**实验四 输入输出处理**

## 一、实验目的

1. 了解File类及常用方法；

2. 掌握字节流的使用方法；

3. 掌握字符流的使用方法；

4. 掌握对象序列化方法。

## 二、实验内容

1. 使用Java语言提供的的输入、输出处理功能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。

2. 编程实现：使用带缓冲区的字节流拷贝一个文本文件。

3. 编程实现：使用带缓冲区的字符流拷贝一个文本文件。

**4. 利用Java输入/输出功能编程实现一个简易通讯录管理程序。每条信息至少包含：姓名、性别、手机号。程序能实现添加、修改、查找、删除等功能。**

5. 采用面向对象的设计思想，设计FileHelper类。FileHelper类中实现3个功能，依次是文件的打开操作、文件的保存操作、文件的拷贝操作，并进行测试。完善下列程序。

import java.io.\*;

public class FileHelper {

public static String openFile(String fileName){ //打开文件

StringBuffer sb=null;

FileInputStream fis=null;

try {

fis= \_\_\_\_\_\_\_\_\_\_\_\_\_ ; //实例化输入流对象

byte b[]=new byte[1024];

int len;

sb=new StringBuffer();

while( ( )!=-1 ){ //读文件并判断是否到达文件尾

String str=new String(b,0,len);

sb.append(str);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

try {

//如输入流非空则关闭输入流

} catch (IOException e) {

e.printStackTrace();

}

}

return sb.toString();

}

public static boolean saveFile(String fileName,String content){

boolean state=false;

FileOutputStream fos=null;

try {

fos=\_\_\_\_\_\_\_\_\_\_\_\_; //实例化输出流对象

//把content写入到文件中

state=true;

} catch (Exception e) {

e.printStackTrace();

}finally {

try {

//如输出流非空则关闭输出流

} catch (IOException e) {

e.printStackTrace();

}

}

return state;

}

public static boolean copyFile(String sourceFileName,String destinationFifleName){

boolean sate =false;

InputStream fis=null;

OutputStream fos=null;

try {

fis=new FileInputStream(sourceFileName);

fos=new FileOutputStream(destinationFifleName);

int len;

byte buffer[]=new byte[1024];

//此处请填写多行

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sate =true;

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

try {

if(fis!=null) fis.close();

if(fos!=null) fos.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return sate;

}

}

6. 请把第5题的FileHelper.java文件复制到当前工程文件夹中，阅读下列程序，写出运行结果，并思考：

（1）变量length的单位是什么，如何将其转换为KB；

单位为B (文件的大小/1024)+1

1. 如何将变量lastModifiedTime的结果转换为**yyyy/MM/dd HH：mm：SS**格式。

使用SimpleDataFormat 转换

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH：mm：SS");

1. 思考变量path和absolutePath的区别是什么。

Path是相对位置

absolutePath 是绝对位置

import java.io.File;

public class TestFile {

public static void main(String[] args) {

File file=new File("src\\FileHelper.java");

System.out.println(file.exists());

String name=file.getName();

String parent=file.getParent();

String absolutePath=file.getAbsolutePath();

String path=file.getPath();

long length =file.length();

long lastModifiedTime=file.lastModified();

System.out.println("Name:"+name);

System.out.println("Parent:"+parent);

System.out.println("AbsolutePath:"+absolutePath);

System.out.println("Path:"+path);

System.out.println("Length:"+length);

System.out.println("LastModifiedTime:"+lastModifiedTime);

}

}

7.（**选做**）利用面向对象的设计思想，创建学生类，学生信息包括：学号、姓名、出生日期。编写程序实现下列功能：

（1） 按表1所示格式，输入学生信息并写入到文件中，要求学生信息至少包含：学号、姓名、出生日期。

表1 学生信息

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 出生日期 |
| 201644001 | 张三 | 1997年2月3日 |
| 201644002 | 李四 | 1998年11月11日 |
| 201644003 | 王五 | 1996年3月2日 |
| 201644004 | 赵六 | 1996年12月12日 |
| 201644005 | 周正 | 1997年10月10日 |

（2） 写程序读出题（1）所得的文件，把读到的学生信息保存到List集合（ArrayList或者LinkedList）中，并输出List集合的内容到屏幕（格式参考表1）。

8. （**选做**）使用RandomAccessFile类将一个文本文件中的内容逆序输出。