

第十一章 File类、IO流-作业

一、File 类

1.基础问答

以下方法都来自于java.io.File类，请简述它们的作用。

```
1 public String getAbsolutePath();
2 public String getPath();
3 public String getName();
4 public long length();
5 public boolean exists();
6 public boolean isFile();
7 public boolean isDirectory();
8 public boolean createNewFile();
9 public boolean mkdir();
10 public boolean mkdirs();
11 public boolean delete();
```

2.编程题

请根据注释中的功能要求和提示补全代码，实现功能。

```
1 import java.io.File;
2
3 public class Test012_ListFiles {
4     public static void main(String[] args) {
5         // 目录路径
6         String path = "D:/";
7         // 创建目录对象
8         File dir = new File(path);
9         // 请在此处补全代码，列出D盘根目录下所有后缀为txt或doc的文件
10        // 注意：1、排除掉目录；2、显示出每个文件的绝对路径
11
12    }
13 }
```

二、IO流

1.基础问答

根据数据的流向，可以将流分为哪些类型？简述它们的特点。

2.基础问答

根据每次处理数据的单位，可以将流分为哪些类型？简述它们的特点。

3.基础问答

JDK 为每种类型的流提供了抽象类以定义标准和规范，请说出他们分别是哪些类？

4.基础问答

InputStream类中定义了三个用于读取字节数据的方法，请回答以下问题：

- 1、简述每个方法的作用以及参数的含义
- 2、三个方法均为int类型返回值，这三种返回值代表的含义有何异同点？

```
1  int read();
2  int read(byte b[]);
3  int read(byte b[], int off, int len);
```

5.基础问答

请回答在什么情况下适合用字节流读写文件？什么情况下适合用字符流读写文件？

6.基础问答

内存缓冲流的优势有哪些？

7.编程题

请按要求完成程序编写，部分代码已给出，请完成剩余部分。

- 将一张图片复制10份，放置在同目录下
- 源文件名为img.jpg，输出文件命名为img_copy1.jpg、img_copy2.jpg.....img_copy10.jpg
- 选择合适的流对象实现功能，要求考虑到性能优化
- 严格遵循IO流的使用规范，进行资源的释放，避免内存泄漏

```
1
2  import java.io.File;
3  import java.io.FileInputStream;
4  import java.io.FileOutputStream;
5  import java.io.IOException;
6  import java.io.InputStream;
7  import java.io.OutputStream;
8
9  public class Test027_MultipleCopy {
10
11      public static void main(String[] args) {
12          // 源文件所在路径
13          String dirPath = "D:/test";
14          // 源文件名称
15          String fileName = "img.jpg";
```

```

16         // 复制数量
17         int number = 10;
18         // 完成剩余代码编写
19         // ...
20     }
21 }

```

8.编程题

编程实现：如果现在有一款只能试用10次的软件，超过10次之后就需要提醒用户购买正版软件。（程序运行一次，使用次数就要减一次）

提示：将试用的次数写到一个文件中，每次启动时对其进行读取并进行判断。

9.编程题

编写两个方法，分别实现如下功能：

- save()方法：创建三个Person类型对象，将它们使用序列化手段写入本地文件中持久化存储
- load()方法：将save()方法中保存的Person对象读取出来，打印输出属性

```

1  package com.briup.homework;
2
3  import java.io.FileInputStream;
4  import java.io.FileOutputStream;
5  import java.io.IOException;
6  import java.io.ObjectInputStream;
7  import java.io.ObjectOutputStream;
8  import java.io.Serializable;
9
10 public class Test029_ObjectStream {
11
12     private final static String FILE_PATH = "D:/test/obj_file";
13
14     public static void save() {
15         // 补全代码
16     }
17
18     public static void load() {
19         // 补全代码
20     }
21
22     public static void main(String[] args) {
23         save();
24         load();
25     }
26 }
27
28 class Person implements Serializable {

```

```

29
30     private static final long serialVersionUID = 4258798319750620830L;
31     private String name;
32     private String gender;
33     public Person() {
34     }
35     public Person(String name, String gender) {
36         this.name = name;
37         this.gender = gender;
38     }
39     public String getName() {
40         return name;
41     }
42     public void setName(String name) {
43         this.name = name;
44     }
45     public String getGender() {
46         return gender;
47     }
48     public void setGender(String gender) {
49         this.gender = gender;
50     }
51     @Override
52     public String toString() {
53         return "Person [name=" + name + ", gender=" + gender + "]";
54     }
55 }

```

10.编程题

编写程序，实现将一个文件从GBK编码转换为UTF-8编码。

```

1  package com.briup.homework;
2
3  import java.io.File;
4  import java.io.FileInputStream;
5  import java.io.FileOutputStream;
6  import java.io.IOException;
7  import java.io.InputStreamReader;
8  import java.io.OutputStreamWriter;
9
10 public class Test0210_CharsetConversion {
11
12     public static void main(String[] args) {
13         // 源文件编码格式
14         String srcEncoding = "GBK";
15         // 要转换的编码格式
16         String distEncoding = "UTF-8";
17         // 目录路径

```

```
18     String dirPath = "D:/test";
19     // 源文件名
20     String fileName = "source.txt";
21     // 创建目录对象
22     File dir = new File(dirPath);
23     // 创建源文件对象
24     File source = new File(dir, fileName);
25     // 补全代码，实现功能
26     // ...
27 }
28 }
```