**作业3**

**题目3（网络编程）：**

简单网络通信程序实现。要求使用流式Socket实现客户端发送消息到服务器，服务器返回收到消息的确认。

姓名： 班级： 学号：

说明：请复制代码和程序运行结果。

1）代码应该有注释

2）复制代码时，在eclipse中将代码字体设置为20，确保代码复制到文档中清晰。

3）复制代码时，应表标明代码对应的java源代码文件和package

代码复制在横线以下：

---------------------------------------------------------

目录结构

.

├─.idea

├─out

│ └─production

│ └─JavaSocket

│ └─com

│ ├─Client

│ ├─company

│ └─Server

└─src

└─com

├─Client

├─company

└─Server

Client.java

package com.Client;

import java.io.\*;

import java.net.Socket;

import java.util.Scanner;

public class Client {

public static void main(String[] args) {

String serverName = "localhost";

int port = Integer.parseInt("8081");

Scanner Sc = new Scanner(System.in);

System.out.println("请输入用户名：");

String name = Sc.nextLine();

try

{

System.out.println("连接到主机：" + serverName + " ，端口号：" + port);

Socket client = new Socket(serverName, port);

System.out.println("远程主机地址：" + client.getRemoteSocketAddress());

OutputStream outToServer = client.getOutputStream();

DataOutputStream out = new DataOutputStream(outToServer);

out.writeUTF(name + "接入服务");

InputStream inFromServer = client.getInputStream();

DataInputStream in = new DataInputStream(inFromServer);

System.out.println("cur");

System.out.println(in.readUTF());

for (;;) {

String message = Sc.nextLine();

out.writeUTF(name + ": " + message);

if (message.equals("fuck")) {

System.out.println("break");

break;

}

System.out.println(in.readUTF());

}

// client.close();

}catch(IOException e)

{

e.printStackTrace();

}

}

}

Server.java

package com.Server;

import java.io.IOException;

public class Server {

public static void main(String[] args) {

int port = Integer.parseInt("8081");

try

{

Thread t = new GreetingServer(port);

t.run();

}catch(IOException e)

{

e.printStackTrace();

}

}

}

GreetingServer.java

package com.Server;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.net.SocketTimeoutException;

public class GreetingServer extends Thread {

private ServerSocket serverSocket;

public GreetingServer(int port) throws IOException

{

serverSocket = new ServerSocket(port);

serverSocket.setSoTimeout(100000);

}

public void run() {

while (true) {

try {

System.out.println("等待远程连接，端口号为：" + serverSocket.getLocalPort() + "...");

Socket server = serverSocket.accept();

System.out.println("远程主机地址：" + server.getRemoteSocketAddress());

DataInputStream in = new DataInputStream(server.getInputStream());

DataOutputStream out = new DataOutputStream(server.getOutputStream());

out.writeUTF(in.readUTF());

for (;;) {

String res = in.readUTF();

System.out.println(res);

// 对接收过来的String进行解析

String resInfo = res.split(":")[1].trim();

if (resInfo.equals("fuck")) {

System.out.println("break");

break;

}

out.writeUTF(res);

}

// out.flush();

// server.close();

} catch (SocketTimeoutException s) {

System.out.println("Socket timed out!");

break;

} catch (IOException e) {

e.printStackTrace();

break;

}

}

}

}

运行结果：

文本

描述已自动生成

文本

描述已自动生成