

Java 程序设计 LAB09

实验目的

- 理解并掌握控制台读入数据、标准输入输出、Scanner类（基础的输入输出）
- 理解并掌握Java文件管理（File类）、输入/输出流类及其衍生类的使用（字节流的顺序读写）
- 理解并掌握读写器及其衍生类的使用（字符流的顺序读写）
- 理解并掌握对象序列化（对象流的读写）

实验题目

1. 如果准备按字节读取一个文件的内容，应当使用 `FileInputStream` 流还是 `FileReader` 流，为什么？ 简答

2. 程序输出简答

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileInputStreamTest {
    public static void main(String[] args) {
        File f =new File("hello.txt");
        byte [] a="abcd".getBytes();
        try{
            FileOutputStream out=new FileOutputStream(f);
            out.write(a);
            out.close();
            FileInputStream in=new FileInputStream(f);
            byte [] tom= new byte[3];
            //Part I
            int m = in.read(tom,0,3);
            System.out.println(m);//3
            String s=new String(tom,0,3);
            System.out.println(s);//abc
            //Part II
            m = in.read(tom,0,3);
            System.out.println(m);//1
            s=new String(tom,0,3);
            System.out.println(s);//dbc
        }
        catch(IOException e) {}
    }
}
```

- (1) 请写出程序的输出
- (2) 解释 `Part I` 和 `Part II` 的输出为什么不同

3. 参考 `System.out` 的定义补全代码，这样你应该能理解 `System.out` 是什么了 编程

```
class MyPrintStream{
    //执行实际输出时可直接调用System.out.println();
    //Your code
}

class MySystem{
    //Your code
}

public class SystemOutDemo {
    public static void main(String[] args){
        MySystem.out.println("hello");
    }
}
```

4. 设计一个方法，用于移除文件中的注释 编程

```
public static void removeComments(String input, String output) throws
IOException;
```

注意：

1. 如果注释符号//不在该行的开头，或者是`/**`风格的注释，不用处理
2. 如果你实在想处理的话参考一下《编译原理》词法分析的部分

input.txt

```
File f = new File("D:\\java");
System.out.println("当前文件是: " +f);//这条注释不用处理
//文件是否存在
System.out.println("判断是否存在: "+f.exists());
//是否是文件夹
System.out.println("判断是否是文件夹: "+f.isDirectory());
//这个也不需要处理，因为该行的开头是空格
```

out.txt

```
File f = new File("D:\\java");
System.out.println("当前文件是: " +f); //这条注释不用处理
System.out.println("判断是否存在: "+f.exists());
System.out.println("判断是否是文件夹: "+f.isDirectory());
//这个也不需要处理, 因为该行的开头是空格
```

5. 设计一个方法, 使用 Java 的输入、输出流将一个文本文件的内容按行读出, 每读出一行就顺序添加行号, 并写入到另一个文件中编程

```
public static void addLineNo(String inputpath,String outputpath) throws
IOException;
```

input.txt

```
File f = new File("D:\\java");
System.out.println("当前文件是: " +f); //这条注释不用处理
//文件是否存在
System.out.println("判断是否存在: "+f.exists());
//是否是文件夹
System.out.println("判断是否是文件夹: "+f.isDirectory());
```

out.txt

```
1. File f = new File("D:\\java");
2. System.out.println("当前文件是: " +f); //这条注释不用处理
3. //文件是否存在
4. System.out.println("判断是否存在: "+f.exists());
5. //是否是文件夹
6. System.out.println("判断是否是文件夹: "+f.isDirectory());
```

6. 复制文件是常见的IO操作, 设计如下方法, 实现复制源文件 sourceFile 到目标文件 targetFile 编程

函数声明:

```
public static void copyFile (String sourceFile, String targetFile) throws
IOException;
```

7. 复制一个文件夹下面所有文件和子文件夹内容到另一文件夹 编程

```
public static void copyDirectory(String sourceDir, String targetDir) throws  
IOException;
```

注意:

文件复制(复用T6代码)、创建目录、递归