|  |  |  |
| --- | --- | --- |
| **Request** | **Response** | **Bemerkungen** |
| auth <username> | accept | Kann einmal authentifiziert werden |
| decline | decline kann mehrmals auftreten wenn der Benutzername nicht akzeptiert wird, dann muss auth erneut durchgeführt werden |
| list\_users | users <usernames> | Kann je nach Bedarf ausgeführt werden;  <usernames> werden per Leerzeichen getrennt |
| smsg <message> | rsmg <username> <message> | Ausführung nach Bedarf möglich |
| quit | bye | Es kann einmal quit aufgerufen werden;  Die Verbindung wird dann geschlossen |

Standardport: 56789

### Spezifikation der Klassen und Methoden

#### Klassen des Chatclients und die dazugehörigen Methoden

**ChatClient Klasse:**

Diese Klasse hat die Aufgabe den Benutzer zu authentifizieren, Sender- und Empfängerthreads zu starten und die Serveradresse, Port zu lesen und den Client zu starten.

In dem Konstruktor wird ein neuer Socket erzeugt, sowie ein Input und Outputstream für den Socket erzeugt.

*void run():* Fragt den Benutzer nach seinem Namen und liest diesen ein, ist dies erfolgreich wird der empfänger- und sender-Thread gestartet.

*boolean authenticate(username):* Hier wird geprüft, ob der Benutzer vom Server authentifiziert werden konnte. Diese Methode ist private und wird in run() benutzt.

*void main(String args[]):* Hier werden die Properties-Dateien eingelesen und der ChatClient gestartet.

**ChatClientSenderThread:**

Diese Klasse erbt von Thread und hat nur die Methode run() implementiert. Diese Methode wartet auf Eingabe von dem Benutzer und liest die Nachrichten ein. Sollte die Nachricht „/quit“ lauten, wird quit in den outputstream gegeben. Außerdem gibt es noch den Befehl „/list-users“, welcher die Teilnehmer Liste des Chats ausdruckt.

**ChatClientEmpfaengerThread**:

Diese Klasse erbt ebenfalls von Thread und implementiert nur die Methode run(). Wenn etwas empfangen wurde gibt er diese aus. Es kann auch sein, dass ein neuer Benutzer authentifiziert wurde, dann fügt er diesen zur Userliste hinzu. Wenn „bye“ empfangen wurde wird der ClientEmpfaengerThread beendet.

**ChatClientUI:**

DieseKlasse ist ein Interface und wird von ChatClientCLI.

*String getEingabe()*

*void showStatusmeldung(String meldung)*

*void showUsers(List<String> users)*

*void showMessage(String username, String message*

**ChatClientCLI:**

Implementantion von ChatClientUI. implementiert.

*String getEingabe():* Liest die nächste Eingabe und gibt diese wieder

*void showStatusmeldung(String meldung):* Gibt die eingegebene aus dem Outputstream wieder

*void showUsers(List<String> users*): Zeigt die Liste der User an

*void showMessage(String username, String message):* Gibt die eingegebene Nachricht aus und gibt sie in der Form(Name: Nachricht aus)