

## Simple-Chat

Entwickeln Sie eine Chatsoftware die Befehle entgegen nimmt und in der beliebig viele User beliebig vielen Chatrooms beitreten können. Sollte es zu einer Fehlermeldung bei der Abarbeitung eines Befehles kommen wird dieser "abgebrochen" und es wird wieder auf eine Eingabe gewartet. Sollte ein Befehl nicht dem vorgegebenen Format entsprechen wird er einfach ignoriert. Case Sensitive Unterstützung bei den Commandos (nicht den Argumenten) ist optional. Nachfolgend die zu implementierenden Befehle:

**create** <room> Erstellt ein Chatroom mit dem Namen vom Argument. Keine weitere Ausgabe. Sollte dieser Chatroom bereits existieren wird "Room already exists!" ausgegeben.

**join** <room> <username> Laesst einen user mit seinem username im Chatroom beitreten. User können auch Chatrooms wechseln in dem dessen username bei einem anderen existierenden Chatroom verwendet wird. Folgende Reihenfolge im Fehlerfall: Existiert der Chatroom noch nicht wird "Room does not exist!" ausgegeben. Existiert dieser username in diesem Chatroom bereits wird "User already joined!" ausgegeben. Wichtig: der username ist über alle Chatrooms eindeutig!

**write** <username> <msg> Schreibt eine Nachricht (msg) als jeweiliger user in seinem beigetretenen Chat. Bei jeder Eingabe dieses Befehls soll zuerst der Chatroomname gefolgt von einer Liste an bisher eingegebenen Nachrichten ausgegeben werden, wobei die letzte Nachricht (inklusive dieser) am Ende angefügt wird. Jede Nachricht soll den jeweiligen Benutzernamen am Anfang beinhalten. Sollte dieser user noch nicht existieren beziehungsweise keinen Chatroom beigetreten sein wird "User not found!" ausgegeben.

**show** Gibt eine Liste (in jeder Zeile einen) an Chatrooms aus die existieren.

Orientieren Sie sich zusätzlich an dem unten vorgegebenen Beispiel. Bei der Eingabe von quit oder EOF soll das Programm beendet werden.

## Beispiel

```
> show
> create room1
> create room1
Room already exists!
> show
room1
> join room1 peter
> join room2 peter
Room does not exist!
> write peter Hallo
room1
peter: Hallo
> join room1 max
> write max Hey
room1
```

---

```
peter: Hallo
max: Hey
> write peter Wie gehts
room1
peter: Hallo
max: Hey
peter: Wie gehts
> create room2
> join room2 peter
> write peter Hallo room2
room2
peter: Hallo room2
> quit
```

## Aufruf

```
./simpleChat
```

## Bewertung

- Dokumentation und Programmierstil: 1 Punkt
- Einhalten der OOP Konzepte: 1 Punkt
- Struktur & Korrektheit: 1 Punkt