

浙江大学计算机科学与技术学院

Java 程序设计课程报告

2022—2023 学年 1 学期

题目	JAVA 聊天软件
学号	3200105705
学生姓名	曹小川
所在专业	计算机科学与技术
所在班级	计科 2001

1 引言

我本次开发的是一个多人在线聊天软件，需要运用到 Java 语言中的 GUI 编程知识，网络编程知识，并发编程知识以及其他 Java 语言的有关知识。通过这样一个综合性的项目设计与开发，可以对课堂上学习到的理论知识进行进一步的理解吸收，更是可以在实践过程中提高自己使用 Java 语言解决实际问题的能力。

1.1 设计目的

聊天软件是互联网时代每个人都在使用的软件，本次项目的目标如下：

- (1) 锻炼使用 Java 语言开发大型项目的能力；
- (2) 熟悉 Java 语言中的 GUI 编程，熟悉 Swing 框架的使用；
- (3) 熟悉 Java 语言中网络编程相关的类和方法，能够利用 Java 语言实现多台计算机之间的网络通信；
- (4) 熟悉 Java 语言中的并发编程，通过并发编程技术实现聊天软件支持多组用户同时使用的功能。

1.2 设计说明

本程序采用 Java 程序设计语言，在 Eclipse 平台下编辑、编译与调试。具体程序分为客户端和服务端两部分，均由本人独立开发完成。

2 总体设计

2.1 功能模块设计

本程序需实现的主要功能有：

- (1) 服务器端可以监听指定的端口，并与发出连接请求的客户端建立 TCP 连接；
- (2) 服务器接收到连接请求并建立连接之后，通过创建一个子线程来实现与该客户端的后续通信。
- (3) 服务器转发一个客户端的消息到所有已连接的客户端；
- (4) 客户端通过指定 IP 地址和端口号，连接到指定的服务器；

(5) 客户端发送消息给其他连接到同一服务器的客户端;

3 详细设计

系统的整体布局为: 整个程序使用了 chat_GUI, Client_Socket, client, Clientlogin_GUI, Server_GUI, Server_Socket, Server 几个类来实现, 分别对应客户端与服务器端。

3.1 客户端主函数设计:

Client 类作为启动的主函数, 负责打开登录窗口, 启动 client 的函数。

3.2 客户端登录界面设计

Clientlogin_GUI 类主要使用了 Java 的 Swing 框架进行设计, 该类继承自 javax.swing 包中的 JFrame 类, 还实现了 ActionListener 接口来响应用户按下登录按钮的事件。

用户手动传入三个参数, 按下登录后检查用户输入是否合法, 如果合法则跳转到聊天界面。

3.3 用户聊天界面设计

使用滑动窗口来作为聊天窗口

用户在下面的输入框输入信息, 通过 socket 封装发送给服务器, 并且点击 clear 即可通过将输入窗口设置为空来清理信息。

ChattingTaDisplay(String msg) 方法的功能为将传入的字符串 msg 显示到 chattingTa, 也即对话框中, 通过调用 JTextArea 类的 append() 方法实现。

UserlistTaDisplay(String msg) 方法的功能为将传入的字符串 msg 显示到 userlistTa, 也即在线用户列表框中, 也是通过调用 JTextArea 类的 append() 方法实现。

3.4 套接字类实现

Client_socket 类主要处理与服务器的连接，需要实现与服务器连接有关的一些方法。主要运用到了 Java 中 Socket 编程的思想以及多线程编程的思想。

Client_socket 类的 connectServer() 方法用于连接服务器，是该类设计的核心。其主要流程为：

- (1) 实例化一个 Socket 对象，指定服务器名称和端口号来请求连接；
- (2) 如果连接成功，接着创建套接字的输入输出流。
- (3) 如果创建也成功，则启动一个子线程来完成后续与服务器的通信。

最主要是 Client_socket 类的 sendMessage(Message msg) 方法，该方法的主要作用是给服务器发送消息，通过向套接字输出流中写入 Message 类的对象 msg 即可实现。

3.5 Server 界面设计

Server_GUI 类除了显示服务器界面，在最开始还通过一个可输入的对话框，获取了要监听的端口号。在最后创建了一个 Server_socket 类的对象，并调用其 listen() 方法来监听指定的端口。

3.6 Server_Socket 类设计

Server_socket 类的 listen() 方法用于监听端口并等待客户端连接，是该类设计的核心。其主要流程为：

- (1) 根据指定的端口号实例化一个 serverSocket 对象；
- (2) 调用 serverSocket 类的 accept() 方法，该方法将一直等待，直到客户端连接到服务器上给定的端口；
- (3) 如果通信建立成功，accept() 方法返回服务器上一个新的 socket 引用，该 socket 连接到客户端的 socket；
- (4) 创建一个子线程处理后续与该客户端的通信；
- (5) 将新创建的子线程加入保存与客户端通信的子线程的 ArrayList 中
- (6) 发送提示消息给其他已连接的客户端，在服务端也输出相关提示信息，然后循环，继续调用 accept() 监听端口。

3.7 服务器主函数端设计

只需要调用 `Server_GUI` 类创建登录界面即可。

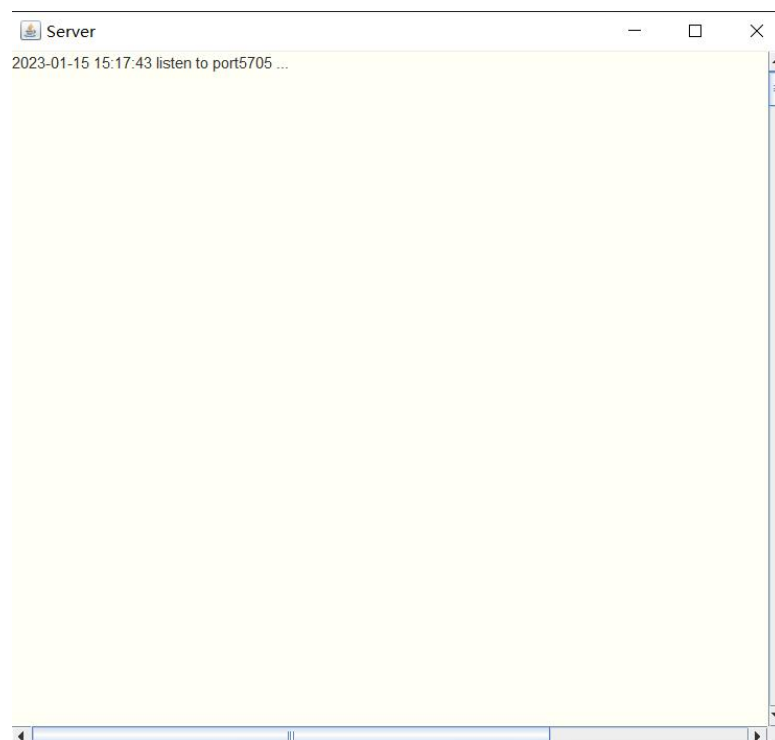
4 测试与运行

4.1 程序测试

服务器端登录：



服务器端初始界面：



客户端登录界面：

client login

Welcome

server IP addr:

port:

user name:

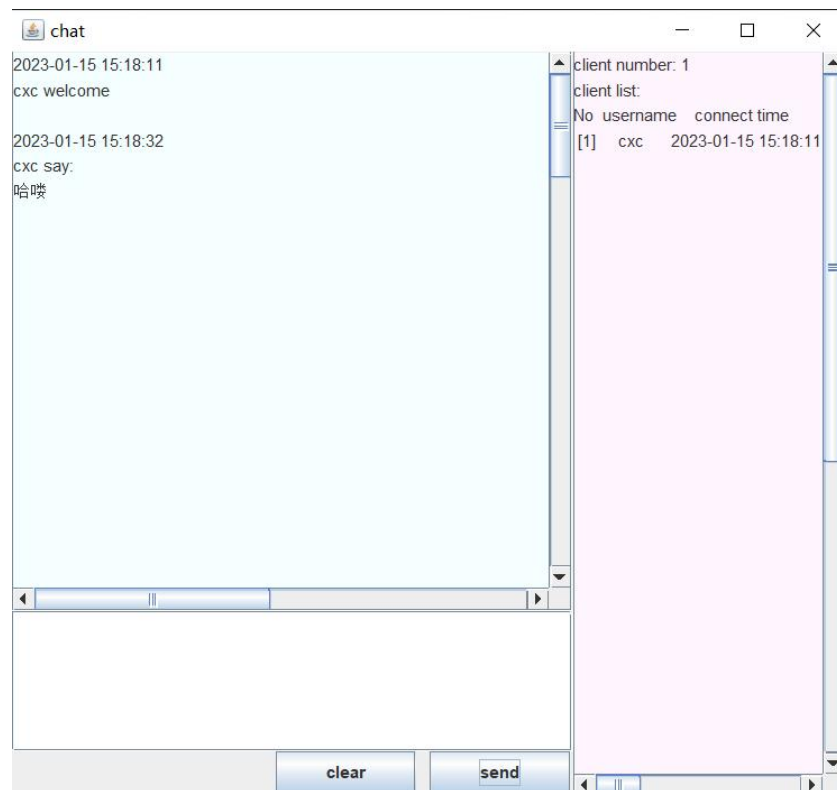
客户端进入初始界面：

chat

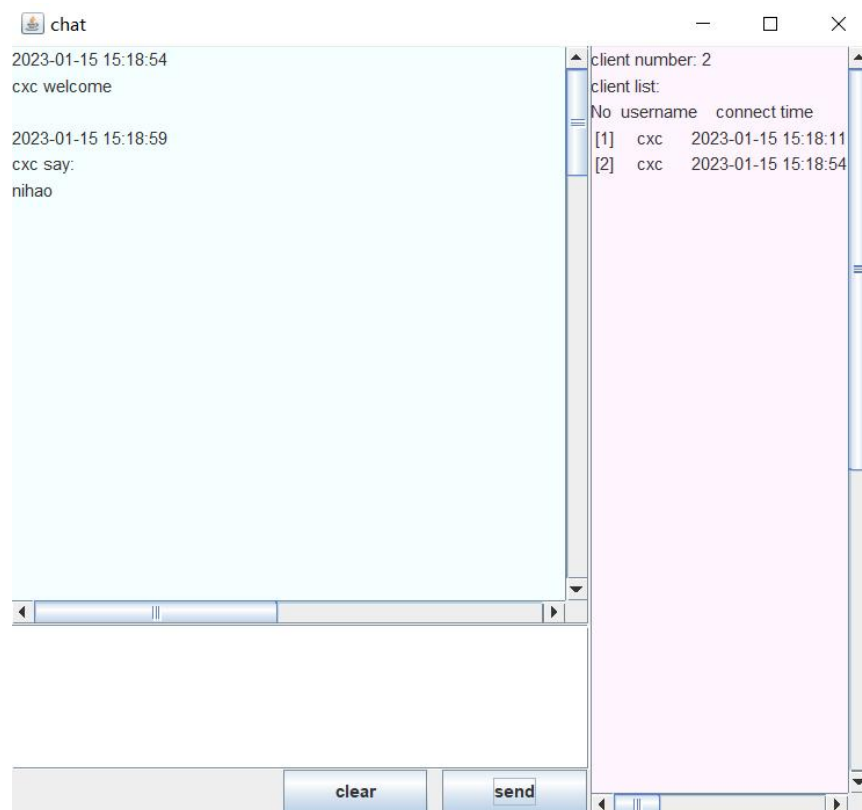
2023-01-15 15:18:11
cxc welcome

client number: 1
client list:
No username connect time
[1] cxc 2023-01-15 15:18:11

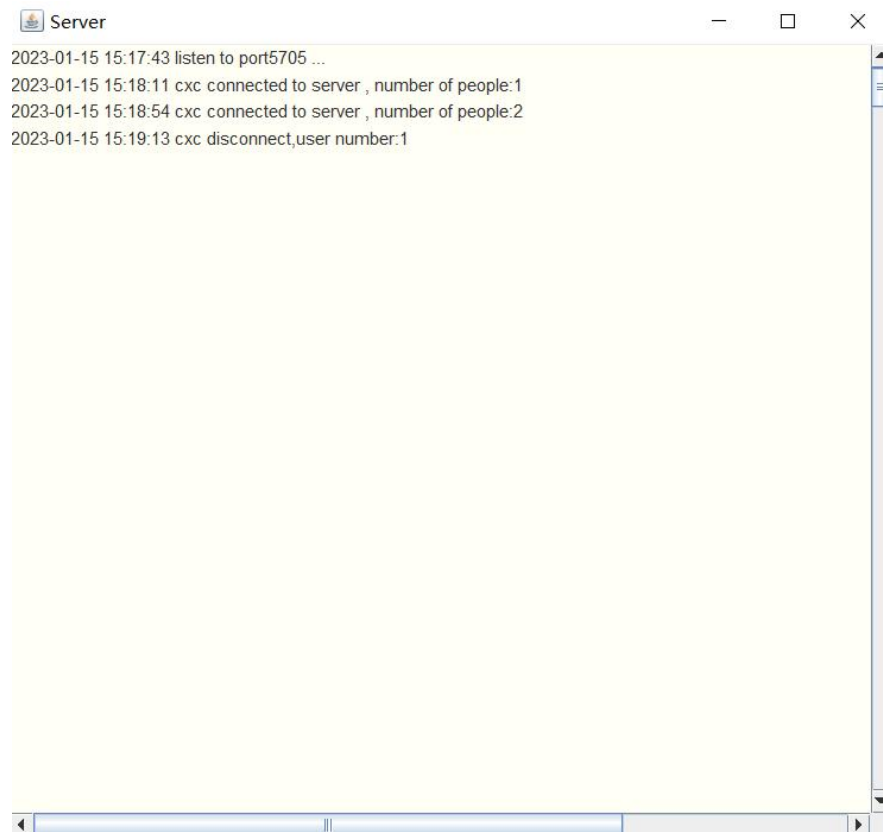
客户端发送信息：



客户端 2 登录:



客户端 2 退出，服务器端界面：



5. 总结

本次我开发了多人聊天软件，主要运用了课上学的 GUI 编程以及多线程编程，网络编程。因为在计网课上以及学过 Socket 编程，所以 socket 部分完成较为轻松，而且 Java 的 socket 编程相较于 C 语言还是要简单一些，我主要在 GUI 编程部分耗费了较多的时间，但是经过实际的操作我发现 Swing 框架也比没有那么困难，最终顺利完成了实验，通过这次实验，我对于 socket 以及 GUI 有了更进一步的掌握，收获颇丰。

