文件的读入和读出-09Task2pro

最适合读写文件内容的是文件字符流。

字符流分为字符输入流和字符输出流（这两个都是抽象类，FileWriter 和 FileReader 分别为字符输出流和字符输入流的实现类。

FileReader（文件字符输入流）就是把一个一个字符从硬盘文件输入到内存中。

构造器形式有两种（建立输入流管道）：

public FileReader（File file）通过文件对象；

public FileReader（String pathname） 通过文件路径和文件接通。

常用的方法有：

public int read（）每次读取一个字符返回，如果没有则返回-1；

public int read（char[ ] buffer）每次用一个字符数组去读取数据，返回字符数组读取了多少个数据。没有数据则返回-1；

注意 catch FileNotFoundException！

为什么文件字符输入流不容易乱码？

因为文件中不管是英文字符还是中午字符都是当作一个字符处理，只要代码和文件的编码方式（一般是 UTF-8）一致，则不会出现乱码。

注意事项：

要真正打印出内容，采用字符数组方法时，需要用 String（buffer）

如 System.out.print(String(buffer,0,len); （意思是从 0 索引出字符开始，用 len 记住每次读取字符数，一直读到 len 为止）

FileWriter（文件字符输出流）：

文件字符输出流（覆盖管道）是把一个个的字符由内存输入到硬盘文件当中；

构造器形式：

public fileWriter（File file）；

public fileWriter（String filepath）；

public fileWriter（File file，boolean append）；（是否追加数据）

public fileWriter（String filepath，boolean append）；

方法：

void write（int c ）；写一个字符；

void write（String str）；写一个字符串；

void write（String str，int off，int len）；写一个字符串的一部分；

void write（char[ ] cbuf）写一个字符数组；

void write（char[ ] cbuf ,int off, int len）写一个字符数组的一部分；

注意事项：

1. 字符输出流写出数据后必须刷新流或者关闭流，写出去的数据才会生效。（数据会先写到缓冲区里）
2. 关闭流的操作包含了刷新。

下面是我将 Song 类写进文件和从文件中读取的代码和结果：

（采用 write（String str）把字符传入进去的）

```
.ass demo6 {  新*
.c static void main(String[] args) {  新*

:ry {
    Writer w = new FileWriter( fileName: "/Users/dugiu/Desktop/java/javapro/java1/src/com/java/Stream/
    w.write( str: "package com.java.Stream;");
    w.write( str: "\r\n");
    w.write( str: "import java.io.Serializable;");
    w.write( str: "\r\n");
    w.write( str: "public class Song implements Serializable {");
    w.write( str: "\r\n");
    w.write( str: " private String title;");
    w.write( str: "\r\n");
    w.write( str: " private String artist;");
    w.write( str: "\r\n");
    w.write( str: " private String album;");
    w.write( str: "\r\n");
    w.write( str: " private String genre;");
    w.write( str: "\r\n");
    w.write( str: " private int length;");
    w.write( str: "\r\n");
    w.write( str: "Song(String title, String artist, String genre, int year, int timesPlayed) {");
    w.write( str: "\r\n");
    w.write( str: " this.title = title;");
    w.write( str: "\r\n");
    w.write( str: " this.artist = artist;");
```

然后可以在文件中得到对应的结果（注意敲进换行符）

得到结果如下:

ain.java demo1.java demo2.java Song.java demo4.java demo6.java demo

```
package com.java.Stream;
import java.io.Serializable;
public class Song implements Serializable {
 private String title;
 private String artist;
 private String album;
 private String genre;
 private int length;
Song(String title, String artist, String genre, int year, int timesPlayed) {
 this.title = title;
 this.artist = artist;
 this.album = album;
 this.genre = genre;
 this.length = year;
} public String getGenre() {
 return this.genre;}
 public int getLength() {
 return this.length;}
public String toString() {
 return(title + " " + artist + " " + genre + " " + year + " " + timesPlayed);}
}
```

在读取数据的时候，采用的是利用循环，采取每次读取一个字符的方法最后从文本中把内容读到命令行中：

Reader r = new FileReader（文件路径）;

int s;

while（(s = r.read()!=-1）{

System.out.print((char)s); （不用 println 否则会读成一串，同时记得将 s 转换为 char 再读取出来)

}

最后结果展示

```
/Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java
 -javaagent:/private/var/folders/h1/r7r8n9h12531mw_jt978_lz40000gn/T/AppTranslocation/33040BDB-6E3A-4
 .jar=62690:/private/var/folders/h1/r7r8n9h12531mw_jt978_lz40000gn/T/AppTranslocation/33040BDB-6E3A-4
 .encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/duqiu/Desk
package com.java.Stream;
import java.io.Serializable;
public class Song implements Serializable {
 private String title;
 private String artist;
 private String album;
 private String genre;
 private int length;
Song(String title, String artist, String genre, int year, int timesPlayed) {
 this.title = title;
 this.artist = artist;
 this.album = album;
 this.genre = genre;
 this.length = year;
} public String getGenre() {
 return this.genre;}
 public int getLength() {
 return this.length;}
public String toString() {
 return(title + " " + artist + " " + genre + " " + year + " " + timesPlayed);}
}
进程已结束，退出代码为 0
```