异 常

常见异常：

（1）ArrayIndexoutofBoundsException

（2）NullPointerException

（3）FileNotFoundException

（4）NumberFormatException

（5）SqlException

（6）IOException

无论什么异常最后都会执行的语句，写在catch后面：Finally{}

1. 异常练习

**package** test1;

**import** java.awt.HeadlessException;

**import** javax.swing.JOptionPane;

**public** **class** Test2 {

**public** **int** sui(){

**try** {

String s1=JOptionPane.*showInputDialog*("请输入一个字符串");

**int** s2=Integer.*parseInt*(s1);

JOptionPane.*showMessageDialog*(**null**, "转换文整数为："+s2);

**return** s2;

} **catch** (HeadlessException e) {

JOptionPane.*showMessageDialog*(**null**, "00");

**return** 1;

} **catch** (NumberFormatException e) {

JOptionPane.*showMessageDialog*(**null**, "输入不能转换为数字");

**return** 2;

}**finally**{

**return** 3;

}

}

**public** **static** **void** main(String[] args) {

Test2 skl=**new** Test2();

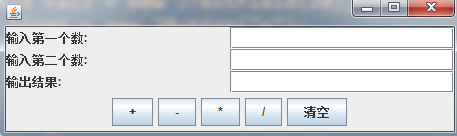
**int** a=skl.sui();

JOptionPane.*showMessageDialog*(**null**, a);

}

}

（2）加减乘除法，除法分母不能为0，有异常处理。



package com.chinasoft.exception;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class MyCalc extends JFrame implements ActionListener {

JTextField jtf1 = new JTextField(10);

JTextField jtf2 = new JTextField(10);

JTextField jtf3 = new JTextField(10);

JButton btn1 = new JButton("+");

JButton btn2 = new JButton("-");

JButton btn3 = new JButton("\*");

JButton btn4 = new JButton("/");

JPanel panel = new JPanel();

public MyCalc() {

btn1.addActionListener(this);

btn2.addActionListener(this);

btn3.addActionListener(this);

btn4.addActionListener(this);

panel.setLayout(new GridLayout(1, 4));

panel.add(btn1);panel.add(btn2);

panel.add(btn3);panel.add(btn4);

this.setLayout(new GridLayout(4, 1));

this.add(jtf1);this.add(panel);

this.add(jtf2);this.add(jtf3);

this.pack();this.setVisible(true);}

public void actionPerformed(ActionEvent arg0) {

String s1 = jtf1.getText().trim();

String s2 = jtf2.getText().trim();

if (s1.isEmpty()) {jtf3.setText("第一个数不能为空");

} else if (s2.isEmpty()) {jtf3.setText("第二个数不能为空");

} else {try { double d1 = Double.parseDouble(s1);

double d2 = Double.parseDouble(s2);

if (arg0.getSource() == btn1) {

jtf3.setText((d1 + d2) + "");}

if (arg0.getSource() == btn2) {

jtf3.setText((d1 - d2) + "");}

if (arg0.getSource() == btn3) {

jtf3.setText((d1 \* d2) + "");}

if (arg0.getSource() == btn4) {

if (d2 == 0) {

jtf3.setText("分母不能为零");} else {

jtf3.setText((d1 / d2) + "");}}

} catch (NumberFormatException e) {

jtf3.setText("请输入数字！"); } }

public static void main(String[] args) {

MyCalc mc = new MyCalc(); }}

Java流

输入流：相对于程序，数据写进来就是输入流。

输出流：相对于程序，数据输出去就是输出流。

字节流：一次处理8位，就是字节流。

字符流：一次处理16位，2个字节，就是字符流。

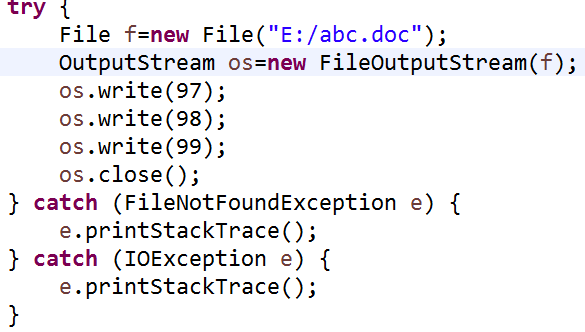
节点流：可以从/向一个特定的IO设备（如磁盘、网络）读/写数

据的流，称为节点流。

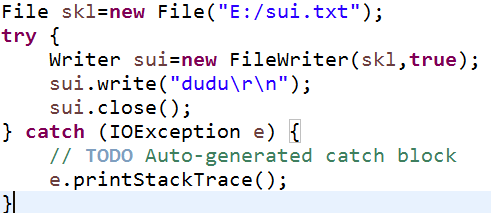
处理流：实现对一个已存在的流的连接和封装，通过所封装的流

的功能调用实现数据读/写功能的流，称为处理流。

父类创建一个子类的字节（节点）流，将97（a）写入到文件中，如没有文件，可以创建文件。

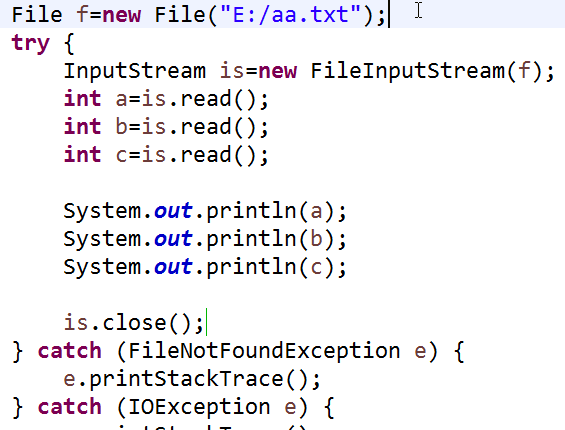


Writer可以写入字符，true表示写入不覆盖，而是在后面添加，\r\n表示回车换行。

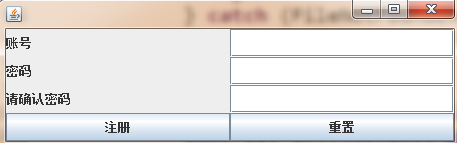


Sui.flush();写于sui.close（）；前面，用于刷新缓存区。

低级流，每一次只读一个字节。



1. 窗体登录注册页面，当注册成功时候把新注册的账号和密码添加到文件中。



**if** (arg0.getSource() == sui9) {

String s1 = sui6.getText();

String s2 = sui7.~~getText~~();

String s3 = sui8.~~getText~~();

**try** {

**boolean** ret = **false**;

Scanner input = **new** Scanner(**new** File("E:/suikailing.txt"));

**if** (s1.isEmpty()) {

JOptionPane.*showMessageDialog*(**null**, "账号不能为空");

} **else** **if** (s2.isEmpty() || s3.isEmpty()) JOptionPane.*showMessageDialog*(**null**, "密码不能为空");

} **else** **if** (!s2.equals(s3)) {

JOptionPane.*showMessageDialog*(**null**, "密码不一样");} **else** {**while** (input.hasNextLine()) { String s4 = input.nextLine();

**int** a = s4.indexOf("#");

String s5 = s4.substring(0, a);

**if** (s5.equals(s1)) {ret = **true**;

JOptionPane.*showMessageDialog*(**null**, "账号已经被注册,请重新输入");}}**if** (ret == **false**) {

**try** {**this**.setVisible(**false**); JOptionPane.*showMessageDialog*(**null**, "注册成功");File ma=**new** File("E:/suikailing.txt");

String eee=sui6.getText()+"#"+sui7.~~getText~~();

Writer qwe=**new** FileWriter(ma,**true**);

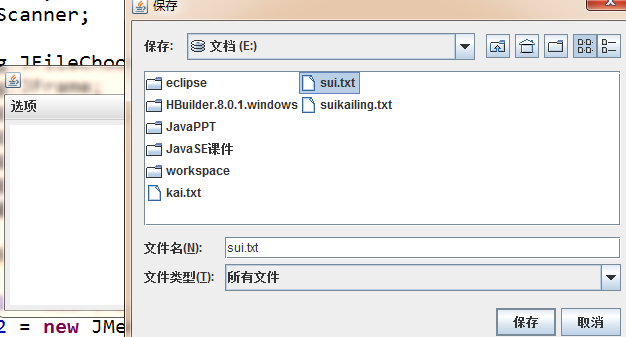
qwe.write(eee+"\r\n");

qwe.close();} **catch** (HeadlessException e) { e.printStackTrace();

} **catch** (IOException e) {e.printStackTrace();}}}

} **catch** (FileNotFoundException e) {

e.printStackTrace(); } }

（3）将显示在文本中的内容保存到文件中。原格式保存

**package test1;**

**import java.awt.BorderLayout;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.io.File;**

**import java.io.FileNotFoundException;**

**import java.io.FileWriter;**

**import java.io.IOException;**

**import java.io.Writer;**

**import java.util.Scanner;**

**import javax.swing.JFileChooser;**

**import javax.swing.JFrame;**

**import javax.swing.JMenu;**

**import javax.swing.JMenuBar;**

**import javax.swing.JMenuItem;**

**import javax.swing.JPanel;**

**import javax.swing.JTextArea;**

**public class Test6 extends JFrame implements ActionListener {**

**JMenuItem sui1 = new JMenuItem("打开");**

**JMenuItem sui2 = new JMenuItem("保存");**

**JMenu kai1 = new JMenu("选项");**

**JPanel jp1 = new JPanel();**

**JTextArea jta = new JTextArea(10, 20);**

**public Test6() {**

**JMenuBar bar = new JMenuBar();**

**kai1.add(sui1);**

**kai1.add(sui2);**

**bar.add(kai1);**

**this.setJMenuBar(bar);**

**sui1.addActionListener(this);**

**sui2.addActionListener(this);**

**jp1.add(jta);**

**add(jp1, BorderLayout.NORTH);**

**pack();**

**setVisible(true);**

**}**

**@Override**

**public void actionPerformed(ActionEvent arg0) {**

**if (arg0.getSource() == sui1) {**

**JFileChooser skl = new JFileChooser();**

**int aaa = skl.showOpenDialog(null);**

**StringBuffer dudu = new StringBuffer();**

**if (aaa == 0) {**

**try {**

**Scanner input = new Scanner(skl.getSelectedFile());**

**while (input.hasNextLine()) {**

**dudu.append(input.nextLine() + "\n");**

**jta.setText(dudu.toString());**

**}**

**} catch (FileNotFoundException e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**if (arg0.getSource() == sui2) {**

**JFileChooser skl = new JFileChooser();**

**int aaa = skl.showSaveDialog(null);**

**if (aaa == 0) {**

**try {**

**Writer nn = new FileWriter(skl.getSelectedFile(), true);**

**nn.write(jta.getText().replace("\n", "\r\n"));**

**nn.flush();**

**nn.close();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**}**

**}**

**}**

**public static void main(String[] args) {**

**Test6 skl = new Test6();**

**}**

**}**

**（4）低级流读取，每次只能读取文件中的一个字节。练习：读取文件中的内容，显示出来。并且把读取出来的内容写到另外一个文件中，实现文件复制。**

**package** test1;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.Reader;

**import** java.io.Writer;

**public** **class** Test7 {

**public** **static** **void** main(String[] args) {

File skl=**new** File("E:/sui.txt");

File skl1=**new** File("E:/skling.txt");

**try** {

//InputStream dudu=new FileInputStream(skl);

Reader dudu=**new** FileReader(skl);

//outputStream du=new FileoutputStream(skl1);

Writer yaya=**new** FileWriter(skl1,**true**);

**for**(**int** i=0;i<skl.length();i++){

**int** a=dudu.read();

**if**(a!=-1){

yaya.write((**char**)a);

System.***out***.print((**char**)a);

} }

yaya.flush();

yaya.close();

dudu.close();

} **catch** (FileNotFoundException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

} }

}

（5）实现对文件的加密解密，用文件选择器做。必须用字节流类型，不能用reader和writer写，因为图片是字节传输。

**package** test1;

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.io.Reader;

**import** java.io.Writer;

**import** javax.swing.JButton;

**import** javax.swing.JFileChooser;

**import** javax.swing.JFrame;

**public** **class** Test8 **extends** JFrame **implements** ActionListener{

JButton jb1=**new** JButton("加密");

JButton jb2=**new** JButton("解密");

**public** Test8() {

**this**.add(jb1,BorderLayout.***NORTH***);

**this**.add(jb2,BorderLayout.***SOUTH***);

jb1.addActionListener(**this**);

jb2.addActionListener(**this**);

pack();

setVisible(**true**);

}

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

JFileChooser sui1=**new** JFileChooser();

JFileChooser sui2=**new** JFileChooser();

**if**(arg0.getSource()==jb1){

**int** kai1=sui1.showOpenDialog(**null**);

**int** kai2=sui2.showSaveDialog(**null**);

**try** {

InputStream suikailing1 = **new** FileInputStream(sui1.getSelectedFile());

OutputStream suikailing2 = **new** FileOutputStream(sui2.getSelectedFile());

**if**(kai1==0){

**for**(**int** i=0;i<sui1.getSelectedFile().length();i++){

**if**(kai2==0){

suikailing2.write(suikailing1.read()+2);

}

}

}

suikailing2.flush();

suikailing2.close();

suikailing1.close();

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**if**(arg0.getSource()==jb2){

**try** {

**int** kai1=sui1.showOpenDialog(**null**);

**int** kai2=sui2.showSaveDialog(**null**);

InputStream suikailing1 = **new** FileInputStream(sui1.getSelectedFile());

OutputStream suikailing2 = **new** FileOutputStream(sui2.getSelectedFile());

**if**(kai1==0){

**for**(**int** i=0;i<sui1.getSelectedFile().length();i++){

**if**(kai2==0){

suikailing2.write(suikailing1.read()-2);

}

}

}

suikailing1.close();

suikailing2.flush();

suikailing2.close();

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**public** **static** **void** main(String[] args) {

Test8 skl=**new** Test8();

}

}

（6）实现大文件的复制suikailing2.write(buffer,0,a);

用inputstream和reader都可以做，reader定义的是char类型的数组。Inputstream定义的是byte类型的数组。

**package** test1;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.io.Reader;

**public** **class** Test10 {

**public** **static** **void** main(String[] args) {

File skl1 = **new** File("E:/sui.avi");

File skl2 = **new** File("E:/sui2.avi");

**try** {InputStream suikailing1 = **new** FileInputStream(skl1);

OutputStream suikailing2 = **new** FileOutputStream(skl2);

Reader skl = **new** FileReader(skl1);

**char** uffer[] = **new** **char**[1000 \* 1024];

skl.read(uffer);

**byte** buffer[] = **new** **byte**[1000 \* 1024];

**int** a = 0;

**while** ((a = skl.read(uffer)) != -1) { suikailing2.write(buffer, 0, a);

}

} **catch** (FileNotFoundException e) { } **catch** (IOException e) {

}

}

}

节点流：

C处理流封装节点流

Dateinputstream dis=new Dateinputstream(is);

Bufferedinputstream bis=new Bufferedinputstream(is);

Dis.readUTF(); 用于网络传输数据。

Outputstream os=new fileoutputstream(文件，true);

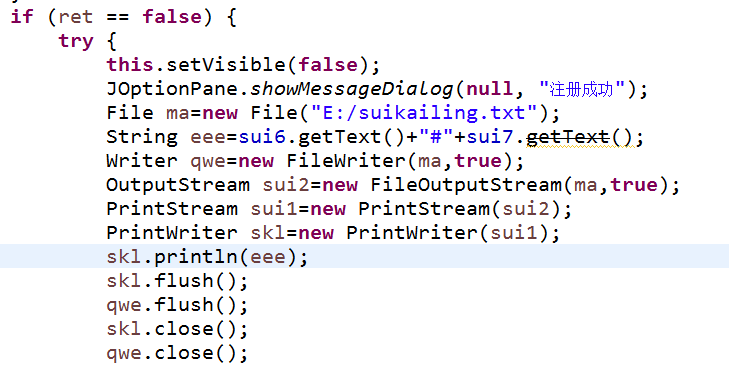
Printstream ps=new printstream(os);

Printwriter pw=new printwriter(ps);

Printwriter可以传字节的对象ps，也可以传writer的对象。

Pw.println();写入时候自动换行。

下面图片是一个简易的写入例子，在字节流上面套2层管子。



字节流转换位字符流：inputstreamreader

Inputstream sui0=new fileinputstream(new filr(“E:/suikailing.txt”));

Reader sui1=new inputstreamreader(sui0);

和reader一样，可以吧字节输出转换位字符输出：

Writer sui3=new outputstreamwriter();

高级处理流(带缓冲)，封装reader低级流，

Bufferedreader sui2=new bufferreader(sui1);

Sui2.readline();每次读取一行

缓冲流：

Bufferinputstream

Bufferoutputstream

Bufferreader

Bufferwriter

（7）从一个文件读取内容输出到另一个文件中，用到BufferedReader，InputStreamReader，BufferedWriter。

**package** test1;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.InputStreamReader;

**import** java.io.OutputStream;

**import** java.io.OutputStreamWriter;

**import** java.io.Reader;

**import** java.io.Writer;

**public** **class** Test11 {

**public** **static** **void** main(String[] args) {

**try** {

InputStream sui0=**new** FileInputStream("E:/.txt");

Reader sui1 = **new** InputStreamReader(sui0);

BufferedReader sui2=**new** BufferedReader(sui1);

OutputStream sui5=**new**

FileOutputStream("E:/suikailing222.txt",**true**);

Writer sui3=**new** OutputStreamWriter(sui5);

BufferedWriter sui4=**new** BufferedWriter(sui3);

String s="";

**while**((s=sui2.readLine())!=**null**){

System.***out***.println(s);

sui4.write(s+"\r\n");

}

sui2.close();

sui1.close();

sui0.close();

sui4.flush();

sui3.flush();

sui5.flush();

sui4.close();

sui3.close();

sui5.close();

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

网 络 编 程

服务器端：

打开服务器

ServerSocket sever = **new** ServerSocket(端口);

接受客户端信息Socket sui1=sever.accept();

设置输入流接收客户端传来的信息：

InputStream sui2=sui1.getInputStream();

将传过来的数据复制给一个字符串并输出：

String sui4=sui3.readUTF();

System.***out***.println(sui4);

客户端：

连接服务器Socket s=**new** Socket(IP地址,端口);

给客户端发送信息，得到输出的信息

OutputStream s2=s.getOutputStream();

将要传给服务器的信息，写给服务器

sui3.writeUTF(“要写入的信息”);

字符串位null值的判断：if（字符串==null）

（8）网络编程：实现客户端和服务器的无限畅聊。

//服务器

**package** test2;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**import** javax.swing.JOptionPane;

**public** **class** Test1 {

**public** **static** **void** main(String[] args) {

ServerSocket sever;

**try** {sever = **new** ServerSocket(1234);

System.***out***.println("服务器已经启动");

**while** (**true**) {

Socket sui1 = sever.accept();

InputStream is = sui1.getInputStream();

DataInputStream dis = **new** DataInputStream(is);

String s2 = dis.readUTF();

String s5 = JOptionPane.*showInputDialog*(s2);

OutputStream os = sui1.getOutputStream();

DataOutputStream dos = **new** DataOutputStream(os);

dos.writeUTF(s5);

**if** (s5.equals("拜拜") || s2.equals("拜拜")) {

dis.close();

is.close();

dos.flush();

os.flush();

dos.close();

os.close();

sever.close();

**break**; } }

} **catch** (IOException e) {

e.printStackTrace();

} } }

//客户端

**package** test2;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.net.Socket;

**import** java.util.Scanner;

**import** javax.swing.JOptionPane;

**public** **class** Test2 {

**public** **static** **void** main(String[] args) {

**try** {**while** (**true**) {

Socket sui1 = **new** Socket("192.168.22.11", 2011);

OutputStream os = sui1.getOutputStream();

DataOutputStream dos = **new** DataOutputStream(os);

String ss = JOptionPane.*showInputDialog*("刘迪");

dos.writeUTF(ss);

InputStream is = sui1.getInputStream();

DataInputStream dis = **new** DataInputStream(is);

String sss = dis.readUTF(); JOptionPane.*showMessageDialog*(**null**, sss);

**if** (ss.equals("拜拜") || sss.equals("拜拜")) {

dis.close();

is.close();

dos.flush();

os.flush();

dos.close();

os.close();

sui1.close();

**break**;} }

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

**（9）**实现客户端和服务器的传输文件。文件选择器，并且可以传送文件名字。

**package** test2;

**import** java.io.DataInput;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**import** javax.swing.JFileChooser;

**import** javax.swing.JOptionPane;

**public** **class** Test5 {

**public** **static** **void** main(String[] args) {

**try** {

ServerSocket skl=**new** ServerSocket(5421);

System.***out***.println("服务器启动");

JFileChooser jfc=**new** JFileChooser();

**byte** buffer[]=**new** **byte**[100\*1024];

**while**(**true**){

String s=JOptionPane.*showInputDialog*("上传下载或退出");

Socket socket=skl.accept();

**if**(s.equals("上传")){

**int** a=jfc.showOpenDialog(**null**);

**if**(a==0){

InputStream is=**new** FileInputStream(jfc.getSelectedFile());

String ss=jfc.getSelectedFile().getName();

DataInputStream dis=**new** DataInputStream(is);

OutputStream os=socket.getOutputStream();

DataOutputStream dos=**new** DataOutputStream(os);

dos.writeUTF(ss);

**int** ab=0; **while**((ab=dis.read(buffer))!=-1){

dos.write(buffer, 0, ab);

}

}

}**else** **if**(s.equals("下载")){

InputStream is=socket.getInputStream();

DataInputStream dis=**new** DataInputStream(is); JOptionPane.*showMessageDialog*(**null**, dis.readUTF());

**int** a=jfc.showSaveDialog(**null**);

**if**(a==0){

OutputStream os=**new** FileOutputStream(jfc.getSelectedFile());

DataOutputStream dos=**new** DataOutputStream(os);

**int** ab=0;

**while**((ab=dis.read(buffer))!=-1){

dos.write(buffer, 0, ab);

}

}

}**else**{

System.*exit*(0);

}

}

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

//客户端

**package** test2;

**import** java.io.DataInput;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.net.Socket;

**public** **class** Test3 {

**public** **static** **void** main(String[] args) {

**try** {

Socket skl = **new** Socket("192.168.22.44", 5678);

File f=**new** File("E:/sui.avi");

OutputStream os=skl.getOutputStream();

DataOutputStream dos=**new** DataOutputStream(os);

InputStream is=**new** FileInputStream(f);

DataInputStream dis=**new** DataInputStream(is);

**byte** buffer[] = **new** **byte**[1000 \* 1024];

**int** a=0;

**while**((a=dis.read(buffer))!=-1){

dos.write(buffer, 0, a);

}

} **catch** (IOException e) {

e.printStackTrace(); }} }