

JAD Project 3 : Multi-Client Chatting Server

1. 项目简介

实现一个多客户端的纯文本聊天服务器，能同时接受多个客户端的连接，并将任意一个客户端发送的文本向所有客户端（包括发送方）转发。

2. 设计思路

2.1 客户端程序

```
public class ChattingClient
```

ChattingClient类实现客户端程序，有以下成员变量：

```
static final int SIZE;  
static final int PORT;  
Socket sk;  
OutputStream sk_out;  
InetAddress addr;  
boolean isrunning;
```

SIZE定义发送消息的最大长度，PORT定义连接到服务器的端口，sk为使用的连接到服务器的Socket，sk_out用来接收该Socket的输出流，addr为客户端的地址，isrunning检测是否输入“quit”关闭客户端。

ChattingClient类中有两个模块实现主要功能：构造函数和输出线程类OutputThread。

```
public ChattingClient()
```

构造函数主要实现Socket的建立和从标准输入获取输入流后写入Socket的过程。

首先利用InetAddress.getByName(null)获取自身地址，之后使用这个地址进行Socket的初始化，创建输出线程对象。之后开启标准输入，循环执行直至输入“quit”表示退出，每次执行过程将输入内容写入Socket的输出流中。在检测到“quit”后关闭Socket和输入终端。

```
class OutputThread extends Thread
```

OutputThread是ChattingClient的内部类，用于不断接受Socket的InputStream，通过线程可以实现写和读同时执行。OutputThread类内只有一个成员变量InputStream，在run函数中的循环中该变量获得socket的InputStream进行输出。循环终止条件为isrunning为false。

2.2 服务器程序

```
public class ChattingServer
```

ChattingServer类实现服务器程序，有以下成员变量：

```
public static final int PORT;
public List<ReceiveThread> ClientList;
ServerSocket server;
int num;
```

PORT定义ServerSocket使用的端口，ClientList存储连接的客户端列表进行消息发送，server为ServerSocket对象负责监听Socket，num记录连接的Client的数量为其编号。

ChattingServer类中有三个模块实现主要功能：构造函数、接受线程类ReceiveThread和发送线程类SendThread。

```
public ChattingServer()
```

在构造函数中先进行num初始化，ServerSocket建立，打印初始化消息。之后进入循环，接受到连接的socket后创建新的接受线程对象并启动，将该对象添加到ClientList表内，之后调用向所有表内的client发送提示消息。

```
class ReceiveThread extends Thread
    static final int MESSAGE_SIZE;
    Socket sk;
    InputStream in;
    OutputStream out;
    int id;
```

接受线程类ReceiveThread有5个成员变量，MESSAGE_SIZE表示最大消息长度，sk、in、out为监听到的Socket信息，id表示client编号。ReceiveThread的run函数中对Socket的输入流进行read操作读入到事先开好的byte数组中，删去多余byte后转化为字符串进行检测，若为“quit”则切断连接，从ClientList中删除该Socket，向其余主机发送提示消息；否则将该client发送的消息转发给所有在List中的client。

```
class SendThread extends Thread
    Socket sk;
    String message;
```

发送线程类SendThread有2个成员变量，sk为待发送消息的Socket，message为待发送的消息。在run函数中获取sk的OutputStream，调用write函数将message转化为byte数组后写入其中，完成消息的发送。

3. 演示与结果

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingServer.jar
[Server start]
```

启动Server

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingClient.jar
[Client 1 joined the chatting room]
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
```

```
PS E:\work\java应用技术\javacode\project3> java project3.ChattingClient
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
```

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingClient.jar
[Client 3 joined the chatting room]
```

在多个终端启动Client，先加入的客户端接受到后加入的客户端进入聊天室的提示消息。

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingClient.jar
[Client 1 joined the chatting room]
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
Hello
[Client 1] : Hello
```

```
PS E:\work\java应用技术\javacode\project3> java project3.ChattingClient
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
[Client 1] : Hello
```

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingClient.jar
[Client 3 joined the chatting room]
[Client 1] : Hello
```

客户端1发送消息，所有客户端都收到其发来的消息。

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingClient.jar
[Client 1 joined the chatting room]
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
Hello
[Client 1] : Hello
[Client 2 left chatting room]
[Client 3] : nice to meet you
```

```
PS E:\work\java应用技术\javacode\project3> java project3.ChattingClient
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
[Client 1] : Hello
quit
[Quitted the chatting room]
PS E:\work\java应用技术\javacode\project3> []
```

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingClient.jar
[Client 3 joined the chatting room]
[Client 1] : Hello
[Client 2 left chatting room]
nice to meet you
[Client 3] : nice to meet you
```

客户端2输入quit，退出聊天室。其他两个客户端收到提示消息，并且可以继续聊天。

```
PS E:\work\java应用技术\javacode\project3> java -jar ChattingServer.jar
[Server start]
[Client 1 joined the chatting room]
[Client 2 joined the chatting room]
[Client 3 joined the chatting room]
[Client 1] : Hello
[Client 2 left chatting room]
[Client 3] : nice to meet you
```

Server端也输出所有消息记录。

4. 心得体会

本次实验用到了Socket网络编程，这是比较不熟悉的内容，因此感到十分困难。但在实际研究过后发现并不复杂，最终完成项目也并不十分困难，只是需要认真学习不熟悉的领域。本次实验中学到的知识也可以加深对计算机网络的理解，对另一门课程的学习也会有所帮助。