**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  **实验七 多线程编程与网络编程**

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 12 日

### 实验七 多线程编程与网络编程

### 实验目的

1、 掌握线程和多线程的概念。

2、 掌握创建线程的两种方法及其区别。

3、 了解线程的启动、终止、同步、互斥和优先级等概念。

4、 掌握 TCP 和 UDP 网络程序设计的模型。

5、 掌握 TCP 和 UDP 程序设计的基本方法和所使用的系统类。

6、 了解基于应用层 http 协议的程序的设计以及使用系统类的组播、广播等网络程序设计

### 实验内容

1、 多线程编程题目 1：编写一个程序，其功能是运行之后，其中有一个线程可以输出 20 次你的学 号，另一个线程会输出 20 次你的姓名。

2、 多线程编程题目 2：编写一个图形界面程序，运行之后，让其中有一个线程能在界面上“实时” 显示系统当前时间（精确到秒获取时间可查询 java.util.Calendar 类，它包含了多个用于获得系统时间 的函数） 。另让一个线程可以在界面上提示当前系统时间下用户该做什么工作（例如当程序判断出系统时 间现在是 8：00 到 9：00，则提示用户该上课；现在是 23：00 到 8：00，则提示用户该休息。 具体测试 时可以将时间段限制到秒级，以便及时查看到程序运行中提示信息的变化） 。

3、 网络编程题目 1：先熟悉所给的示例，了解网络编程的一般基本概念。然后两个同学为一组，分 别设计一对控制台下的 TCP 通信程序的客户端和服务器端，然后双方要能够相互发送和反馈信息进行无 限次连续通信，直到其中一方发送表示结束通信的“end”字符串，然后接收方也返回一个“end”，双方 结束通信。然后改进程序，将自己的程序改进为具有图形界面的软件（即设计成一个简单的网络聊天程 序）。可在本地机上做测试，或两个同学为一组在不同的机器上作测试都可以，要将服务器地址和端口号 作为参数，在运行中可以进行修改。对于本机上测试，服务器地址使用 127.0.0.1 或者本机实际 IP 地址。 在网内不同机器上测试，要给出正确的服务器 IP 地址或名称。

4、 网络编程题目 2：观察和分析 QQ 等即时通讯软件的界面和功能，通过查阅资料，思考和分析其 设计思想和实现方法，设计一款自己的通讯软件，实现诸如消息传递、好友管理、系统设置、文件收发 等若干功能。

### 分析与设计

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) {

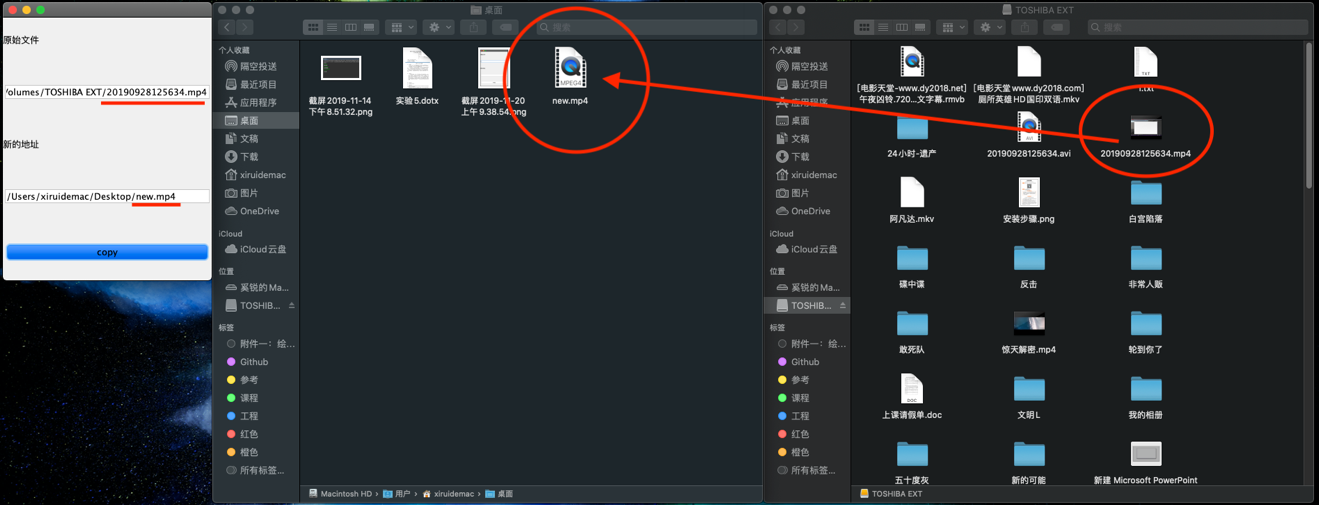
}

});

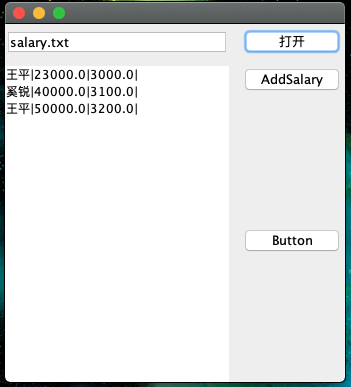
}

}

### 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。