

Teamname: Bernie und Ert

Vorgehensweise und Entscheidungsfindung

Wir haben ihr Beispiel TCPServer und TCPClient als Vorlage genommen. In den Server hat sich bei Ihnen scheinbar ein Fehler eingeschlichen. In Zeile 33 rufen Sie `this.run()`; auf, obwohl sie in Zeile 16 einen Executor nutzen, der ebenfalls ja die `run()` Methode aufruft. Dies führt zu interessanten `IOExceptions`, wenn man mit dem TCPClient darauf zugreift.

Die Enums haben wir nicht in eigene Dateien ausgelagert, sondern direkt im Server integriert.

Über die `main`-Methode im TCPServer wird der Server auf einem vorgegebenen Port gestartet, alternativ könnte man den Übergabeparameter `args[]` nutzen, um bspw. Bindadresse und einen beliebigen Port zu übergeben.

Um gezielt den STATE des Server zu ändern, haben wir einen weiteren Thread/Runnable erzeugt, über den wir, während der Server rennt, über Konsoleneingaben den Status ändern können.

In Der Klasse Connection haben wir keine weiteren, zusätzlichen Status oder Modi eingeführt, bei der zufälligen Wahl des Modes bedienen wir uns eines Objekts der Klasse Random ☺

Der Client ist extrem einfach gestrickt, er wird gestartet und verbindet sich mit dem Server, anschließend kann man einen String auf der Konsole eingeben und erhält die Antwort vom Server. Der Socket wird geschlossen und der Client beendet sich.

Man könnte den Client auch in einer Endlosschleife laufen lassen und mehrere Instanzen davon starten, um den Server mit neuen Verbindungsthreads zu bombardieren.