课程设计报告二：聊天室系统

1. 课程设计要求与目的

目的：编写一个小型Java聊天室系统，掌握Java网络通信、多线程、IO文件操作等高级应用编程技能。

要求：以课本第15章 Java网络通信例15.3、15.4的源代码为基础，编写一个小型Java聊天室系统。

完成如下功能：

1、多客户端模式下，实现客户与客户的单独通信，要求信息通过服务器中转（DONE）

2、端到端的通信，实现并行通信模式（不再是你说一句，我说一句，一端的信息发送不受另一端的影响）（DONE）

3、实现端到端的文件传输（DONE）

4、添加图形界面（选做）（DONE）

1. 系统设计

服务器端和客户端之间的通信TCP靠实现客户端与客户端以及客户端与服务器端之间通信

**客户端：**首先运行login.java文件，弹出框口，输入用户名，端口号，Ip地址，并通过事件判断用户输入的合法性。如果输入合法字符则调用ChatClient()构造函数，构造函数有三个参数**public** ChatClient(String userInputName,String ip,**int** port)，关闭当前窗口打开新窗口（聊天界面），并将用户的username,ip.port显示在新窗口上，并调用connect（）函数，connect()开启一个新线程用来接受消息。并且把用户username，输入流和输出流传给服务器端，服务器端通过链表保存用户信息。

1.用户首先通过下拉菜单选择发送消息的对象，并将聊天对像的username发送到服务器端，代码实现：



1. 服务器端通过客户端发送的用户民检索聊天对象的输入流和输出流反馈给改用户
2. 用户在输入跨过中输入发送的消息，在该模块中进行判断，如果是文件路径，并且文件存在在本机中那么用户发送文件，否则发送消息，代码实现：

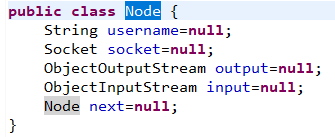


其中message就是输入框的内容。传输的时候会先给出一个标志字段，如果是传输文件就按照接受文件方式接受，如果是传消息就按普通方式接受消息

1. 用户接受消息模块是ChatReceive.java，服务器端会判断信息类型，有系统消息，用户列表，聊天消息三类，其中聊天消息可以是文件发送或者是消息发送，通过equalsIgnoreCase函数判断，不区分大小写，然后进行相应的工作
2. 客户端聊天下面的显示人数和下拉菜单选择收发消息对象，每次新用户上线，服务器端都会重新发送用户具体信息到客户端，客户端显示

**服务器端**：服务器端主要是通过链表实现用户信息的存储，然后定义了根据索引，名字查找相应的node字段以及遍历整个链表

代码：



存储的是每个用户的username,socket,objectInputStream,objectOutputStream,还有指向的下一个Node的信息

1. 服务器转发消息：用户通过Socket传输数据到服务器端，服务器端通过该用户聊天对象的usrname找到聊天对象的outputStream或者inputStream然后进行收发消息
2. 传文件的时候会在服务器端本机上保存用户之间的传输文件
3. 服务器端也可以实现和任意客户端之间的通信，服务器端也会显示所有聊天信息，包括群发消息和私发消息
4. 系统实现

**服务器端用户信息的存储实现并实现相应的遍历搜索**：用自定义链表实现用户username,objectInputStream,objectOutputStream等信息的存储，并自定义链表的搜索遍历

代码：**public** **class** Node {

String username=**null**;

Socket socket=**null**;

ObjectOutputStream output=**null**;

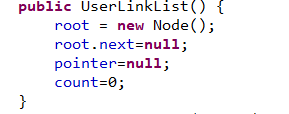
ObjectInputStream input=**null**;

Node next=**null**;

}

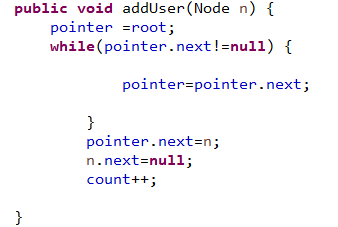
定义节点类，组成链表的基本元素，并定义next指向链表中的下一结点

链表的初始化：在userLinkList.java中实现，定义了userLinkList类元素是node节点：



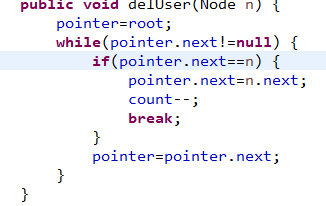
定义一个只有一个node节点的链表，root节点

在链表中添加新节点：



函数传参，参数是新节点，先吧point指向根节点，遍历链表，找到末尾节点，然后pointer.next=n，添加该节点到链表中，count加一（节点个数加一）

删除链表中的某个节点：



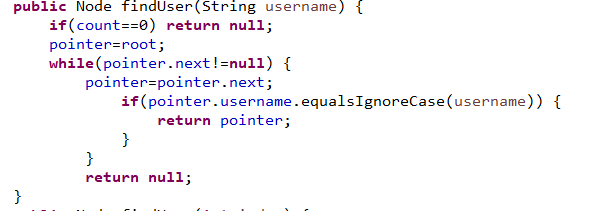
通过链表遍历找到n节点所在的位置，然后使他前方的节点指向他后方的节点，然后count减一就实现了指定节点的删除工作

获取链表节点数：

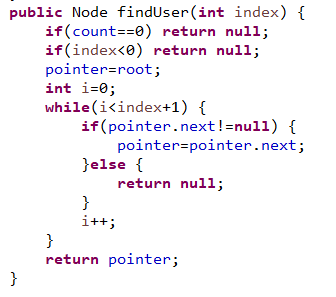


返回count值，count用来记录当前链表节点个数

通过username找相应的节点：



首先将pointer指向root（第一个节点），pointer也是node类型的变量，通过next遍历，比较username的值，若找到就返回pointer，它就指向了要找的节点。如果没找到则返回null；  
 通过索引找相应的节点：



首先进行判断，链表是否为空或者是否处于异常状态，如果是返回空，否则通过遍历找到索引对应的节点，返回相应节点

**客户端用户登录界面的实现：**主要是用了java awt.\*和javax.swing.\*包，布局用的是GridLayout和BorderLayout两种布局方式，通过在登录按钮绑定事件监听器检验用户的输入合法性，在输入端口号的时候由于用户输入的是字符串，在进行判断的时候需要Integer.*parseInt（）*将字符串变成整型在进行判断，字符串String类型的数值相互比较时候不能直接双等，需要用equal()方法进行判断

**聊天界面的实现：**主要是用了java awt.\*和javax.swing.\*包，布局用的是GridLayout和BorderLayout两种布局方式，在登陆框输入用户合法信息后会变化窗口，在login.java中单击按钮后会创建新对象并调用connect()函数是实现客户端与服务端之间的连接，其中发消息是一个线程，接受消息又是一个线程

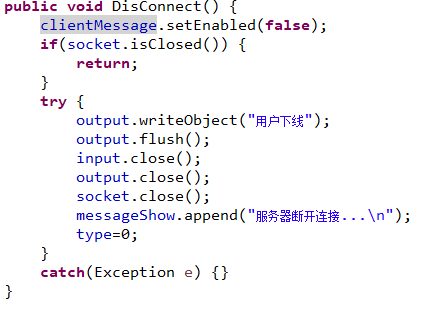
客户端与服务器端之间的连接：



把该用户的用户名，ObjectOutputStream，ObjectInputStream传到服务器端，服务器端把用户信息保存在链表中

客户端与服务器端断开连接：

与服务期断开连接，关闭各种流，并将发送消息的按钮设置销毁，并在来聊天界面显示服务器断开连接



用户像服务器端发送消息及文件：通过object流传送消息（以对象方式），传输文件时候，如果用户输入的是文件地址，且该文件存在，则进行文件传输，通过read方法读取文件的内容以字节的形式写入到输出流，再flush到输入流。服务器端通过判断普通消息还是文件进行相应操作，如果是文件，则以字节的形式读取文件的内容存储到新文件中

客户端聊天界面相关信息同步的设立：客户端信息需要与服务器端数据进行同步，每次有新用户登录或者用户退出，相应的链表得到更新，然后反馈到客户端的界面进行局部刷新

**服务器端相关实现：**服务器端界面主要是用了java awt.\*和javax.swing.\*包，布局用的是GridLayout和BorderLayout两种布局方式，服务器在两通信用户之间扮演的是中转站的角色，接受发送消息客户的信息转发个其聊天对象

服务端设置端口号：在PortConf.java中实现相关端口号的设置

服务器段信息转发实现：在ServerRecive.java文件中实现相关操作，群发共呢个实现，通过之前定义的遍历链表的方法实现对所有用户群发消息的操作，并且通过标志位的设定判断用户进行那些操作以及传输什么信息

1. 系统测试

|  |  |
| --- | --- |
| 操作 | 结果显示 |
| 运行服务器端 |  |
| 服务器默认端口号是8888，更改端口号 |  |
| 更改后的端口号为6666，启动服务器端的服务 |  |
| 服务器端停止服务 |  |
| 点击是 |  |
| 退出服务器 |  |
| 点击是 | 界面消失 |
| 客户端登录（运行客户端login.java） |  |
| 用户民为空（异常操作） |  |
| 点击登陆后，显示用户名不能为空 |  |
| ip地址并不合法（异常操作） |  |
| 点击登录，提示wrong ip |  |
| 端口号输入不合法（异常操作） |  |
| 点击登录，显示错误的端口号请填写整数，且范围在0~65535之间 |  |
| 用户正常登录（服务器端开启端口号为8888监听） |  |
| 点击登录，显示连接服务器成功，聊天界面显示用户名，ip，port，在线用户 |  |
| 服务器端，显示在线人数以及用户上线情况 |  |
| 再次登录一个新用户，客户端在线人数显示有两人 |  |
| 服务器端显示王思聪上线，在线用户两人 |  |
| 再登录一个用户 |  |
| 用户之间的私聊（王思聪同黄思聪之间） | 通过下拉菜单栏选择聊天对象 |
| 用户黄思聪界面 |  |
| 用户王思聪界面 |  |
| 用户李宸界面，未收到任何消息 |  |
| 服务器端，显示消息 |  |
| 群发消息（李宸发出） |  |
| 用户王思聪界面 |  |
| 用户黄思聪界面 |  |
| 服务器端界面 |  |
| 用户退出登录（李宸退出） | 用户李宸聊天界面消失 |
| 用户王思聪界面显示李宸下线，在线用户两人 |  |
| 用户黄思聪界面显示李宸下线，在线用户两人 |  |
| 服务器端显示用户李宸下线，在线用户两人 |  |
| 发送文件（在本地计算机g盘有123.txt）文件，显示发送文件 |  |
| 服务器端在E盘产生文件夹temp,保存改文件 |  |
| 客户端在E盘产生文件夹client,保存文件 |  |
| 聊天结束，服务器端停止服务 |  |
| 客户端显示服务器已关闭 |  |
| 服务器端发消息 |  |

1. 课程设计总结

java聊天器实现的系统功能：

1、多客户端模式下，实现客户与客户的单独通信，要求信息通过服务器中转

2、端到端的通信，实现并行通信模式（不再是你说一句，我说一句，一端的信息发送不受另一端的影响）

3、实现端到端的文件传输

4、显示在线人数以及聊天对象（下拉菜单栏显示），异步刷新

5、私发消息，群发消息，私法文件，群发文件

在做java聊天器时候遇到许多困难，也犯过许多低级错误（把作用为标志的汉字打错以至于在进行字符串相等判断的时候一直出BUG），尤其是在进行消息同步的时候的时候，想过用许多方法实现比如（可以共同使用同一个footer），但最后发现只要用户连接，退出，就可以调用相关函数全部渲染相关窗口信息就行了，用函数有用到一些数据结构相关的知识，实现链表，以及查询，遍历等。

计算机与其它专业最大的不同点就是与计算机的相关几乎所有知识都是开源的，你可以去上网理解别人的体会，去了解别人的思路，所谓见多识广看的多了自然也懂的多，在遇到问题的时候自己就经常百度甚至翻墙去查阅相关资料，再结合自己的体会修改代码，在做这个项目的时候也理解到了版本控制的重要性。

多用git。