**桂林电子科技大学 2019-2020学年 第1学期**

**《Java程序设计》实 验 报 告**

|  |
| --- |
| 指导教师意见：  教 师  签 名  成绩 |

实验名称  **实验六 输入输出流与文件操作**

计算机与信息安全 学院 **计算机类**  专 业

班 第 实 验 小 组

姓 名 奚锐 学 号  **1800300636**

实 验 日 期 2019 年 11 月 2 日

### 实验六 输入输出流与文件操作

### 实验目的

1、编写一个 Java 程序，能将硬盘上某个文件夹下的一个文件拷贝到另一个指定的文件夹中。注意， 测试时要能通过小文件和大文件测试，验证你的程序具有通用性，保证不管文件大小都要能拷贝成功。

2、假设已经有一个文本文件中存放着职工的工资记录。每个职工的记录包含姓名、工资、津贴三项。 每条记录可以存放于一行（三项之间的间隔符可以自己决定），也可以将每条记录的三项依次分别存放在 文本文件中的每一行。请设计一个程序，可以让用户选择打开这个文件查看其内容，并可以让用户选择 把每个职工的工资增加 10%之后又存回原来的文件。注意，职工工资记录的条数可能成百上千条甚至更多， 要能通过你所设计的程序，批量地一次性修改所有职工工资记录。

### 实验内容

1、编写一个 Java 程序，能将硬盘上某个文件夹下的一个文件拷贝到另一个指定的文件夹中。注意， 测试时要能通过小文件和大文件测试，验证你的程序具有通用性，保证不管文件大小都要能拷贝成功。

2、假设已经有一个文本文件中存放着职工的工资记录。每个职工的记录包含姓名、工资、津贴三项。 每条记录可以存放于一行(三项之间的间隔符可以自己决定)，也可以将每条记录的三项依次分别存放在 文本文件中的每一行。请设计一个程序，可以让用户选择打开这个文件查看其内容，并可以让用户选择 把每个职工的工资增加 10%之后又存回原来的文件。注意，职工工资记录的条数可能成百上千条甚至更多， 要能通过你所设计的程序，批量地一次性修改所有职工工资记录。

### 分析与设计

第一题设计一个文件拷贝的GUI，放置两个Text用于获得初始的文件位置与新文件的位置，再设置一个Button用于启动程序。文件拷贝时，创建一个FileinputStream的实例in和FileOutputStream的实例out,再定义一个字节数组buffer作为字节读取的缓冲区。在开始按钮按下时，in将文件内容读到buffer中，而out将buffer中的内容随机写到新的文件地址中去。文件读取完毕后关闭in和out.

第二题需要定义一个自定义类myrecord并实现Serializable接口，添加name,salary,allowance属性，在ToString方法中输出name+"|"+salary+"|"+allowance+"|".

在设计一个读取文件的UI界面，放置一个Text用于获取文件地址，一个Button1用于读取文件，一个Button2用于批量修改薪水并保存文件。Button1中添加一个鼠标监听器。在鼠标被点击时，用一个ObjectInputStream将文件中的内容读取出来并转换成实体类。并将转换出来的实体类放入一个ArrayList中。Button2被点击时，将所有ArrayList里的元素的salary属性增加10%，修改完后自动保存文件。

这是一个文件拷贝器

package helloWorld;//工程的包名

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class Filecopyer {

private JTextField textField1;

private JTextField textField2;

private JButton copybutton;

private JPanel mainpanel;

public static void main(String []args){

Filecopyer mycopyer=new Filecopyer();

JFrame frame=new JFrame();

frame.setContentPane(mycopyer.mainpanel);

frame.setSize(300,400);

frame.setVisible(true);

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

mycopyer.copybutton.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

String a=new String();

byte []buf=new byte[10];

try {

File source = new File(mycopyer.textField1.getText());

FileInputStream in = new FileInputStream(source);

FileOutputStream out =new FileOutputStream(mycopyer.textField2.getText());

int count=0;

while((count=in.read(buf))!=-1){

out.write(buf,0,count);

}

in.close();

out.close();

}

catch(IOException ee){

System.out.println("文件没找到");

}

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

});

}

}

这是一个批量修改薪水的文件读取器

package helloWorld;

import javax.swing.\*;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.io.\*;

import java.util.ArrayList;

import java.util.List;

class myrecord implements Serializable {

private String name;

private float salary;

private float allowance;

public myrecord(){

name="无名氏";

salary=0;

allowance=0;

}

public myrecord(String name,float salary,float allowance)

{

this.name=name;

this.allowance=allowance;

this.salary=salary;

}

public String toString(){

return name+"|"+salary+"|"+allowance+"|";

}

public float getAllowance() {

return allowance;

}

public float getSalary() {

return salary;

}

public String getName() {

return name;

}

public void setAllowance(float allowance) {

this.allowance = allowance;

}

public void setName(String name) {

this.name = name;

}

public void setSalary(float salary) {

this.salary = salary;

}

}

public class FIlereader {

private JButton OpenButton;

private JList list1;

private JPanel mainpanel;

private JTextField textField1;

private JButton addsalary;

private JButton button2;

public static void main(String []args){

JFrame frame =new JFrame();

FIlereader myreader=new FIlereader();

frame.setContentPane(myreader.mainpanel);

frame.setVisible(true);

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

frame.setSize(800,600);

try {

ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("salary.txt"));

myrecord a=new myrecord("王平",23000f,3000f);

myrecord b=new myrecord("奚锐",40000f,3100f);

myrecord c=new myrecord("王平",50000f,3200f);

out.writeObject(a);

out.writeObject(b);

out.writeObject(c);

out.flush();

out.close();

}

catch (IOException e) {

e.printStackTrace();

}

myreader.OpenButton.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

ArrayList<myrecord> newrecord =new ArrayList<myrecord>();

try {

ObjectInputStream in =new ObjectInputStream(new FileInputStream(myreader.textField1.getText()));

for(int i=1;i<=3;i++){

myrecord t=new myrecord();

t=(myrecord)in.readObject();

newrecord.add(t);

}

in.close();

} catch (IOException | ClassNotFoundException ex) {

ex.printStackTrace();

}

myreader.list1.setListData(newrecord.toArray());

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

});

myreader.addsalary.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

ArrayList<myrecord> newrecord =new ArrayList<myrecord>();

try {

ObjectInputStream in =new ObjectInputStream(new FileInputStream(myreader.textField1.getText()));

for(int i=1;i<=3;i++){

myrecord t=new myrecord();

t=(myrecord)in.readObject();

t.setSalary(t.getSalary()\*1.1f);//增加10%

newrecord.add(t);

}

in.close();

} catch (IOException | ClassNotFoundException ex) {

ex.printStackTrace();

}

myreader.list1.setListData(newrecord.toArray());

try {

ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(myreader.textField1.getText()));

for(int i=0;i<3;i++){

out.writeObject(newrecord.get(i));

}

out.flush();

out.close();

}

catch (IOException ea) {

ea.printStackTrace();

}

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

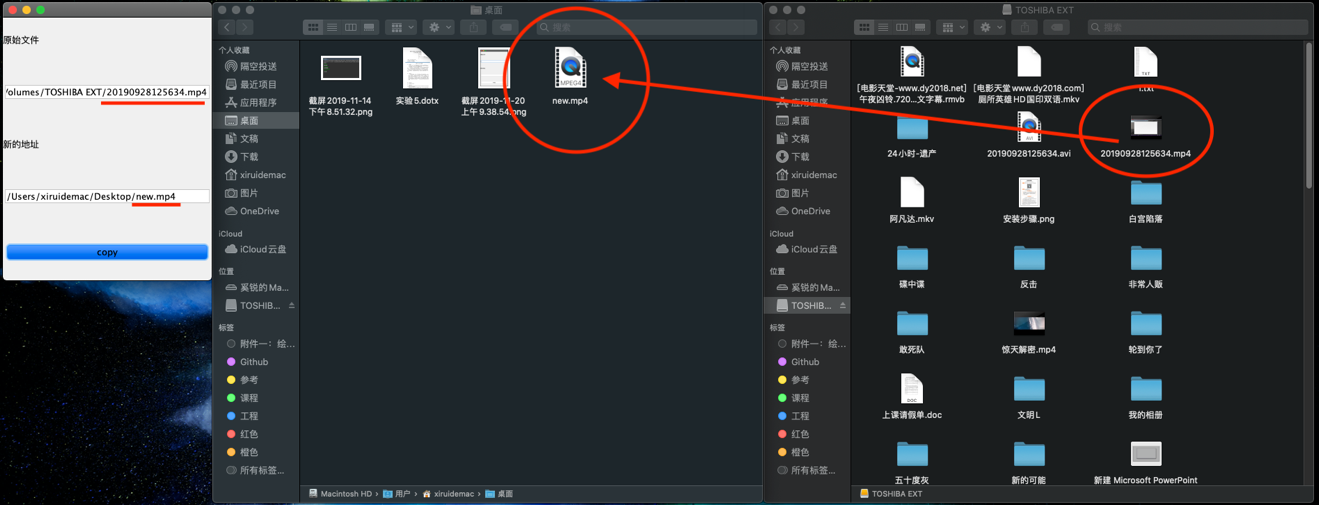
});

}

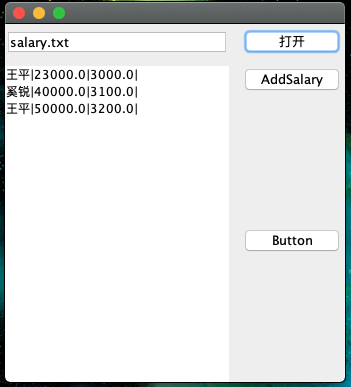
}

### 测试数据及运行结果

文件拷贝器：



文件字段批量修改器：



### 问题与总结

使用文件输入输出流进行文件拷贝操作时，可以将任意文件作为字节转存再写到新的地址中去，但当文件较大时，传输的速度过慢，应考虑使用缓存流来提高转存速度。

在进行文件内容批量修改字段时，需要为文件内存储的记录建立一个相应的类recod，该类需要实现Serializable接口，这样才可以将字段内容修改后转换成字符串再写入文件中，在读文件时，需要使用ObjectInputStream将文件内的字段转换成实体类。