

```

package com.sxt.chat02;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

/**
 * 在线聊天室：服务器
 * 目标：使用多线程实现多个客户可以正常收发多条信息
 * 问题：其他客户必须等待之前的客户退出，才能继续 排队
 * @author 江
 *
 */

public class TMultiChat {
    public static void main(String[] args) throws IOException {
        System.err.println("--server--");
        //创建服务器端
        ServerSocket server=new ServerSocket(8888);
        while(true) {
            //阻塞式连接
            Socket client=server.accept();
            System.err.println("一个客户端建立连接");

            new Thread(()->{
                DataInputStream dis=null;
                DataOutputStream dos=null;
                try {
                    dis = new
DataInputStream(client.getInputStream());
                    dos=new
DataOutputStream(client.getOutputStream());
                } catch (IOException e1) {

```

```

        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

    boolean isRunning=true;
    while(isRunning){
        //3. 接受消息
        String msg;
        try {
            msg = dis.readUTF();
            //返回消息
            dos.writeUTF(msg);
            dos.flush();
        } catch (IOException e) {
            e.printStackTrace();
            isRunning=false;//停止线程
        }

    }

    try {
        if(null!=dos) {
            //释放资源
            dos.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        if(null!=client) {
            //释放资源
            client.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

```

```
}
```

```
try {
```

```
    if(null!=dis) {
```

```
        //释放资源
```

```
        dis.close();
```

```
    }
```

```
} catch (IOException e) {
```

```
    e.printStackTrace();
```

```
}
```

```
}).start();
```

```
}
```

```
}
```

```
}
```