用戶端程式一對一ex322

ClientOneToOne\_ServerFrame

package com.zzk;

import java.awt.BorderLayout;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Hashtable;

import java.util.Iterator;

import java.util.Set;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class ClientOneToOne\_ServerFrame extends JFrame {

private JTextArea ta\_info;

private ServerSocket server; // 宣告ServerSocket對像

private Socket socket; // 宣告Socket對像socket

private Hashtable<String, Socket> map = new Hashtable<String, Socket>();// 用於儲存連接到服務器的使用者和客戶端套接字對像

public void createSocket() {

try {

server = new ServerSocket(1978);

while (true) {

ta\_info.append("等待新客戶連接......\n");

socket = server.accept();// 建立套接字對像

ta\_info.append("客戶端連接成功。" + socket + "\n");

new ServerThread(socket).start();// 建立並啟動線程對像

}

} catch (IOException e) {

e.printStackTrace();

}

}

class ServerThread extends Thread {

Socket socket;

public ServerThread(Socket socket) {

this.socket = socket;

}

public void run() {

try {

BufferedReader in = new BufferedReader(new InputStreamReader(

socket.getInputStream()));// 建立輸入流對像

while (true) {

String info = in.readLine();// 讀取資訊

String key = "";

if (info.startsWith("使用者：")) {// 增加登入使用者到客戶端列表

key = info.substring(3, info.length());// 獲得使用者名稱並作為鍵使用

map.put(key, socket);// 增加鍵值對

Set<String> set = map.keySet();// 獲得集合中所有鍵的Set檢視

Iterator<String> keyIt = set.iterator();// 獲得所有鍵的迭代器

while (keyIt.hasNext()) {

String receiveKey = keyIt.next();// 獲得表示接收資訊的鍵

Socket s = map.get(receiveKey);// 獲得與該鍵對應的套接字對像

PrintWriter out = new PrintWriter(s

.getOutputStream(), true);// 建立輸出流對像

Iterator<String> keyIt1 = set.iterator();// 獲得所有鍵的迭代器

while (keyIt1.hasNext()) {

String receiveKey1 = keyIt1.next();// 獲得鍵，用於向客戶端增加使用者列表

out.println(receiveKey1);// 發送資訊

out.flush();// 更新輸出緩衝區

}

}

} else {// 轉發接收的訊息

key = info.substring(info.indexOf("：發送給：") + 5, info

.indexOf("：的資訊是："));// 獲得接收方的key值,即接收方的使用者名稱

String sendUser = info.substring(0, info

.indexOf("：發送給："));// 獲得發送方的key值,即發送方的使用者名稱

Set<String> set = map.keySet();// 獲得集合中所有鍵的Set檢視

Iterator<String> keyIt = set.iterator();// 獲得所有鍵的迭代器

while (keyIt.hasNext()) {

String receiveKey = keyIt.next();// 獲得表示接收資訊的鍵

if (key.equals(receiveKey)

&& !sendUser.equals(receiveKey)) {// 如果是發送方，但不是使用者本身

Socket s = map.get(receiveKey);// 獲得與該鍵對應的套接字對像

PrintWriter out = new PrintWriter(s

.getOutputStream(), true);// 建立輸出流對像

out.println("MSG:"+info);// 發送資訊

out.flush();// 更新輸出緩衝區

}

}

}

}

} catch (IOException e) {

ta\_info.append(socket + "已經退出。\n");

}

}

}

/\*\*

\* Launch the application

\*

\* @param args

\*/

public static void main(String args[]) {

ClientOneToOne\_ServerFrame frame = new ClientOneToOne\_ServerFrame();

frame.setVisible(true);

frame.createSocket();

}

/\*\*

\* Create the frame

\*/

public ClientOneToOne\_ServerFrame() {

super();

setTitle("客戶端一對一通訊——服務器端程式");

setBounds(100, 100, 385, 266);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

final JScrollPane scrollPane = new JScrollPane();

getContentPane().add(scrollPane, BorderLayout.CENTER);

ta\_info = new JTextArea();

scrollPane.setViewportView(ta\_info);

}

}

ClientOneToOne\_ClientFrame

package com.zzk;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.EventQueue;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Socket;

import java.net.UnknownHostException;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JList;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JSplitPane;

import javax.swing.JTextArea;

import javax.swing.JTextField;

public class ClientOneToOne\_ClientFrame extends JFrame {

private JTextField tf\_newUser;

private JList user\_list;

private JTextArea ta\_info;

private JTextField tf\_send;

PrintWriter out;// 宣告輸出流對像

private boolean loginFlag = false;// 為true時表示已經登入，為false時表示未登入

/\*\*

\* Launch the application

\*

\* @param args

\*/

public static void main(String args[]) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ClientOneToOne\_ClientFrame frame = new ClientOneToOne\_ClientFrame();

frame.setVisible(true);

frame.createClientSocket();// 呼叫方法建立套接字對像

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public void createClientSocket() {

try {

Socket socket = new Socket("localhost", 1978);// 建立套接字對像

out = new PrintWriter(socket.getOutputStream(), true);// 建立輸出流對像

new ClientThread(socket).start();// 建立並啟動線程對像

} catch (UnknownHostException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

class ClientThread extends Thread {

Socket socket;

public ClientThread(Socket socket) {

this.socket = socket;

}

public void run() {

try {

BufferedReader in = new BufferedReader(new InputStreamReader(

socket.getInputStream()));// 建立輸入流對像

DefaultComboBoxModel model = (DefaultComboBoxModel) user\_list

.getModel();// 獲得列表框的模型

while (true) {

String info = in.readLine().trim();// 讀取資訊

if (!info.startsWith("MSG:")) {

boolean itemFlag = false;// 標記是否為列表框增加列記錄，為true不增加，為false增加

for (int i = 0; i < model.getSize(); i++) {

if (info.equals((String) model.getElementAt(i))) {

itemFlag = true;

}

}

if (!itemFlag) {

model.addElement(info);// 增加列記錄

} else {

itemFlag = false;

}

} else {

ta\_info.append(info + "\n");// 在純文字域中顯示資訊

if (info.equals("88")) {

break;// 結束線程

}

}

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

private void send() {

if (!loginFlag) {

JOptionPane.showMessageDialog(null, "請先登入。");

return;

}

String sendUserName = tf\_newUser.getText().trim();

String info = tf\_send.getText();// 獲得輸入的資訊

if (info.equals("")) {

return;// 如果沒輸入資訊則傳回，即不發送

}

String receiveUserName = (String) user\_list.getSelectedValue();// 獲得接收資訊的使用者

String msg = sendUserName + "：發送給：" + receiveUserName + "：的資訊是： "

+ info;// 定義發送的資訊

if (info.equals("88")) {

System.exit(0);// 如果沒輸入資訊是88，則退出

}

out.println(msg);// 發送資訊

out.flush();// 更新輸出緩衝區

tf\_send.setText(null);// 清空純文字框

}

/\*\*

\* Create the frame

\*/

public ClientOneToOne\_ClientFrame() {

super();

setTitle("客戶端一對一通訊——客戶端程式");

setBounds(100, 100, 385, 288);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

final JPanel panel = new JPanel();

getContentPane().add(panel, BorderLayout.SOUTH);

final JLabel label = new JLabel();

label.setText("輸入聊天內容：");

panel.add(label);

tf\_send = new JTextField();

tf\_send.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

send();// 呼叫方法發送資訊

}

});

tf\_send.setPreferredSize(new Dimension(180, 25));

panel.add(tf\_send);

final JButton button = new JButton();

button.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

send();// 呼叫方法發送資訊

}

});

button.setText("發 送");

panel.add(button);

final JSplitPane splitPane = new JSplitPane();

splitPane.setDividerLocation(100);

getContentPane().add(splitPane, BorderLayout.CENTER);

final JScrollPane scrollPane = new JScrollPane();

splitPane.setRightComponent(scrollPane);

ta\_info = new JTextArea();

scrollPane.setViewportView(ta\_info);

final JScrollPane scrollPane\_1 = new JScrollPane();

splitPane.setLeftComponent(scrollPane\_1);

user\_list = new JList();

user\_list.setModel(new DefaultComboBoxModel(new String[] { "" }));

scrollPane\_1.setViewportView(user\_list);

final JPanel panel\_1 = new JPanel();

getContentPane().add(panel\_1, BorderLayout.NORTH);

final JLabel label\_1 = new JLabel();

label\_1.setText("輸入使用者名稱稱：");

panel\_1.add(label\_1);

tf\_newUser = new JTextField();

tf\_newUser.setPreferredSize(new Dimension(180, 25));

panel\_1.add(tf\_newUser);

final JButton button\_1 = new JButton();

button\_1.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

if (loginFlag) {

JOptionPane.showMessageDialog(null, "在同一視窗只能登入一次。");

return;

}

String userName = tf\_newUser.getText().trim();// 獲得登入使用者名稱

out.println("使用者：" + userName);// 發送登入使用者的名稱

out.flush();// 更新輸出緩衝區

tf\_newUser.setEnabled(false);

loginFlag = true;

}

});

button\_1.setText("登 錄");

panel\_1.add(button\_1);

}

}