

基于Socket通讯原理的网络 聊天室通讯

阶段1

- 第一阶段：利用Socket通讯原理，来实现服务器和客户端之间发送信息和接收信息的简单信息通讯，
- 扩展要求：其服务器和客户端各有界面。

基本设计思路（有图形界面）

- 设计client.java和server.java为主要核心文件
- 定义 infomation 类
- 设计ClientFrame.java和ServerFrame.java两个界面类
- 可以设计ChatListener.java的一个监听器类为客户端界面和服务端界面添加监听。

阶段2

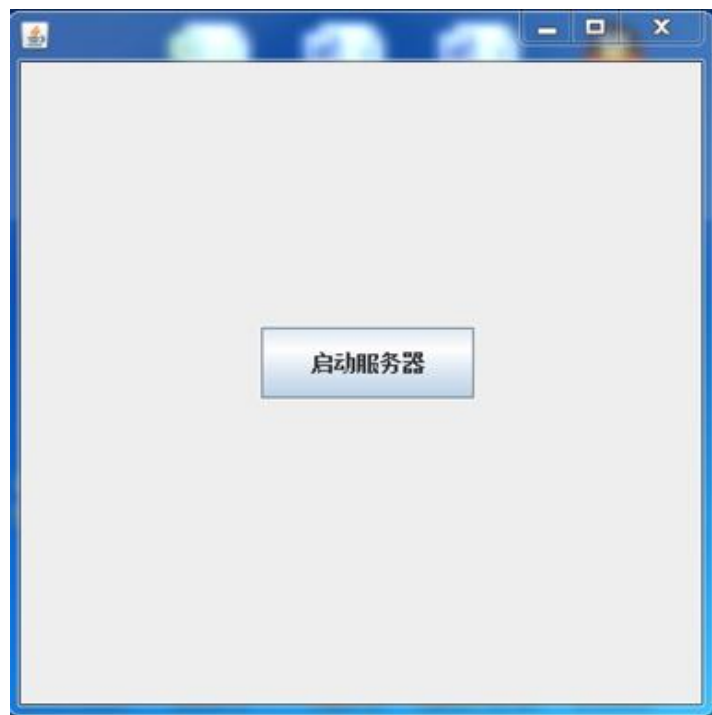
- 依然是服务器与客户端一对一的模式。
- 添加2-3个线程
- 线程1 客户端接受消息的线程 ClientGetThread.java
- 线程2 服务器接受信息线程ServerGetThread.java
- 或者再加一个线程类3服务器发送信息线程ServerSendThread.java

基本设计思路2（有图形界面）

- 设计client.java和server.java为主要核心文件
- 定义 infomation 类
- 设计ClientFrame.java和ServerFrame.java两个界面类
- 可以设计ChatListener.java的一个监听器类为客户端界面和服务端界面添加监听。
- 线程1 客户端接受消息的线程 ClientGetThread.java
- 线程2 服务器接受信息线程ServerGetThread.java

阶段3

- 第三个版本：用多线程来使得多个客户端可以进行收发消息，即服务器端利用多线程为每一个客户端进行服务。



- 服务器端：
- (1) 服务器程序：
- --能接受多个客户端的连接请求（注意：为了仿真实现可通过用户名不同来区分不同的客户端）
- 思考： (1) 多个连接应该产生几个Socket对象？
- (2) 怎么存放和区分这几个Socket对象？
- --一旦户端选择了聊天对象后和服务器连接上了，应该创建一个线程进行对该次连接的Socket进行服务

- 服务线程（可以是1个收信息线程， 1个发信息线程）
- 主要工作
 - （1）接收客户端发送的信息，（注册信息， 登录信息， 断开连接信息， 其他信息等）
 - （2）解析客户端发送的信息内容：信息来源A、内容、信息发送的目的用户B等
 - （2）能够找到将内容转发给用户B（关键是用用户B和对应的Socket）
 - （3）能够实现群发功能

客户端:

(1) 客户端程序: 客户端程序可以连接服务器并向服务器发送信息(发送方用户名, 发送方IP, 时间, 聊天内容, 接收方用户姓名和IP等)。

线程1 客户端接受消息的线程 ClientGetThread.java

