# 文件输入输出流

### 1.编写一个程序，读取源代码文件的内容并在控制台输出。如果源文件不存在，则显示相应的错误信息。

public class test01 {

public static void main(String[] args) {

File f = new File("test01.java");//文件当前目录下，在eclipse下是该工程目录下。

try {

FileReader fr = new FileReader(f);//将文件读取到内容中

int m; //int包含char的范围

while((m=fr.read())!=-1){

System.out.print((char)(m)); //强制转为char

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

### 2.编写一个程序实现如下功能，从当前目录下的文件input.txt中读取80个字节（实际读到的字节数可能比80少）并将读来的字节写入当前目录下的文件output.txt中

public class test01 {

public static void main(String[] args) {

File f1 = new File("input.txt");

File f2 = new File("output.txt");

try {

FileInputStream fis = new FileInputStream(f1);

FileOutputStream fos = new FileOutputStream(f2);

byte[] temp = new byte[80];//定义一个字节数组

fis.read(temp);//读到内存中

fos.write(temp);//写到文件

fis.close();

fos.close();

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("运行结束");

}

}

### 3,使用java的输入/输出流技术将一个文本文件的内容按行读出，每读出一行就顺序添加行号，并写入到另一个文件中。

public class test01 {

public static void main(String[] args) {

File f1 = new File("input.txt");

File f2 = new File("output.txt");

try {

FileReader fr = new FileReader(f1);

FileWriter fw = new FileWriter(f2);

BufferedReader br = new BufferedReader(fr);

BufferedWriter bw = new BufferedWriter(fw);

String temp;

int i=1;//行号

while((temp=br.readLine())!=null){

bw.write(i+","+temp);

bw.newLine();//换行

i++;

}

bw.flush();//把缓冲区内容写到文件,如果没有这条语句，输出文件为空。

//使用缓存型流时操作完成后必须加上flush语句。

br.close();

bw.close();

br.close();

bw.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

System.out.println("运行结束");

}

}

4.编写一个程序，接收从键盘输入的数据，并把从键盘输入的内容写到input.txt文件中，如果输入"quit"，则程序结束。

public class Test {

public static void main(String[] args) {

File f = new File("input.txt");

try {

FileWriter fw = new FileWriter(f);

Scanner scanner = new Scanner(System.in);

String temp;

while(!((temp=scanner.nextLine()).equals("quit"))){

fw.write(temp);

}

fw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

5.编写一个程序实现如下功能，文件input.txt是无行结构（无换行符）的汉语文件，从input中读取字符，写入文件output.txt中，每10个字符一行（最后一行可能少于10个字）

public class test01 {

public static void main(String[] args) {

File f1=new File("input.txt");

File f2=new File("output.txt");

try {

FileReader fr=new FileReader(f1);

FileWriter fw=new FileWriter(f2);

char temp[]=new char[10];

int len;

while((len=fr.read(temp))!=-1)

{

if(len==10)

fw.write(new String(temp)+"\r\n");

/\*

\* windows下的文本文件换行符:\r\n linux/unix下的文本文件换行符:\r

\* Mac下的文本文件换行符:\n

\*/

else

fw.write(temp, 0, len);

}

fr.close();

fw.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

System.out.println("程序结束");

}

}

### 6.逐行读取汉字文件，复制到另一个文件

public class test01 {

public static void main(String[] args) {

File f = new File("input.txt");

File outFile=new File("output.txt");

InputStreamReader read = null;

OutputStreamWriter write=null;

try {

read = new InputStreamReader (new FileInputStream(f));//注意事先设置好input.txt的编码格式为UTF-8，否则输出乱码

write=new OutputStreamWriter(new FileOutputStream(outFile));

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

BufferedReader reader=new BufferedReader(read);

String line;

try {

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

System.out.println(line);

write.write(line+"\r\n");

}

write.flush();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}