

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!

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

8.2 IO流

8. IO

```
练习:
将d盘的一个文本文件复制到d盘。
```

分析:

03.

09.

10.

8.2.2 IO流常用基类-字符流

读取d盘demo.txt文件中的数据,将这些数据写入到d盘copyText_1.txt文件当中。既然是操作文本数 据,使用字符流。 方式1:使用read()读取文本文件数据。

代码: 01. import java.io.FileReader; 02. import java.io.FileWriter;

//2、创建一个目的,用于存储读到数据。

public class CopyTextTest{ 05. public static void main(String[] args) throws IOException { 06. //1、读取一个已有的文本文件,使用字符读取流和文件相关联 07. FileReader fr = new FileReader("demo.txt"); 08.

import java.io.IOException;

FileWriter fw = new FileWriter("copyText_1.txt"); 11. 12. //3、频繁的读写操作 13. 14. int ch = 0; while((ch = fr.read()) != -1){ 15. fw.write(ch); 17. } 18. //4、关闭字节流 19. 20. fw.close(); 21. fr.close(); 22. } 23. } 复制代码 运行结果: - 0 X ■ 管理员: C:\Windows\system32\cmd.exe D:\code\day21>javac CopyTextTest.java

D:\code\day21>java CopyTextTest 📗 ▶ 计算机 ▶ 本地磁盘 (D:) ▶ code ▶ day21 ¥ 44 编辑(E) 查看(V) 工具(T) 帮助(H) 共享▼ 组织▼ 包含到库中 ▼ 新建文件夹 名称 修改日期 类型 🎓 收藏夹 copyText_1.txt 2015/6/7 14:59 🧻 copyText_1.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

文本文档 abcde 方式2:使用read(char[])读取文本文件数据。 代码:

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

01.

02. 03.

04.

15.

23.

24.

25.

26. 27.

29.

30. 31.

字符流的缓冲区

对应类:

P.S.

09.

11. 12.

13.

14.

15.

16.

17.

18.

19. 20.

21. 22.

23.

24.

25. 26.

27.

02.

03.

04.

05.

06.

07.

08.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

}

运行结果:

复制代码

BufferedWriter

BufferedReader

缓冲区的出现提高了对数据的读写效率。

public class CopyTextTest{ 05. 06. private static final int BUFFER_SIZE = 1024; 07. 08. public static void main(String[] args){ 09. FileReader fr = null; FileWriter fw = null; 10. 11. try{ 12. fr = new FileReader("demo.txt"); 13. fw = new FileWriter("copytest_2.txt"); 14.

16. char[] buf = new char[BUFFER_SIZE]; 17. 18. //定义一个变量记录读取到的字符数(其实就是往数组里装的字符个 数) int len = 0; 20. 21. while((len = fr.read(buf)) != -1){ 22.

fw.write(buf,0,len);

throw new RuntimeException("读写失败!");

fw.close();

} catch(IOException e){

}

} finally{

} catch(Exception e){

if(fw != null){

try{

//创建一个临时容器,用于缓存读取到的字符

System.out.println(e.toString()); 32. 33. 34. } if(fr != null){ 35. 36. try{ 37. fw.close(); } catch(IOException e){ 38. 39. System.out.println(e.toString()); 40. } 42. } 43. 44. } 复制代码 运行结果: _ D X ■ 管理员: C:\Windows\system32\cmd.exe D:\code\day21>javac CopyTextTest.java D:\code\day21>java CopyTextTest → 计算机 → 本地磁盘 (D:) → code → day21 ¥ 64 搜索 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 copytest_2.txt 2015/6/7 15:02 文本文档 🧻 copytest_2.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) abcde

缓冲区要结合流才可以使用。 作用:在流的基础上对流的功能进行了增强。 示例6:提高写入效率,使用缓冲区。 01. import java.io.BufferedWriter; 02. import java.io.FileWriter; 03. import java.io.IOException; 04. public class BufferedWriterDemo{ 05. public static void main(String[] args) throws IOException{ 06. FileWriter fw = new FileWriter("buf.txt"); 07. 08.

//为了提高写入的效率,使用了字符流的缓冲区

for(int x = 1; x <= 4; x++){

bufw.newLine();

bufw.flush();

}

}

画 管理员: C:\Windows\system32\cmd.exe

bufw.flush();

fw.close();

D:∖code∖day21>javac BufferedWriterDemo.java

import java.io.FileNotFoundException;

public class BufferedReaderDemo{

String line = null;

}

}

复制代码

}

运行结果:

abcdef1 abcdef2 abcdef3 bcdef4

05.

06.

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.

33.

35.

运行结果:

文件(F)

组织▼

☆ 收藏夹

🎍 下载

夏面 東面

abcdef1 abcdef2 abcdef3 abcdef4

01. 02.

03.

05.

06.

07.

08. 09.

10. 11.

12. 13.

14. 15.

16. 17. 18.

19.

20.

104:

107: 108:

109: 110:

111: 112:

113: 114:

115:

示例:

02.

03. 04.

05. 06.

07.

08.

09.

10. 11. 12.

13.

15. 16.

17.

}

8.2.3 装饰设计模式

对原有类进行了功能的改变,增强。

void chifan(){

//采用装饰的方式增强Person类

//这个类的出现是为了增强Person而出现的

private Person p;

NewPerson(Person p){

this.p = p;

public void chifan(){

D:∖code∖day21>javac PersonDemo.java

装饰和继承都能实现一样的特点:进行功能的扩展增强。有什么区别呢?

|--BufferTextWriter:加入了缓冲技术的操作文本的对象

|--BufferMediaWriter:加入了缓冲技术的操作媒体的对象

这样就会发现只为提高功能,导致继承体系越来越臃肿,不够灵活。

以上方式并不理想,如果这个体系需要再进行功能扩展,又多了更多流对象。

可不可以将缓冲进行单独的封装,哪个对象需要缓冲就将哪个对象和缓冲关联。

如果想要对操作的动作进行效率的提高,按照面向对象,可以通过继承的方式对具体的对象进行功能

D:∖code∖day21>java PersonDemo

首先有一个继承体系:

|--TextWriter:用于操作文本

|--TextWriter:用于操作文本

|--MediaWriter:用于操作媒体

既然加入的都是同一种技术--缓冲。

class Buffer {

|--MediaWriter:用于操作媒体

可见:装饰比继承灵活。

练习:

分析:

05.

06. 07.

08.

09. 10.

12. 13.

14.

15.

16.

17.

18. 19.

20.

21.

22.

23.

25.

缓冲区中

}

中获取数据

|--BufferedWriter:用于提高效率

特点:装饰类和被装饰类都必须所属同一个接口或者父类。

自定义一个读取缓冲区类,模拟一个BufferedReader。

前一种是让缓冲和自己的流对象相结合。

Buffer(TextWriter w){}

的扩展,那么就需要加入缓冲技术。

|--MediaWriter:用于操作媒体

Writer

Writer

重新思考问题:

01.

02.

class Person{

}

class NewPerson{

}

运行结果:

■ buf_copy.txt - 记事本

复制代码

字符流缓冲区:

bufr.close();

■ 管理员: C:\Windows\system32\cmd.exe

D:∖code∖day21>javac BufferedReaderDemo.java

写入换行使用BufferedWriter类中的newLine()方法。

public class CopyTextBufTest{

String line = null;

bufw.write(line);

bufw.newLine();

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

bufw.write(ch);

bufw.flush();

//方式一

}

}

*/

bufr.close();

bufw.close();

■ 管理员: C:\Windows\system32\cmd.exe

D:\code\day21>java CopyTextBufTest

___ 打开

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

import java.io.FileReader;

import java.io.IOException;

}

}

■ 管理员: C:\Windows\system32\cmd.exe

101:import java.io.FileReader; 102:import java.io.IOException;

复制代码

lnr.close();

D:\code\day21>javac LineNumberReaderDemo.java

String line = null;

lnr.setLineNumber(100);

D:\code\day21>java LineNumberReaderDemo

103:import java.io.LineNumberReader;

105:public class LineNumberReaderDemo{

import java.io.LineNumberReader;

public class LineNumberReaderDemo{

String line = null;

lnr.setLineNumber(100);

D:\code\day21>javac CopyTextBufTest.java

(□) □ → → 计算机 → 本地磁盘 (D:) → code → day21

新建文件夹

名称

buf.txt

buf_copy.txt

编辑(E) 查看(V) 工具(T) 帮助(H)

//方式二

int ch = 0;

D:∖code∖day21>java BufferedReaderDemo

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

BufferedReader bufr = new BufferedReader(fr);

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

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

System.out.println(line);

import java.io.FileReader;

D:∖code∖day21>java BufferedWriterDemo

bufw.write("abcdef" + x);

//使用缓冲区的刷新方法将数据刷目的地中

//关闭缓冲区,其实关闭的就是被缓冲的流对象

//写入内容换行方法: newLine();

BufferedWriter bufw = new BufferedWriter(fw);

//使用缓冲区的写入方法将数据先写入到缓冲区中

//创建了一个字符写入流的缓冲区对象,并且指定与要被缓冲的流对象相关联

- 0

- - X

_ _ _ X

类型

- - X

III

文本文档

文本文档

修改日期

2015/6/7 15:05

2015/6/7 15:10

→ 计算机 → 本地磁盘 (D:) → code → day21 ¥ 44 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 2015/6/7 15:05 buf.txt 🏮 下载 _ O 🎒 buf.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) abcdef1 abcdef2 abcdef3 abcdef4 示例7:提高读取效率,使用缓冲区。 01. import java.io.BufferedReader;

读取一行数据使用BufferedReader类中的readLine()方法。 bufr.read():这个read方法是从缓冲区中读取字符数据,所以覆盖了父类中的read方法。 bufr.readLine():另外开辟了一个缓冲区,存储的是原缓冲区一行的数据,不包含换行符。 原理:使用了读取缓冲区的read方法,将读取到的字符进行缓冲并判断换行标记,将标记前的缓冲数 据变成字符串返回。 示例8: 01. import java.io.BufferedReader; import java.io.BufferedWriter; 02. import java.io.FileReader; 03. 04. import java.io.FileWriter;

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

BufferedReader bufr = new BufferedReader(fr);

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

BufferedWriter bufw = new BufferedWriter(fw);

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

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

LineNumberReader 跟踪行号的缓冲字符输入流。此类定义了方法 setLineNumber(int) 和 getLineNumber(),它们可分 别用于设置和获取当前行号。 示例:

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

LineNumberReader lnr = new LineNumberReader(fr);

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

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

LineNumberReader Inr = new LineNumberReader(fr);

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

System.out.println("吃饭");

System.out.println("开胃酒");

FileReader fr = new FileReader("LineNumberReaderDemo.java");

System.out.println(lnr.getLineNumber() + ":" + line);

FileReader fr = new FileReader("LineNumberReaderDemo.java");

System.out.println(lnr.getLineNumber() + ":" + line);

p.chifan(); 18. System.out.println("甜点"); 19. 20. } 21. } 22. //采用继承的方式增强Person类 23. 24. class NewPerson2 extends Person{ public void chifan(){ 25. System.out.println("开胃酒"); 26. 27. super.chifan(); System.out.println("甜点"); 28. 29. 30. } 31. 32. public class PersonDemo{ 33. public static void main(String[] args){ Person p = new Person(); 34. NewPerson np1 = new NewPerson(p); 35. np1.chifan(); 36. 37. System.out.println("----"); 38. 39. 40. NewPerson2 np2 = new NewPerson2(); np2.chifan(); 41. 42. } 43. } 复制代码 运行结果: - - X ■ 管理员: C:\Windows\system32\cmd.exe

Buffer(MediaWriter w){} 04. } 复制代码 简化为: 01. 02. class BufferedWriter extends Writer{ BufferedWriter(Writer w){} 03. } 04. 复制代码 Writer |--TextWriter:用于操作文本

缓冲区中无非就是封装了一个数组,并对外提供了更多的方法对数组进行访问,其实这些方法最终操 作的都是数组的角标。 缓冲的原理: 其实就是从源中获取一批数据到缓冲区中,再从缓冲区中不断地取出一个一个数据。 在此次取完后,再从源中继续取一批数据进缓冲区,当源中的数据取完时,用-1作为结束标记。 代码: 01. import java.io.FileReader; 02. import java.io.IOException; 03.

import java.io.Reader; class MyBufferedReader{ private Reader r; //定义一个数组作为缓冲区 private char[] buf = new char[1024]; //定义一个指针用于操作这个数组中的元素,当操作到最后一个元素后,指针应该归零 private int pos = 0; //定义一个计数器用于记录缓冲区中的数据个数,当该数据减到0,就从源中继续获取数据到 private int count = 0; MyBufferedReader(Reader r){ this .r = r;//该方法从缓冲区中一次取一个字符 public int myRead() throws IOException { //从源中获取一批数据到缓冲区中,需要先做判断,只有计数器为0时,才需要从源 if (count == 0){

26. count = r.read(buf); 27. //每次获取数据到缓冲区后,角标归零 28. pos = 0; 29. } 30. 31. if (count < 0) 32. 33. return -1; 35. char ch = buf[pos]; 36. 37. pos++; count--; 38. 39. 40. return ch; 41. } 42. public String myReadLine() throws IOException { 43. StringBuilder sb = new StringBuilder(); 44. 45. int ch = 0; 46. 47. while $((ch = myRead()) != -1){}$ if (ch == '\r') 49. continue; 50. if (ch == '\n') 51. return sb.toString(); //将从缓冲区读到的字符,存储到缓存行数据的缓冲区中 52. sb.append((char)ch); 53. 54. } 56. if (sb.length() != 0){ 57. return sb.toString(); 58. return null ; 59. } 60. 61. public void myClose() throws IOException { 62. r.close(); 63. 64. } 65. } 66. 67. public class MyBufferedReaderDemo{ public static void main(String[] args) throws IOException { 68. FileReader fr = new FileReader("buf.txt"); 69. 70. MyBufferedReader bufr = new MyBufferedReader(fr); 71. 72. 73. String line = null ; 74. 75. while ((line = bufr.myReadLine()) != null){ System.out.println(line); 76. 77. } 78. 79 bufr.myClose(); 80. } 81. 复制代码 运行结果: _ 0 X ■ 管理员: C:\Windows\system32\cmd.exe D:∖code∖day21>javac MyBufferedReaderDemo.java D:\code\day21>java MyBufferedReaderDemo abcdef1 abcdef2 abcdef3 abcdef4 8.2.4 IO流常用基类-字节流 基本操作与字符流类相同。但它不仅可以操作字符,还可以操作其他媒体文件。 示例1: 01. import java.io.FileOutputStream; 02. import java.io.IOException; 03. public class ByteStreamDemo{ 04. public static void main(String[] args) throws IOException { 05. 06. demo_write(); } 07. 08. 09. public static void demo_write() throws IOException { 10. //1、创建字节输出流对象,用于操作文件 11. FileOutputStream fos = new FileOutputStream("bytedemo.txt"); 12. //2、写数据,直接写入到了目的地中 13. fos.write("abcdefg".getBytes()); 14. 15. //关闭资源动作要完成 16. fos.close(); 17. 18. } 19. 复制代码 运行结果: ■ 管理员: C:\Windows\system32\cmd.exe D:\code\day21>javac ByteStreamDemo.java D:\code\day21>java ByteStreamDemo ▶ 计算机 ▶ 本地磁盘 (D:) ▶ code ▶ day21 文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H) 组织▼ 包含到库中 ▼ 共享▼ 新建文件夹 名称 修改日期 类型 ☆ 收藏夹 🤚 下载 bytedemo.txt 2015/6/7 15:22 文本文档 🧻 bytedemo.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) abcdefg 示例2: 01. import java.io.FileInputStream; 02. import java.io.IOException; 03. public class ByteStreamDemo{ 04. public static void main(String[] args) throws IOException { 05. demo_read1(); 06. System.out.println("----"); 07. 08. demo_read2(); 09. System.out.println("----"); 10. demo_read3(); } 11. 12. //读取方式一 13. public static void demo_read1() throws IOException { 14. //1、创建一个读取流对象,和指定文件关联 15. FileInputStream fis = new FileInputStream("bytedemo.txt"); 16. 17. //打印字符字节大小,不过要少用,文件太大,可能内存溢出 18. 19. byte[] buf = new byte[fis.available()]; fis.read(buf); 20. System.out.println(new String(buf)); 21. 22. 23. fis.close(); 24. } 25. //读取方式二 26. 27. public static void demo_read2() throws IOException { 28. FileInputStream fis = new FileInputStream("bytedemo.txt"); 29. 30. //建议使用这种读取数据的方式 31. byte[] buf = new byte[1024]; 32. 33. int len = 0; 34. 35. while((len = fis.read(buf)) != -1){

字节流的缓冲区:同样是提高了字节流的读写效率。 练习: 通过几种方式对MP3的进行拷贝,比较它们的效率。 代码: 01. 02.

03.

04.

05.

06.

07.

08.

09.

10.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

41.

42.

43.

45.

}

复制代码

}

运行结果:

bufis.close();

bufos.close();

■ 管理员: C:\Windows\system32\cmd.exe

D:\code\day21>java CopyMp3Test

文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)

组织▼ 1开

및 最近访问的位置

☆ 收藏夹 ▶ 下载

直 桌面

D:\code\day21>javac CopyMp3Test.java

(□ □ □ → 計算机 → 本地磁盘 (D:) → code → day21

新建文件夹 名称

1.mp3

36.

37. 38.

39. 40.

41. 42.

45.

46. 47.

48. 49.

50.

51.

52.

53.

54. 55. 56.

}

运行结果:

abcdefg abcdefg

abcdefg

P.S.

复制代码

■ 管理员: C:\Windows\system32\cmd.exe

D:\code\day21>java ByteStreamDemo

D:∖code∖day21>javac ByteStreamDemo.java

import java.io.BufferedInputStream;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class CopyMp3Test{

copy_1();

copy_2();

int len = 0;

}

 $import\ java. io. Buffered Output Stream;$

}

//读取方式三

}

fis.close();

//一次读取一个字节

int ch = 0;

fis.close();

System.out.println(new String(buf,0,len));

FileInputStream fis = new FileInputStream("bytedemo.txt");

- - X

public static void demo_read3() throws IOException {

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

System.out.print((char)ch);

FileOutputStream、FileInputStream的flush方法内容为空,没有任何实现,调用没有意义。

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

FileInputStream fis = new FileInputStream("0.mp3");

FileOutputStream fos = new FileOutputStream("1.mp3");

public static void copy_1() throws IOException {

byte[] buf = new byte[1024];

while((len = fis.read(buf)) != -1){

fos.write(buf,0,len);

25. fis.close(); 26. fos.close(); 27. } 28. public static void copy_2() throws IOException { 29. FileInputStream fis = new FileInputStream("0.mp3"); 30. BufferedInputStream bufis = new BufferedInputStream(fis); 31. 32. FileOutputStream fos = new FileOutputStream("2.mp3"); 33. BufferedOutputStream bufos = new BufferedOutputStream(fos); 34. 35. int ch = 0; 36. 37. while((ch = bufis.read()) != -1){ 38. bufos.write(ch); 39. 40. }

_ _ _ X

大小

4,998 KB

4,998 KB

4,998 KB

▼ 4 / 搜索 day21

类型

MP3 文件

2015/6/7 15:28 MP3 文件

2015/6/7 15:29 MP3 文件

修改日期

~END~

wa of wa कोड केर कोड

~爱上海,爱黑马~

2015/6/7 15:29