Chp13 IO 框架

Key Point

- File 类
- 流的分类
- 基本字节流
- 字节过滤流
- 基本字符流、桥转换
- 字符过滤流
- 对象序列化

练习

1. （File 类）以下关于 File 类说法正确的是：
   A. 一个 File 对象代表了操作系统中的一个文件或者文件夹
   B. 可以使用 File 对象创建和删除一个文件
   C. 可以使用 File 对象创建和删除一个文件夹
   D. 当一个 File 对象被垃圾回收时，系统上对应的文件或文件夹也被删除

2. （File 类）有如下代码：

```
public class TestFile{
    public static void main(String args[]){
        File file = new File(“chp13/corejava.txt”);
    }
}
```

请选择一个正确答案：
   A. corejava.txt 文件在系统中被创建
   B. 在 windows 系统上运行出错，因为路径分隔符不正确
   C. corejava.txt 文件在系统中没有被创建
   D. 如果 corejava.txt 文件已存在，则抛出一个异常

3. （File 类）将下列代码补充完整

```
class TestMyFile{
    public static void main(String args[]) throws Exception{
        File file;
        //创建一个 File 对象表示当前目录下的“hello.txt”文件
        //判断该文件是否存在
        //如果该文件存在，则输出该文件的完整路径
    }
```

}

4. （流的分类）对于 FileInputStream 来说，从方向上来分，它是_____流，从数据单位上分，它是_____流，从功能上分，它是_____流。
5. （字节流, FileInputStream）FileInputStream 有三个重载的 read 方法，其中 1) 无参的 read 方法返回值为___类型，表示_____ 2) int read(byte[] bs)方法返回值表示_____，参数表示_____ 3) int read(byte[] bs, int offset, int len) 方法返回值表示_____，参数分别表示_____。

6. （FileInputStream）下面关于 FileInputStream 类型说法正确的是：
A. 创建 FileInputStream 对象是为了读取硬盘上的文件
B. 创建 FileInputStream 对象时，如果硬盘上对应的文件不存在，则抛出一个异常
C. 利用 FileInputStream 对象可以创建文件
D. FileInputStream 对象读取文件时，只能读取文本文件。

7. （FileOutputStream）填空：
创建 FileOutputStream 对象时，如果对应的文件在硬盘上不存在，则会_____；如果对应的文件在硬盘上已经存在，则_____；
如果使用 FileOutputStream(String path, boolean append) 这个构造方法创建 FileOutputStream 对象，并给定第二个参数为 true，则效果为_____。 创建 FileOutputStream 时_____（会|不会）产生异常。

8. 代码改错
```
class TestFileInputStream{
    public static void main(String args[]){
        FileInputStream fin = new FileInputStream(“test.txt”);
        try{
            System.out.println( fin.read() );
            fin.close();
        }catch(Exception e){}
    }
}
```

9. （FileInputStream 和 FileOutputStream）利用 FileInputStream 和 FileOutputStream，完成下面的要求：
1） 用 FileOutputStream 在当前目录下创建一个文件“test.txt”，并向文件输出“Hello World”，如果文件已存在，则在原有文件内容后面追加。
2）用 FileInputStream 读入 test.txt 文件,并在控制台上打印出 test.txt 中的内容。
3） 要求用 try-catch-finally 处理异常，并且关闭流应放在 finally 块中。

10. （Data 流）利用 Data 流，完成下面操作：

1） 判断当前目录下是否存在一个“test.dat”的文件，如果该文件不存在，则往该文件中写入一个 long 类型的数值：10000L

2） 如果该文件存在，则从该文件中读出数值，并把该数值加 1 之后，再存回文件中。

11. （字符流、桥转换）要想从某个文件中获得一个字符输出流，则至少有以下三种方式

A. 利用 FileWriter 类

B. 利用 PrintWriter 类

C. 利用 FileOutputStream 类，并通过 OutputStreamWriter 类获得 Writer

请简述这三种方式获得 Writer 的区别。

12. （字节流、字符流）以下几种文件格式，应当使用字节流还是字符流？

1）.java 源文件

2）.class 字节码文件

3）.html 网页文件

4）.jpg 图像文件

5）.mp3 音乐文件

6）配置文件.bash_profile

7）.jar 文件

13. （过滤流）连线题。把过滤流和相应的功能用线连起来。注意，左右两边不是一一对应

的关系。

ObjectInputStream	字节流
ObjectOutputStream	字符流
BufferInputStream	读八种基本类型
BufferedOutputStream	写八种基本类型
DataInputStream	读对象
DataOutputStream	写对象
PrintWriter	缓冲功能
PrintStream	读入一行文本
BufferedReader	写字符串并换行

14. （对象序列化）

为了让某对象能够被序列化，要求其实现_____接口；

为了让该对象某个属性不参与序列化，应当使用_____修饰符。

15. *（字符流、桥转换）完成下面功能：

事先在当前目录下准备好一个 test.txt 的文本文件，要求该文本文件是使用 GBK 编码的多行文本文件。如：

test.txt

窗前明月光

疑是地上霜

举头望明月
低头思故乡
利用字节流+桥转换读入这个文本文件，然后按照行的顺序，以 UTF-8 的编码方式，写到 test2.txt 文件中，例：
test2.txt
低头思故乡
举头望明月
疑是地上霜
窗前明月光

16. *（Data 流）有以下代码

```
public class Check{
    public static void main(String args[]) throws Exception{
        FileOutputStream fout = new FileOutputStream(“test.dat”);
        DataOutputStream dout = new DataOutputStream(fout);
        dout.writeInt(1);
        dout.writeDouble(0.01);
        dout.close();
    }
}
```

问：这个程序总共往文件中写入了多少字节？
A. 2
B. 8
C. 12
D. 16
E. 字节数取决于具体平台

17. *（对象序列化）
在 PrintWriter 中，有一个方法 print(Object obj)
在 ObjectOutputStream 中，有一个方法 writeObject(Object obj)
请简述这两个方法的区别

18. *（对象序列化）写出下面代码运行结果

```
import java.io.*;
class Address implements Serializable{
    private String addressName;
    private String zipCode;
    //构造方法
    //set/get 方法
    public String toString(){
        return addressName + “ ” + zipCode;
    }
}
class Student implements Serializable {
```

```
    private String name;
    private transient int age;
    private Address addr;
    //构造方法…
    //set/get 方法…
    public String toString(){
        return name + “ ” + age + “” + addr.toString();
    }
}
public class TestObjectStream{
    public static void main(String args[]) throws Exception{
        Address addr = new Address(“Beijing”, “100000”);
        Student stu = new Student(“Tom”, 18, addr);
        ObjectOutputStream oos = new ObjectOutputStream(
        new FileOutputStream(“stu.dat”) );
        oos.writeObject(stu);
        oos.close();

        ObjectInputStream oin = new ObjectInputStream(
        new FileInputStream(“stu.dat”) );
        Student stu2 = (Student) oin.readObject();
        oin.close();
        System.out.println(stu2);
    }
}
```

19. *（对象序列化）有以下代码：

```
import java.io.*;
class Address{
    private String addressName;
    private String zipCode;
    //构造方法…
    //get/set 方法…
    }
class Worker implements Serializable{
    private String name;
    private int age;
    private Address address;
    //构造方法…
    //get/set 方法…
}
public class TestSerializable {
    public static void main(String args[]) throws Exception{
        Address addr = new Address("Beijing", "100000");
```

```
        Worker w = new Worker("Tom", 18, addr);
        ObjectOutputStream oout = new ObjectOutputStream(
        new FileOutputStream("fout.dat") );
        oout.writeObject(w);
        oout.close();
    }
}
```

选择正确答案

A. 该程序编译出错

B. 编译正常，运行时异常

C. 编译正常，运行时也正常。

20. *（字节流，BufferedReader）完成下面操作。

在当前目录下创建一个 worldcup.txt 的文本文件，其格式如下：

2006/意大利

2002/巴西

…

该文件采用“年份/世界杯冠军”的方式保存每一年世界杯冠军的信息。

要求：读入该文件的基础上，让用户输入一个年份，输出该年的世界杯冠军。如果该年没有举办世界杯，则输出“没有举办世界杯”

21. **（Buffered 流，缓冲区）有下面代码

```
import java.io.*;
public class TestBufferedWriter{
    public static void main(String args[]) throws Exception{
        FileWriter fw = new FileWriter(“test.txt”);
        BufferedWriter bw = new BufferedWriter(fw);
        String str = “Hello World”;
        bw.write(str);
        /*1*/
    }
}
```

在/*1*/处放入什么代码，能够使得 test.txt 文件被正确写入？

A. bw.close()

B. bw.flush();

C. fw.close();

22. **（Data 流）在原有自动分配 id 的 Account 对象基础上，利用 Data 流，完成下面的要求：

要求每次启动程序时，id 的自动分配都能在上一次运行的基础上继续。例如，假设有以下代码：

```
public class TestAccount{
    public static void main(String args[]){
```

```
        Account a1 = new Account();
        Account a2 = new Account();
        Account a3 = new Account();
        System.out.println(a1.getId() );
        System.out.println(a2.getId() );
        System.out.println(a3.getId() );
    }
}
```

编译之后，第一次运行
java TestAccount 时，输出
100001
100002
100003
第二次运行
java TestAccount 时，输出
100004
100005
100006

23. **（综合）
从命令行中读入一个文件名，判断该文件是否存在。如果该文件存在，则在原文件相同路径下创建一个文件名为“copy_原文件名”的新文件，该文件内容为原文件的拷贝。 例如：读入 /home/java/photo.jpg 则创建一个文件 /home/java/copy_photo.jpg 新文件内容和原文件内容相同。

24. ***（可选，综合）用两种方式保存对象。
有 Worker 对象，部分代码如下：

```
class Worker{
    private String name;
    private int age
    private double salary;
    //构造方法
    …
    //get/set 方法
    …
    //toString 方法
    …
}
```

1） 完善 Worker 对象，并使其能使用对象序列化机制。
2） 利用 ObjectOutputStream 存入两个 Worker 对象， 然后使用

ObjectInputStream 读出这两个对象，并打印这两个对象的信息。
3） 写一个方法 saveWorkerToFile(Worker w, File file)，该方法完成下面的功能：
假设有一个 Worker 对象 w1，File 对象 f1 如下：
Worker w1 = new Worker(“Tom”, 30, 5000);
File f1 = new File(“test.txt”);
则调用 saveWorkerToFile(w1, f1)，会在 test.txt 中增加一行：
Tom/30/5000
4） 写一个方法 List<Worker> readWorkerFromFile(File file)，该方法读某个文件，从文件信息中创建一个 Worker 类型的 List。
例如，假设文件内容如下：
Tom/30/5000
Jim/25/3000
Terry/33/4500
则返回一个包含三个 Worker 对象的 List。