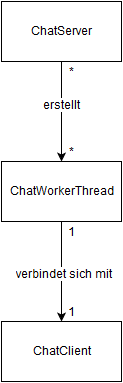
1.

TCP weil:

TCP stellt zuverlässige reihenfolge-erhaltende Byte-Übertragung ("pipe") zwischen Client und Server bereit

2.

Zeichencodierung: UTF

Port: wird gewählt und bleibt dieser Port (standard bei uns 4567)

**Präfixes vom Server zum Client:**

.auth:

server sendet diesen Befehl nach dem Verbindungsaufbau um anzudeuten dass eine authentifizierung getätigt werden muss, hierbei muss der client dann einfach nur den namen als String schicken

.authfail:

sendet der Server dem Client wenn der Name den dieser gewählt hat bereits vergeben ist

.ack:

Sendet der Server dem Client nachdem er den Namen akzeptiert hat und schaltet den Client frei

.new:

Sendet der Server an alle Clients um anzudeuten dass jemand neues den Chatraum verlassen hat, um so ggf. die Chatteilnehmerliste anzupassen. Der Client kann dann ggf. mit „.clients“ eine Anfrage an den Server schicken um eine aktualisierte Teilnehmerliste zu bekommen oder diese Nachricht ignorieren.

.clients:

Sendet der Server an den Client um alle aktuell im Chatraum verfügbaren Clients mit Namen mitzuteilen. Bsp. „.clients Hans; Peter; Klaus“

.message

Sendet der Server an den Client um eine neue Nachricht zu übersenden. Bsp. „.message hallo“.

**Präfixes vom Client zum Server:**

.quit

Sendet der Client an den Server um anzudeuten den Chatraum verlassen zu wollen und der Server sendet das zurück damit der Client die Verbindung auch beendet

.message

Werden einer Message die an alle gesendet werden soll voran gestellt. Alles was ohne dieses Präfix gesendet wird, wird einfach ignoriert.

.clients

Sendet der Client an den Server und der Server antwortet mit der Liste der Clients die im Chatraum sind, getrennt von einem Semikolon.  
Beispiel „.clients Hans; Peter; Klaus“

Beispielsession bei Auth Erfolg:

Nachdem die Verbindung aufgebaut worden ist:  
Server: .auth  
Client: Tim  
Server: .ack  
Server: .new  
Server: .message Tim hat den Chatraum betreten!  
Client: .message Hallo  
Server: .message Tim: Hallo

Beispielsession bei Auth Misserfolg:

Server: .auth  
Client: Tim  
Server: .authfail  
Server: .auth  
Client: Tim2  
Server: .new  
Server: .message Tim2 hat den Chatraum betreten!  
Client: .message Hallo  
Server: .message Tim2: Hallo

3.

ChatServer

Methods:

run: (Portnummer) startet einen Chatserver welcher neue Verbindungen annimmt und ChatWorkerThreads mit den Sockets erstellt  
  
  
ChatWorkerThread

Methods:

constructor: (socket) erzeugt einen neuen ChatWorkerThread mit übergebenem TCP-Socket

run: authentifiziert zuerst den benutzer um danach den input des clients zu verarbeiten und zu broadcasten

broadcast: (string) sendet den übergebenen String an alle teilnehmenden clients des „Chatraums“

Chatclient

Methods:

constructor: erstellt einen neuen Chatclient

run: erstellt zunächst die Socketverbindung zum Server um danach den input vom server zu verarbeiten

Chatclient benötigt einen Eventmanager der eingegebene Nachrichten zum ChatWorkerThread über den Socket schickt.