

| 学习目标

- * 能够使用转换流读取写入指定编码的文本文件

 - * `InputStreamReader(FileInputStream,"GBK");`

 - * `OutputStreamWriter(FileOutputStream,"GBK");`

- * 能够掌握编码表的意义

 - * ASCII(128),ISO-8859-1(256),GB2312(6763汉字),GBK(21003),Unicode(UTF-16,UTF-8)

- * 能够掌握SequenceInputStream的使用

 - * 几个流合并（文件合并）

- * 能够了解StringReader与StringWriter的使用

- * 能够掌握打印流的使用

 - * `PrintStream`

 - * `System.out.println();`

 - * `out:PrintStream`

 - * `PrintWriter`

- * 能够使用序列化流读取写出对象到文件

 - * `ObjectInputStream`

 - `readObject`

 - * `ObjectOutputStream`

 - `writeObject`

 - * `Serializable,uid`

 - * `transient`

- * 能够使用GZIPInputStream和GZIPOutputStream对文件进行压缩和解压

 - * 压缩和解压

 - * tomcat也使用这种，但是只是提供配置文件，让你配置

* 回顾

- * File

- * InputStream

```
* FileInputStream ,
```

ObjectInputStream,ByteArrayInputStream,SequenceInputStream,BufferedInputStream

- * OutputStream

* Reader

* FileReader,BufferedReader,InputStreamReader,StringReader

* Writer

* Property

* 装饰者模式

* 开闭原则

* 能够使用转换流读取写入指定编码的文本文件

```

1 * 字节流转换成字符流
2 * InputStreamReader, OutputStreamReader
3 * 使用BufferedReader--->InputStreamReader--->FileInputStream
4 public static void main(String[] args) throws IOException {
5     // 字节流怎么转换成字符流
6     BufferedReader br=new BufferedReader(
7         new InputStreamReader(
8             new FileInputStream("test6.txt"));
9     String line = br.readLine();
10    System.out.println(line);
11    br.close();
12 }
13 * 读不同文件由于编码不一样，会出现乱码
14 * 亮哥教育 （GBK）
15 * 浜ﻮ攢欒欒榑 （UTF-8）：用了GBK编码读了UTF-8的文件
16
17 * 读GBK的文件，写UTF-8的文件
18 public static void main(String[] args) throws IOException {
19     // 读GBK的文件，写UTF-8的文件
20     BufferedReader br = new BufferedReader(new InputStreamReader(new FileIn

```

```

21         BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(new FileO
22         String line=null;
23         while((line=br.readLine())!=null) {
24             bw.write(line);
25             bw.newLine();
26         }
27         bw.close();
28         br.close();
29     }
30

```

* 能够掌握编码表的意义

* 常见的概念

* 编码：按照某种规则，将字符存储到计算机中。

* 解码：将存储在计算机中的二进制数按照某种规则解析显示出来。

* 乱码：按照A规则存储，再按照B规则解析，就会导致乱码现象。

* 字符编码: 就是一套自然语言的字符与二进制数之间的对应规则。

* 编码表（字符集（Charset））：是一个系统支持的所有字符的集合，包括各国家文字、标点符号、图形符号、数字等。计算机要准确的存储和识别各种字符集符号，需要进行字符编码，一套字符集必然至少有一套字符编码。

* 常见字符集有ASCII编码表、ISO-8859-1编码表，GBK编码表、Unicode编码表（UTF-8，UTF-16）等。

```

1  * ASCII编码表
2      * 128 个
3  * ISO-8859-1编码表
4      * ISO 组织在 ASCII 码基础上又制定了一些列标准用来扩展 ASCII 编码
5      * 256 个字符
6      * 不支持中文
7  * GB2312
8      * 6763 个汉字
9  * GBK
10     * 21003 个汉字
11 * Unicode
12     * ISO 试图想创建一个全新的超语言字典，世界上所有的语言都可以通过这本字典来相互翻

```

```
13      * UTF-16
14      * 定长两个字节
15      * UTF-8
16      * 一种变长（一至四个字节为每个字符编码）
17
18 * 温馨提醒：
19      * 做项目的时候需要统一编码：建议使用UTF-8
```

* 能够掌握SequenceInputStream的使用

```
1 * 两个流拼接
2 public static void main(String[] args) throws IOException {
3     SequenceInputStream sis=new SequenceInputStream(new FileInputStream("te
4     BufferedReader br=new BufferedReader(new InputStreamReader(sis));
5     String line=null;
6     while((line=br.readLine())!=null) {
7         System.out.println(line);
8     }
9     br.close();
10 }
11
12 *多个流拼接
13 public static void main(String[] args) throws IOException {
14     Vector<FileInputStream> vector=new Vector<FileInputStream>();
15     vector.add(new FileInputStream("test1.txt"));
16     vector.add(new FileInputStream("test2.txt"));
17     vector.add(new FileInputStream("test3.txt"));
18     //Enumeration : 类似于迭代器
19     SequenceInputStream sis=new SequenceInputStream(vector.elements());
20     BufferedReader br=new BufferedReader(new InputStreamReader(sis));
21     String line=null;
22     while((line=br.readLine())!=null) {
23         System.out.println(line);
24     }
25     br.close();
26 }
27
```

* 能够了解StringReader与StringWriter的使用

```
1 * 了解
2 public static void main(String[] args) throws IOException {
3     // 把字符串存到文件
4     StringReader in=new StringReader("HelloWorld");
5     FileWriter out=new FileWriter("test8.txt");
6     copy(in, out);
7 }
8
9 public static void copy(Reader in,Writer out) throws IOException {
10     BufferedReader br=new BufferedReader(in);
11     String line=null;
12     while((line=br.readLine())!=null) {
13         out.write(line);
14     }
15     out.close();
16     br.close();
17 }
```

* 能够掌握打印流的使用

* 打印流提供了非常方便的打印功能，可以打印任何的数据类型

* 特点：打印，不抛异常

* PrintStream打印的目的：File对象，字符串路径，字节输出流。

* PrintWriter打印的目的：File对象，字符串路径，字节输出流, 字符输出流。

```
1 * 设置系统打印流的流向，输出到ps.txt
2     * System.out.println("");
3 public static void main(String[] args) throws IOException {
4     System.out.println("HelloWorld123");
5     PrintStream ps=new PrintStream("ps.txt");
6     // 设置系统打印流的流向，输出到ps.txt
7     System.setOut(ps);
8     System.out.println("HelloWorld");
9 }
10
```

```

11 public static void main(String[] args) throws IOException {
12     PrintStream ps=new PrintStream("ps12.txt");
13     ps.println("亮哥教育");
14     ps.println("做教育我们是认真的。");
15     ps.close();
16 }
17 public static void main(String[] args) throws IOException {
18     PrintWriter pw=new PrintWriter("pw123.txt");
19     pw.println("亮哥教育");
20     pw.println("做教育我们是认真的");
21     pw.close();
22 }

```

* 能够使用序列化流读取写出对象到文件

* 对象的输出流将指定的对象写入到文件的过程，就是将对象序列化的过程，对象的输入流将指定序列化好的文件读出来的过程，就是对象反序列化的过程

* 对象序列化：内存中对象写到硬盘文件

* 对象反序列化：硬盘文件变成内存对象

* 一个对象要想序列化，需要实现java.io.Serializable

* 不实现会报异常：NotSerializableException

* 一个对象属性假如不想被序列化，可以使用transient修饰

* Serializable 接口给需要序列化的类，提供了一个序列版本号。serialVersionUID 该版本号的目的在于验证序列化的对象和对应类是否版本匹配。

* 假如不匹配会报：InvalidClassException

```

1 * 测试
2 public class User implements Serializable{
3     private static final long serialVersionUID = -8798022641974233938L;
4     private int id;
5     private String name;
6     private transient int age;// 不序列化此属性
7     public User() {
8         super();

```

```
9      }
10     public User(int id, String name, int age) {
11         super();
12         this.id = id;
13         this.name = name;
14         this.age = age;
15     }
16     public int getId() {
17         return id;
18     }
19     public void setId(int id) {
20         this.id = id;
21     }
22     public String getName() {
23         return name;
24     }
25     public void setName(String name) {
26         this.name = name;
27     }
28     public int getAge() {
29         return age;
30     }
31     public void setAge(int age) {
32         this.age = age;
33     }
34     @Override
35     public int hashCode() {
36         final int prime = 31;
37         int result = 1;
38         result = prime * result + id;
39         result = prime * result + ((name == null) ? 0 : name.hashCode());
40         return result;
41     }
42     @Override
43     public boolean equals(Object obj) {
44         if (this == obj)
45             return true;
46         if (obj == null)
47             return false;
48         if (getClass() != obj.getClass())
```

```

49         return false;
50     User other = (User) obj;
51     if (id != other.id)
52         return false;
53     if (name == null) {
54         if (other.name != null)
55             return false;
56     } else if (!name.equals(other.name))
57         return false;
58     return true;
59 }
60 @Override
61 public String toString() {
62     return "User [id=" + id + ", name=" + name + ", age=" + age + "]";
63 }
64
65 }
66 * 序列化
67 public static void main(String[] args) {
68     try(ObjectOutputStream bos=new ObjectOutputStream(new FileOutputStream(
69         User user=new User(1001,"xiaohei",28);
70         bos.writeObject(user);
71     })catch (IOException e) {
72         e.printStackTrace();
73     }
74     System.out.println("序列化对象成功");
75 }、
76
77 * 反序列化
78 public static void main(String[] args) {
79     try(ObjectInputStream bis=new ObjectInputStream(new FileInputStream("us
80         User user=(User) bis.readObject();
81         System.out.println(user);
82     } catch (IOException e) {
83         e.printStackTrace();
84     } catch (ClassNotFoundException e) {
85         e.printStackTrace();
86     }
87 }
88

```



```

89 * 序列化集合
90 * 序列化
91 public static void main(String[] args) {
92     try(ObjectOutputStream bos=new ObjectOutputStream(new FileOutputStream(
93         List<User> list=new ArrayList<User>());
94         User u1=new User(1001,"xiaohei",28);
95         User u2=new User(1002,"xiaobai",29);
96         list.add(u1);
97         list.add(u2);
98         bos.writeObject(list);
99     }catch (IOException e) {
100         e.printStackTrace();
101     }
102     System.out.println("序列对象成功");
103 }
104
105 * 反序列化
106 public static void main(String[] args) {
107     try(ObjectInputStream bis=new ObjectInputStream(new FileInputStream("li
108         List<User> users=(List<User>) bis.readObject());
109         System.out.println(users);
110     } catch (IOException e) {
111         e.printStackTrace();
112     } catch (ClassNotFoundException e) {
113         e.printStackTrace();
114     }
115 }
116

```

* 能够使用GZIPInputStream和GZIPOutputStream对文件进行压缩和解压

```

1 * 压缩
2 public static void main(String[] args) {
3     //9.42 MB
4     String srcFile="E:\\avi\\01学习目标的介绍.wmv";
5     //7.78 MB
6     String destFile="E:\\\\avi\\\\01学习目标的介绍.zip";

```

```

7      try(GZIPOutputStream gout=new GZIPOutputStream(
8          new FileOutputStream(destFile));
9          FileInputStream fis=new FileInputStream(srcFile)){
10         // 流的读写
11         int len=0;
12         byte[] buffer=new byte[1024*8];
13         while((len=fis.read(buffer))!=-1) {
14             gout.write(buffer, 0, len);
15         }
16     }catch (IOException e) {
17         e.printStackTrace();
18     }
19     System.out.println("压缩结束...");
20 }
21
22 * 解压
23 public static void main(String[] args) {
24     //7.78 MB
25     String srcFile="E:\\avi\\01学习目标的介绍.zip";
26     //9.42 MB
27     String destFile="E:\\\\avi\\\\01学习目标的介绍.wmv";
28     try(GZIPInputStream gis=new GZIPInputStream(
29         new FileInputStream(srcFile));
30         FileOutputStream fos=new FileOutputStream(destFile)
31     ){
32         int len=0;
33         byte[] buffer=new byte[1024*8];
34         while((len=gis.read(buffer))!=-1) {
35             fos.write(buffer, 0, len);
36         }
37     } catch (IOException e) {
38         e.printStackTrace();
39     }
40     System.out.println("解压结束...");
41 }
42
43 * 工具类
44 public class GZipUtils {
45     public static void zip(File srcFile,File destFile) {
46         try(GZIPOutputStream gout=new GZIPOutputStream(

```

```
47         new FileOutputStream(destFile));
48         FileInputStream fis=new FileInputStream(srcFile)){
49             // 流的读写
50             int len=0;
51             byte[] buffer=new byte[1024*8];
52             while((len=fis.read(buffer))!=-1) {
53                 gout.write(buffer, 0, len);
54             }
55         }catch (IOException e) {
56             e.printStackTrace();
57         }
58     }
59
60     public static void zip(File srcFile) {
61         zip(srcFile, new File("temp.zip"));
62     }
63
64     public static void zip(String srcPath,String destPath) {
65         zip(new File(srcPath),new File(destPath));
66     }
67
68     public static void zip(String srcPath) {
69         zip(srcPath,"temp.zip");
70     }
71
72
73
74     public static void unZip(File srcFile,File destFile) {
75         try(GZIPInputStream gis=new GZIPInputStream(
76             new FileInputStream(srcFile));
77             FileOutputStream fos=new FileOutputStream(destFile)
78             ){
79             int len=0;
80             byte[] buffer=new byte[1024*8];
81             while((len=gis.read(buffer))!=-1) {
82                 fos.write(buffer, 0, len);
83             }
84         } catch (IOException e) {
85             e.printStackTrace();
86         }
```

```

87     }
88
89     public static void unZip(File srcFile) {
90         unZip(srcFile, new File("temp.zip"));
91     }
92
93     public static void unZip(String srcPath,String destPath) {
94         unZip(new File(srcPath), new File(destPath));
95     }
96
97     public static void unZip(String srcPath) {
98         unZip(srcPath,"temp.zip");
99     }
100
101 }
102
103 测试工具类:
104 public static void main(String[] args) {
105     //9.42 MB
106     String srcFile="E:\\avi\\01学习目标的介绍.wmv";
107     //7.78 MB
108     String destFile="E:\\avi\\01学习目标的介绍.zip";
109     GZipUtils.zip(srcFile, destFile);
110     System.out.println("压缩成功");
111 }
112
113 public static void main(String[] args) {
114     //7.78 MB
115     String srcFile="E:\\avi\\01学习目标的介绍.zip";
116     //9.42 MB
117     String destFile="E:\\\\avi\\\\01学习目标的介绍.wmv";
118
119     GZipUtils.unZip(srcFile, destFile);
120
121     System.out.println("解压成功...");
122 }

```

