

Transferarbeit

Applikationsentwickler

1. Auftrag

Für ein Chatprogramm sollen Sie den entsprechenden Server sowie Client schreiben. Der Server soll auf der Konsolenbasis Nachrichten von Chat-Benutzern (Clients) aufnehmen und diese hinterlegen. Die Clients sollen die Möglichkeit haben, den Verlauf dieser Nachrichten vom Server anzufragen. Der Chat-Client soll als kleine GUI-Applikation aufgebaut werden.

2. Endprodukt

Server und Client sollen über das Protokoll TCP kommunizieren. Dabei soll der Server jede neue Client-Verbindung in einem separaten Thread behandeln. Verbindet sich ein Benutzer mit dem Server, soll dieser dem Server einen Benutzernamen übergeben. Der Server wiederum soll beim Erhalt von Chat-Nachrichten diese im Chat-Verlauf mit dem entsprechenden Benutzernamen hinterlegen: *Benutzer1: Hallo*

Der Server soll den Chatverlauf in einer ArrayList aufbewahren. Das Lesen und Schreiben dieser ArrayList bzw. des Chatverlaufes soll über **adaptive Semaphor** gelöst werden: Es sollen bis zu 10 Threads gleichzeitig von dieser Liste lesen, jedoch nur ein Thread darauf schreiben können. Sie können dazu die Java interne Semaphor verwenden:

<https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Semaphore.html>

Die Grundlagen zu den Semaphor können Sie aus der Beschreibung zur Fallstudie (parallele und verteilte Systeme, Applikationsentwickler) entnehmen.

Das Client-Programm soll mit einer einfachen GUI-Oberfläche ausgestattet sein und folgende Elemente aufweisen:

- Textfeld für Benutzername
- Verbindungsknopf
- Textbereich für Chatverlauf
- Textfeld für neue Chatnachricht
- Sende-Knopf (um die neue Chatnachricht zu versenden)

Der Client soll jede Sekunde den Verlauf der letzten 50 Chatnachrichten vom Server verlangen und diese im Chatverlauf-Textbereich anzeigen.

3. Informationen

1. Kommunikation

Um verschiedene Aktionen beim Gegenpartner ausführen zu können, sollen diese folgende Befehlsnachrichten verstehen:

	Server	Client
<i>send</i>	erhält Nachricht	erhält Nachricht
<i>get</i>	soll den Verlauf der letzten Nachrichten mit der angegebenen Anzahl an Zeilen zurück geben	-
<i>register</i>	soll für eine Verbindung einen Namen hinterlegen	-

Für die Kommunikation können Sie eine Zeichenkettenstruktur verwenden, indem Sie den Befehl vor der Nachricht hinterlegen und mit dem Zeichen # abtrennen: *send#Hallo*

Dabei entsteht folgende Struktur:

- *send#Nachricht*
- *get#Anzahl Zeilen*
- *register#Namen*