.
 16.
                 System.out.println(str);
 17.
 18.
                 isr.close();
 19.
            }
 20.
     }
       复制代码
  运行结果:
                                                                  - - X
 ■ 管理员: C:\Windows\system32\cmd.exe
 D:∖code∖day21>javac TransStreamDemo.java
 D:∖code∖day21>java TransStreamDemo
  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。那就不可以使用FileWriter,因为FileWriter内部使用的默认的本
地码表。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.
  字符流继承体系简图
                                    字符流
                                                                       ⓐ 标注
                         Reader
                                            Writer
                                                               BufferedWriter
 BufferedReader
                    InputStreamReader
                                         OutputStreamWriter
                      FileReader
                                            FileWriter
  字节流继承体系简图
                                  字节流
                    InputStream
                                           OutputStream
                                                               FileOutputStream
                   FilterInputStream
  FileInputStream
                                         FilterOutputStream
                 BufferedInputStream
                                          BufferedOutputStream
  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:
      import java.io.File;
 02.
 03. public class FileDemo{
            public static void main(String[] args){
 04.
                 constructorDemo();
 05.
 06.
            }
 07.
            public static void constructorDemo(){
 08.
                  //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
 09.
                  //方式一
 10.
                  File f1 = new File("d:\\demo\\a.txt" );
 12.
                  //方式二
 13.
                  File f2 = new File("d:\\demo" ,"a.txt" );
 14.
 15.
                  //方式三
                  File f = new File("d:\\demo" );
 17.
 18.
 19.
                 File f3 = new File(f, "a.txt" );
 20.
                  //考虑到Windows和Linux系统通用
 21.
                  File f4 = new File("d:" + File.separator + "demo" +
 22.
      File.separator + "a.txt" );
 23.
           }
 24.
       }
       复制代码
  运行结果:
                                                                  - - X

■ 管理员: C:\Windows\system32\cmd.exe

 D:\code\day21>javac FileDemo.java
 D:\code\day21>java FileDemo
  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;
       import java.util.Date;
 03.
      public class FileMethodDemo{
 05.
             public static void main(String[] args){
 06.
 07.
                  getDemo();
            }
 08.
             public static void getDemo(){
 10.
                  File file1 = new File("a.txt" );
 11.
                  File file2 = new File("d:\\demo\\a.txt" );
 12.
 13.
                  String name = file2.getName();
                  String absPath = file2.getAbsolutePath(); //绝对路径
 15.
                  String path1 = file1.getPath();
 16.
                  String path2 = file2.getPath();
 17.
                  long len = file2.length();
 18.
                  long time = file2.lastModified();
 19.
                  //相对路径不同,父目录不同
 20.
                  //如果此路径名没有指定父目录,则返回 null
 21.
 22.
                  String parent1 = file1.getParent();
                  String parent2 = file2.getParent();
 23.
 24.
                  Date date = new Date(time);
 25.
 26.
                  DateFormat df =
       DateFormat.getDateTimeInstance(DateFormat.LONG,DateFormat.LONG);
 27.
                  String str_time = df.format(date);
 28.
                  System.out.println( "name:" + name);
 29.
                  System.out.println( "absPath:" + absPath);
 30.
                  System.out.println( "path1:" + path1);
 32.
                  System.out.println( "path2:" + path2);
                  System.out.println( "len:" + len);
 33.
                  System.out.println( "str_time:" + str_time);
                  System.out.println( "parent1:" + parent1);
 35.
                  System.out.println( "parent2:" + parent2);
 36.
 37.
            }
       }
 38.
       复制代码
 运行结果:
                                                                  _ D X
 ■ 管理员: C:\Windows\system32\cmd.exe
 D:\code\day21>javac FileMethodDemo.java
  D:\code\day21>.java FileMethodDemo
 name:a.txt
 absPath:d:\demo\a.txt
 path1:a.txt
 path2:d:\demo\a.txt
 len:0
 str_time:1970年1月1日 上午08时00分00秒
 parent1:null
 parent2:d:\demo
  2、创建和删除
  示例3:
       import java.io.File;
 01.
       import java.io.IOException;
 02.
 03.
 04.
       public class FileMethodDemo{
             public static void main(String[] args) throws IOException {
 05.
 06.
                  createAndDeleteDemo();
 07.
            }
 08.
 09.
             public static void createAndDeleteDemo() throws IOException {
                  File file = new File("file.txt" );
 10.
 11.
                   //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
 12.
                   boolean b1 = file.createNewFile();
 13.
                  System.out.println( "b1 = " + b1);
 14.
 15.
                  //使用delete方法删除文件夹的时候,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" );
                  //使用mkdir可以创建多级目录
 21.
                  boolean b3 = dir.mkdir();//make directory
 22.
                  System.out.println( "b3 = " + b3);
 23.
 24.
                   //最里层目录被干掉,dir代表的是最里层的目录
 25.
 26.
                   boolean b4 = dir.delete();
 27.
            }
 28.
       }
       复制代码
  运行结果:
                                                                  - - X
 ■ 管理员: C:\Windows\system32\cmd.exe
 D:∖code∖day21>javac FileMethodDemo.java
 D:∖code∖day21>java FileMethodDemo
 b1 = true
 b2 = true
 b3 = false
  3、判断
  示例4:
       import java.io.File;
 01.
 02.
       import java.io.IOException;
 03.
       public class FileMethodDemo{
 04.
          public static void main(String[] args) throws IOException {
 05.
             isDemo();
 06.
 07.
          }
 08.
```

是,就加上buffer

需求1:复制一个文本文件

源:InputStream Reader

目的: OutputStream Writer

1、明确源和目的。

2、是否是纯文本?

源:Reader

目的:Writer

硬盘:File

硬盘:File

4、需要额外功能吗?

需要,高效

1、明确源和目的。

2、是否是纯文本?

是!

源:InputStream Reader

目的: OutputStream Writer

FileReader fr = new FileReader("a.txt");

FileWriter fw = new FileWriter("b.txt");

需求2:读取键盘录入信息,并写入到一个文件中

BufferedReader bufr = new BufferedReader(new FileReader("a.txt"));

BufferedWriter bufw = new BufferedWriter(new FileWriter("b.txt"));

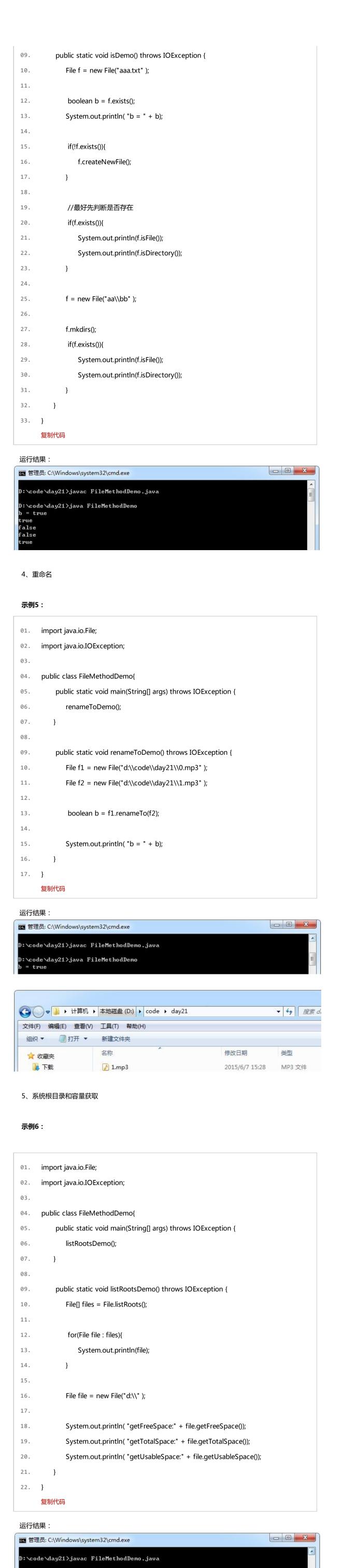
\_\_\_\_\_

3、明确具体设备。

是!

源:

目的:



public static void main(String[] args) throws IOException { listDemo(); 06. 07. } 08. 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 //调用list方法的File对象中封装的必须是目录,否则会产生NullPointerException 13. //如果访问的是系统级目录也会发生空指针异常 14. //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 15. String[] names = file.list(); 16. 17. for(String name: names){ 18. System.out.println(name); 19. } 20. 21. }

D:∖code∖day21>java FileMethodDemo

需求:获取目录下的文件以及文件夹的名称。

getFreeSpace:137235386368 getTotalSpace:138521907200 getUsableSpace:137235386368

import java.io.File;

import java.io.IOException;

public class FileListDemo{

C:\ D: \ E:\

代码:

01.

02.

03.

04.

05.

22.

}

运行结果:

\$Recycle.Bin Boot

Documents and Settings

bootmgr

Intel

03.

04.

05.

06.

07.

08.

09.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

03.

04.

05.

07.

08.

09.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

运行结果:

}

}

复制代码

FileListDemo.java

■ 管理员: C:\Windows\system32\cmd.exe

D:∖code∖day21>java FileListDemo

D:\code\day21>javac FileListDemo.java

}

}

}

}

public class FileListDemo{

listDemo();

public static void main(String[] args){

File dir = new File("d:\\code\\day21");

String[] names = dir.list( new SuffixFilter(".java" ));

public static void listDemo(){

for(String name: names){

class SuffixFilter implements FilenameFilter{

public SuffixFilter(String suffix){

public boolean accept(File dir, String name) {

return name.endsWith(suffix);

this.suffix = suffix;

private String suffix;

super();

System.out.println(name);

}

}

}

}

}

public class FileListDemo{

listDemo();

public static void main(String[] args){

File dir = new File("d:\\code\\day21");

String[] names = dir.list( new FilterByJava());

public static void listDemo(){

for(String name: names){

class FilterByJava implements FilenameFilter{

public boolean accept(File dir,String name){

return name.endsWith(".java");

System.out.println(name);

复制代码

画 管理员: C:\Windows\system32\cmd.exe

D:∖code∖day21>java FileListDemo

D:\code\day21>javac FileListDemo.java

myclass OEMSF pagefile.sys PerfLogs Program Files Program Files (x86) ProgramData System Volume Information win7 激活工具 Windows 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; import java.io.FilenameFilter; 02.

- - X

Ш

24. } 复制代码 运行结果: \_ D X ■ 管理员: C:\Windows\system32\cmd.exe D:\code\day21>javac FileListDemo.java D:\code\day21>java FileListDemo FileListDemo.java FileMethodDemo.java LineNumberReaderDemo.java MyBufferedReaderDemo.java PersonDemo.java ReadKey.java TransStreamDemo.java P.S. 由于搜索后缀为"java"便直接将"java"写死到代码中,不便于修改。 如果要求搜索后缀为".txt"的文件又该怎么办呢? 因此,为了提升程序的通用性,修改后的代码如下: 代码: import java.io.File; 01. import java.io.FilenameFilter; 02.

FileMethodDemo.java LineNumberReaderDemo.java MyBufferedReaderDemo.java PersonDemo.java ReadKey.java TransStreamDemo.java ~END~ ~爱上海,爱黑马~

\_ 0 X