Java09-Task2

InputStream OutStream 相关学习笔记：

InputStream、OutStream：

字节输入流，字节输出流。相当于把硬盘中文件和内存以字节形式进行数据转换。

其中这些均为抽象类，而相应的实现类如 FileInputStream 和 FileOutputStream 可以实现对文件数据的读取和写出。

字节输入流和字节输出流适合用来进行文本的复制，而字符输入流和字符输出流适合进行文本内容的读取。

以 FileInputStream 为例，可利用 read 方法进行一个个的字节读取。

首先建立一个文件字节输入流管道：

InputStream is = new FileInputStream（File file）或者

InputStream is2 = new FileInputStream（String Path）

然后通过 is 调用 read 方法，可以是 is.read() 代表一次读入一个字节；

也可以是 is.read(buffer) 其中 buffer 是预先创建的一个 byte[1024]数组，代表一次可读入三个字节。

注意读取汉字时，因为 UTF-8 编码，每个汉字由三个字节组成，所以单独用 read 方法时，可能会出现乱码现象。而运用数组，也可能会出现乱码，为了避免，可每次读取一个文本的所有代码。（这也是适合文本复制原因之一）

对题目的理解分析：

首先，关于什么叫做序列化对象，就是将对象的状态（属性和值）转换为字节的过程，用于存储在文件中。

在我的几次尝试中：

我先是用 songs 类的对象去调用 getSongs 方法，用一个 List<Song>对象来接该方法的返回值，再把该对象打出来，但是执行结果会抛出未实现 Serializable 的异常，（因为我只在 Songs 类后 implements 了 Serializable 接口，而没有在 Song 中实现，对应的对象当然会抛异常）

第二次尝试时考虑到了最好不用成员方法返回值进行序列化，因此我重新定义了一个类叫 songList，在该类中定了成员变量再相应的序列化。

如何实现对象的串行化到一个文本中？

首先要实现 Serializable 接口（但是仅起标记作用）；

需要使用 ObjectOutputStream，

如下所示：

```
try {
    FileOutputStream fos = new FileOutputStream( name: "/Users/dugiu/Desktop/java/javapro/java
    ObjectOutputStream out = new ObjectOutputStream(fos);
   out.writeObject(list);
    out.close();
    fos.close();
} catch (IOException e) {
    throw new RuntimeException(e);
}
```

从而实现将对象串行化，将对象的状态转换为一系列字节。

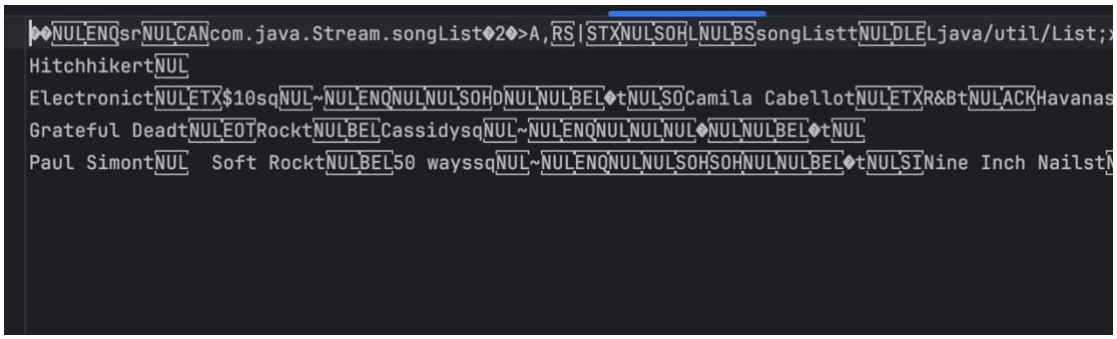
而如何反串行化？

利用 ObjecInputStream 的 readObject（）方法。如下所示：

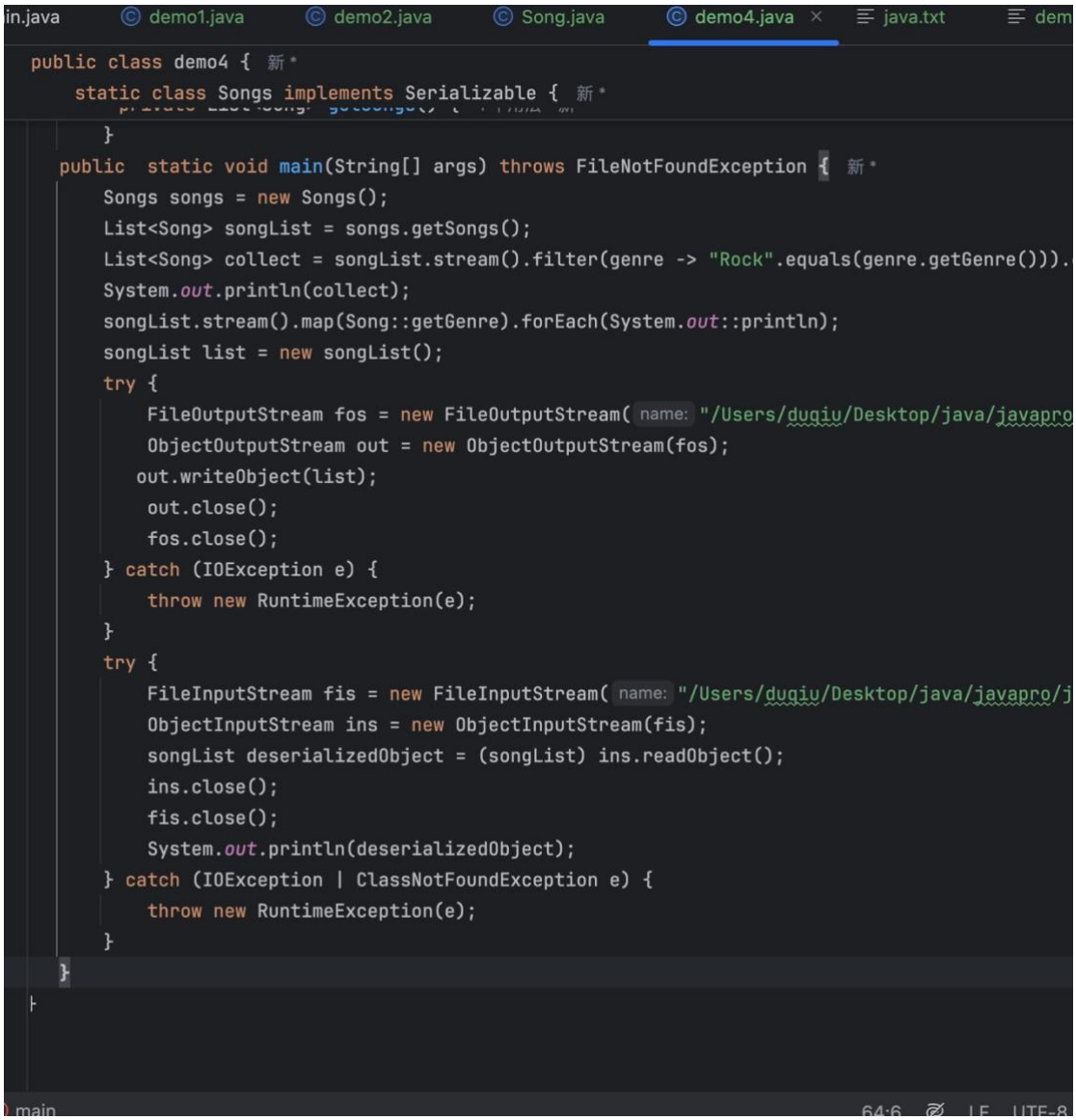
```
try {
    FileInputStream fis = new FileInputStream( name: "/Users/dugiu/Desktop/java/javapro/java1/
    ObjectInputStream ins = new ObjectInputStream(fis);
    songList deserializedObject = (songList) ins.readObject();
    ins.close();
    fis.close();
    System.out.println(deserializedObject);
} catch (IOException | ClassNotFoundException e) {
    throw new RuntimeException(e);
}
```

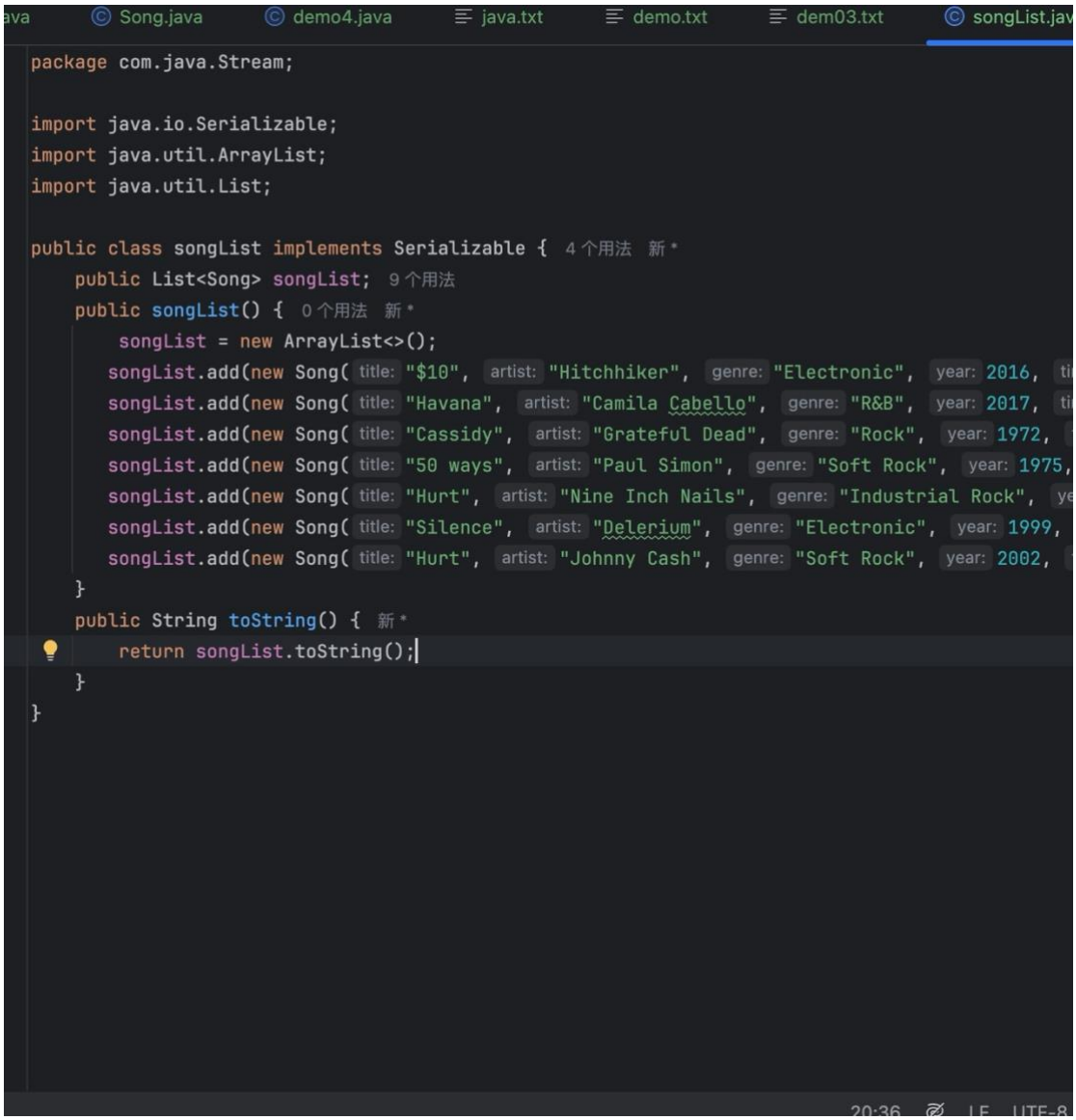
不足之处：

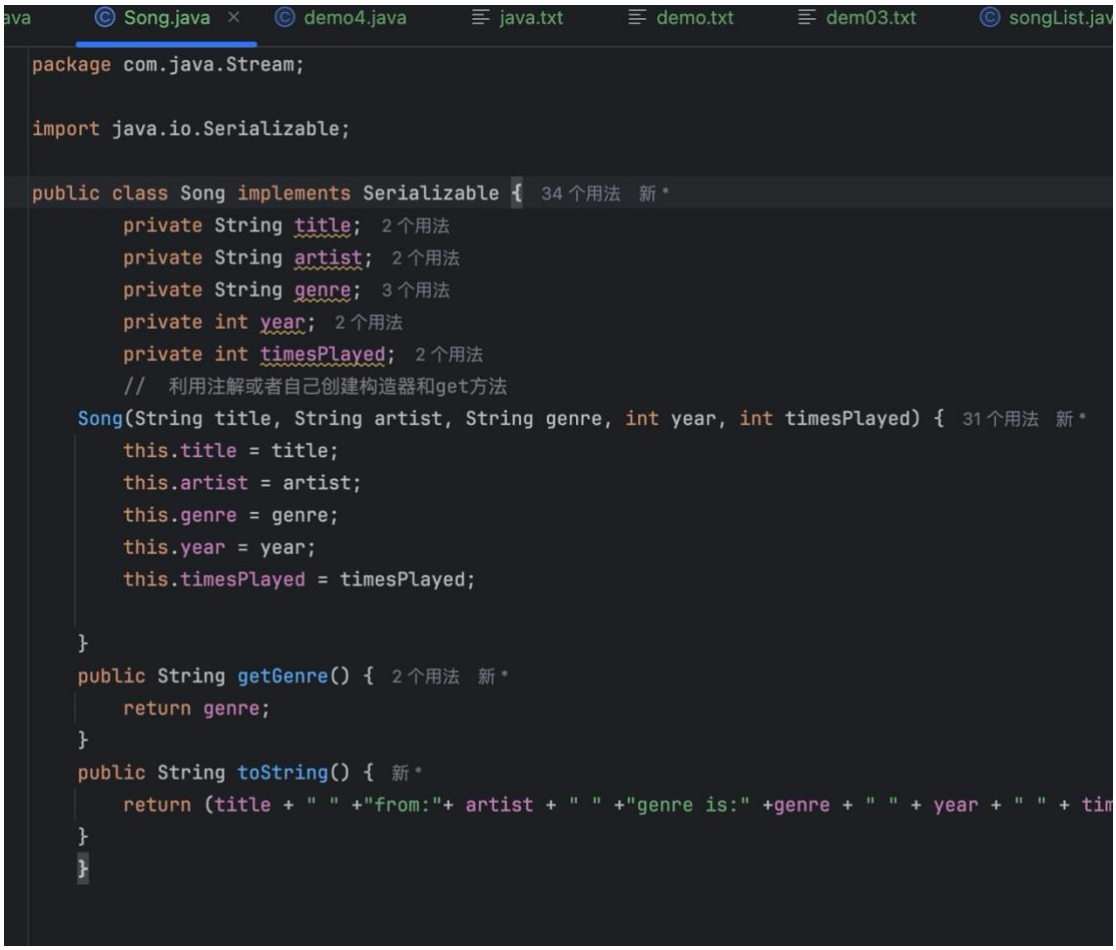
在将对象串行化后，得到了“乱码”（其实是特殊存法）（因为序列号和反序列号是基于二进制流的，不受制于任何编码形式），只有通过反序列化才可以将内容读取出来（我不是很清楚哪里出了问题，导致尝试过很多次，一直读不出来）



该部分的代码展现如下：


in.java
demo1.java
demo2.java
Song.java
demo4.java
java.txt
dem
public class demo4 { 新*
static class Songs implements Serializable { 新*
}
public static void main(String[] args) throws FileNotFoundException { 新*
Songs songs = new Songs();
List<Song> songList = songs.getSongs();
List<Song> collect = songList.stream().filter(genre -> "Rock".equals(genre.getGenre())).
System.out.println(collect);
songList.stream().map(Song::getGenre).forEach(System.out::println);
songList list = new songList();
try {
FileOutputStream fos = new FileOutputStream(name: "/Users/dugiu/Desktop/java/javapro
ObjectOutputStream out = new ObjectOutputStream(fos);
out.writeObject(list);
out.close();
fos.close();
} catch (IOException e) {
throw new RuntimeException(e);
}
try {
FileInputStream fis = new FileInputStream(name: "/Users/dugiu/Desktop/java/javapro/j
ObjectInputStream ins = new ObjectInputStream(fis);
songList deserializedObject = (songList) ins.readObject();
ins.close();
fis.close();
System.out.println(deserializedObject);
} catch (IOException | ClassNotFoundException e) {
throw new RuntimeException(e);
}
}
main
64:6
LF
UTF-8


ava
Song.java
demo4.java
java.txt
demo.txt
dem03.txt
songList.jav
package com.java.Stream;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
public class songList implements Serializable { 4 个用法 新 *
public List<Song> songList; 9 个用法
public songList() { 0 个用法 新 *
songList = new ArrayList<>();
songList.add(new Song(title: "$10", artist: "Hitchhiker", genre: "Electronic", year: 2016,
songList.add(new Song(title: "Havana", artist: "Camila Cabello", genre: "R&B", year: 2017,
songList.add(new Song(title: "Cassidy", artist: "Grateful Dead", genre: "Rock", year: 1972,
songList.add(new Song(title: "50 ways", artist: "Paul Simon", genre: "Soft Rock", year: 1975,
songList.add(new Song(title: "Hurt", artist: "Nine Inch Nails", genre: "Industrial Rock",
songList.add(new Song(title: "Silence", artist: "Delerium", genre: "Electronic", year: 1999,
songList.add(new Song(title: "Hurt", artist: "Johnny Cash", genre: "Soft Rock", year: 2002,
}
public String toString() { 新 *
return songList.toString();
}
}
20:36
LF
UTF-8


ava
Song.java
demo4.java
java.txt
demo.txt
dem03.txt
songList.jav
package com.java.Stream;
import java.io.Serializable;
public class Song implements Serializable { 34 个用法 新 *
private String title; 2 个用法
private String artist; 2 个用法
private String genre; 3 个用法
private int year; 2 个用法
private int timesPlayed; 2 个用法
// 利用注解或者自己创建构造器和get方法
Song(String title, String artist, String genre, int year, int timesPlayed) { 31 个用法 新 *
this.title = title;
this.artist = artist;
this.genre = genre;
this.year = year;
this.timesPlayed = timesPlayed;
}
public String getGenre() { 2 个用法 新 *
return genre;
}
public String toString() { 新 *
return (title + " " +"from:"+ artist + " " +"genre is:" +genre + " " + year + " " + tim
}
}