

# Dokumentation Projekt Teil 2

## Verwendung des Programms

Dieses Chatprogramm ist ein Konsolenprogramm und muss dementsprechend in der Kommandozeile gestartet werden. Dazu ist der Befehl "java -jar ChatClient.jar" notwendig. Um den Befehl so auszuführen sollte die Kommandozeile bereits in den Ordner geleitet sein in der diese Datei liegt.

Die Jar Datei liegt im Ordner jar im Projektverzeichnis.

Gesteuert werden die Menüs und der sehr große Teil mit Eingaben, die einer der Zahlen gleichkommen, die als ID dienen.

Bedeutet mit 1 kann der erste Punkte gewählt werden und mit 2 der zweite Punkt und so weiter. 99 wird verwendet, um ein Menü zurück zu springen oder das Programm zu beenden.

Dieses Programm wurde unter der Java Version 10.0.2 geschrieben. Die Tests waren mit der Version 10.0.2 und 11.0.1 erfolgreich. Vorherige Java Versionen werden nicht unterstützt!

## Überblick über den Programmcode

Dateien im .java Format

### Package nachrichten

Die Klassen des Packages namens nachrichten sind für das Verschicken, sowie die Anfragen von Nachrichten zuständig.

### **Sender**

Diese Superklasse definiert die abstrakte Methode senden, deren erster Parameter der Empfänger und deren zweiter die zu verschickenden Daten sind. Die Methode senden behandelt die generellen Fehler des Verschickens und ruft die abstrakte Methode sendenZuServer, dessen Aufgabe darin besteht, die Daten zu versenden.

### **BildSender**

Die Subklasse BildSender ist für das Verschicken von Bildern zuständig. Die Daten, die beim Aufrufen der Methode sendenZuServer übergeben werden müssen, wird in dieser Klasse als Dateipfad interpretiert. Deswegen wird dann intern die Datei überprüft, gelesen und dann weggeschickt.

### **TextSender**

Die Klasse TextSender erbt die nötigen Attribute und Methoden der Superklasse Sender. Die überschriebene Methode sendenZuServer ist für das Verschicken eines Textes zuständig. Außerdem gibt es zwei weitere Methoden, die das Empfangen von Nachrichten ermöglichen.

**Package user:****AktuellerBenutzer**

Die Klasse des aktuellen Benutzers wird benötigt, da wir uns zu Beginn zur Verifizierung am Server mit Benutzernamen und Passwort anmelden müssen. Da wir jedes Mal zur Interaktion mit dem Server unseren Benutzernamen sowie das Passwort mitschicken müssen, besitzt diese Klasse jeweils für diese beiden Variablen geeignete getter-Methoden.

**Personen**

Die Superklasse Personen hat die Aufgabe, die Liste aller auf dem Server registrierten Nutzern in einer ArrayList zu speichern, damit die jeweiligen Subklassen nicht jedes Mal diese Liste vom Server beziehen muss. Diese Klasse wurde abstrakt gewählt, da wir die Methode getListe() in den Subklassen unterschiedlich händeln.

**Gruppe**

Um verschiedene Gruppen anlegen zu können braucht jede Gruppe eine eigene Variabel mit ihrem Gruppennamen. Über den Konstruktor kann ein Objekt vom Typ Gruppe erzeugt werden, welches einen Gruppennamen und Eine Liste der Nutzer enthalten muss.

**Nutzer**

Hier wird eine Liste mit allen möglichen Empfängern der Gruppe wsp18 aktualisiert und gespeichert. Die Nutzer können mit ihrer internen ID ausgegeben werden sowie darüber angesprochen werden.

**Package main:****MainKlasse**

Das Package main mit der Klasse MainKlasse beinhaltet die Java-Main Methode, um eine lauffähige Version von dem Chat Programm haben zu können. Diese Klasse instanziert ein Objekt von der Klasse Klient und startet die Methode programm() um das Chat Programm zu starten.

**Package basis:****Klient**

Im Package basis ist die Klasse Klient enthalten. Diese ist das Herz des Programms und von hier aus wird alles andere aufgerufen.

Der Klient verwaltet die Eingaben des Users und die gibt passenden Menüs aus. Der Quellcode ist in 3 Teile unterteilt. Diese wäre der Start mit dem Hauptmenü, das Menü zum Versenden der Nachrichten oder auch Bildnachrichten und als letztes das Menü zum Verwalten der Gruppen.

Mit dem Konstruktor wird der Nutzer gebeten seine Anmeldedaten einzugeben, diese werden in einer Instanz der Klasse AktuellerBenutzer gespeichert. Aus diesen Daten wird seine persönliche Nutzerliste gezogen in eine Instanz der Klasse Nutzer und die Instanzen der Klasse TextSender und BildSender werden neu aufgebaut, um die Anmeldedaten zu bekommen.

Dies lässt sich aus dem Hauptmenü heraus wiederholen.  
Außerdem lässt sich von hier aus auch die letzten 100 Nachrichten anzeigen oder alle Nachrichten neuer als eine bestimmte Nachrichten ID.

Vom Hauptmenü kann dann ins Menü zum Verwalten der Gruppen weitergeleitet werden. Hier können alle Gruppen über ihre Namen aufgelistet werden und außerdem neue erstellt werden.

Für das interne Speichern wird eine ArrayList verwendet die die Klasse Gruppe als Typ beinhaltet. Damit kann dynamisch immer wieder eine neue Gruppe angelegt werden und an das Array angehängen.

Als zweites ist das Menü zum Senden von Nachrichten vorhanden. Hier kann der User Textnachrichten oder Bildnachrichten an einen bestimmten User oder an eine bestimmte Gruppe versenden. Mit Hilfe der Polymorphie reicht ein Aufruf für beide Nachrichtenarten. in der jeweiligen Methode zum Versenden an eine Person oder für die Nachricht an die Gruppe.