

ChartServer

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。

一些约定

- Server将占用 4657 端口，Client需要连接 4657 端口
- Client发送的所有消息将会向所有客户端转发。当客户端发送 "END" 时，Server将会视作该Client结束对话而放弃与之的连接。

运行JAR

1. 打开 `./jar` 文件夹
2. 运行命令 `java -jar ChatServer`

编译运行

1. 打开 `./src` 文件夹
2. 运行编译指令 `javac NetTest.java`
3. 运行指令 `java NetTest`

类关系

- NetTest
 - Server
 - ClientManager
 - ClientReader
 - ClientMessageSender
 - MessageSender

实现细节

ClientManager

当有任意Client发起连接请求时，Server将会创建一个 `ClientManager` 类实例来记录该Client的Socket以及其他信息。该类实例将会被放入一个 `ArrayList`。

```
1 | ArrayList<ClientManager> clientManagers = new ArrayList<>();
```

Client读写线程分离

每一个 `ClientManager` 将会拥有两个对应的线程，分别为：

- `ClientReader`：负责读取对应的Client传来的消息
- `ClientMessageSender`：负责向对应的Client发送消息

两个线程互相独立。通过位于 `ClientManager` 的布尔变量 `isRunning` 来调配是否结束运行。

群发消息

`MessageSender` 为Server下的单例。当有 `ClientManager` 收到消息时，将会通知Server调用 `MessageSender` 中的方法来群发消息。

消息队列

整个Server采用了消息队列来处理并行的大量的消息涌入。

```
1 Queue<String> msgQueue = new LinkedList<>();
```

在 `MessageSender` 和每一个 `ClientMessageSender` 中都有消息队列来有序调配消息的发送。