Programm : RNP02_Client_Server_Chat
Fach: Rechnernetze WS2017
Autoren: Nelli Welker, Etienne Onasch

Spezifikation aller Klassen und Methoden

pkg: mainClasses:

AbstractClientServerThread extends AbstractWriteThread

->Abstrakte Klasse, die Funktionalitäten von Clients und Servern zusammenfasst

--protected selfMessage(String) : void

->gibt Nachicht mit System.out.println aus.
Für weiter Funktionalität, zB. Weiterleitung an eine Gui, muss diese Methode in einer erbenden Klasse überschrieben werden.

--protected selfMessage(String...) : void

->siehe selfMessage(String). Allerdings sind mehrere Parameter möglich.

--terminate() : void

->schließt alle Streams

AbstractWriteThread extends Thread

->Abstrakte Klasse, die Funktionalität für das Schreiben auf einen Outputstream und das Lesen aus einem Inputstream bereitstellt.

--public write(String) : void

-> schreibt eine Nachicht auf den Outputstream

ClientLoginThread extends AbstractWriteThread

-> Dieser Thread ist dafür zuständig der ServerThread Klasse die Anmeldelogik abzunehmen. Für jeden sich anmeldenden User wird ein neues ClientLoginThread Objekt erzeugt. Ein neuer User muss also nicht warten bis alle vorherigen Anmeldevorgänge abgeschlossen sind.

--private init() : void

->initialisiert alle Streams

--private aksUserName() : String

->ermittelt Usernamen des Clients

--private login() : void

->ist dafür zuständig den Client (unter gewissen Voraussetzungen) endgültig einzuloggen.

--public run() : void

->startet die login() Methode.

ClientThread extends AbstractClientServerThread

-> Diese Klasse repräsentiert einen Client.

--private init() : void

-> Initialisierung aller Streams

--public setGui(ClientGui) : void

-> dem ClientThread wird eine Gui übergeben, damit beim Client eintreffende Nachichten der GUI übermittelt werden können.

--protected selfMessage(String) : void

-> Überschreibt die Methode in AbstractClientServerThread.
Wurde eine GUI übergeben, wird ihr die Nachicht übergeben.
Ansonst wird die selfMessage(String) Methode der Superklasse ausgeführt.

--public run() : void

-> Hält den Client am Laufen. Ausserdem werden hier alle eintreffenden, den Client betreffenden Befehle bearbeitet.

Helpers

--public static validateName(String) : boolean

-> Diese Methode stellt fest, ob ein Username den Server Regularien entspricht.

ServerThread extends AbstractClientServerThread

->Diese Klasse repräsentiert einen Server.

--public disconnect() : void

->Diese Methode beendet den Client.

--pkg private static addToLog(String) : void

->Diese Methode schreibt eine Nachicht in den Server Log.

--pkg private static printLog() : void

->gibt den Server Log aus.

--public run() : void

-> Diese Methode hält den Server am Laufen. Für jeden Neuen User wird ein neuer Socket mit der Methode ServerSocket.accept() erzeugt.
Dieser wird einem neuen ClientLoginThread übergeben.

ConsoleReader extends Thread (Inner class of ServerThread)

->Innere Klasse von ServerThread. Ein ConsoleReader ist dafür zuständig Befehle eines Serveradmins von der Konsole einzulesen.

--public run() : void

->Hier werden alle Serveradmin Befehle abgehandelt.

UserThread extends AbstractWriteThread

->War ein login in einem ClientLoginThread erfolgreich, wird ein neuer UserThread erstellt. Hier wird dann im wesentlichen der komplette Chat inkl. aller User Kommandos a bgehandelt.

--private init() : void

-> Initialisierung aller Streams

--public getUserName() : String

-> gibt den Usernamen zurück.

--public getRoom() : String

->gibt den Raum zurück in dem sich der User befindet.

--public setRoom(String) : void

->stellt einen neuen Raum ein.

--public kick() : void

-> "kicked" den User vom Server. Beendet den UserThread und schickt einen Befehl an den

ClientThread, dass sich dieser beenden soll.

--public synchronized write(String) : void
-> Überschreibt die write Methode in AbstactWriteThread. Schreibt die Nachicht in den Userlog und leitet sie danach an die write(String) Methode der Superklasse weiter.
--public run() : void
->Hält den UserThread am Laufen. Ausserdem werden hier alle User Kommandos abgehandelt.

pkg: utils:

Commands
-> Diese Utility Klasse enthält alle Kommandos als statische Konstanten.
Constants
-> Diese Utility Klasse enthält verschiedene statische numerische Konstanten.

pkg: gui:

ClientGui
->Die GUI, die dem User während des Chats angezeigt wird.
Hier können der sichtbare Chatverlauf betrachtet und neue Nachichten geschrieben werden.
LoginGui
->Die GUI, die dem User vor dem eigentlich Chatfenster angezeigt wird.
Der User gibt hier seinen Usernamen, die Serveradresse und den Servernamen ein.
Positionings
->Utility Klasse, die statische Konstanten und Methoden für die richtige Darstellungsweise der GUIs bereitstellt.