

# Rechnernetze und verteilte Systeme

## Übungsblatt 4

**Ausgabe:** 31. Oktober, **Abgabe:** 7. November, 10 Uhr

### Programmieraufgabe 4.1 (10 Punkte)

Implementieren Sie in Java 1.6 die fehlenden Teile einer einfachen Client/Server-Anwendung und kommentieren Sie ihren Code sinnvoll! Auf der Übungsseite finden Sie entsprechende Vorlagen (Server.java, Client.java, DataProvider.java, DataReceiver.java, IClient.java, AServer.java, Util.java, Test.java).<sup>1</sup>

Die Anwendung soll folgende Eigenschaften erfüllen:

- Der Server soll gestartet werden und auf eingehende Client-Verbindungen warten.
- Eine festgelegte Anzahl Clients (siehe Test.java) soll gestartet werden. Sie stellen eine Verbindung zum Server her und übermitteln ihre Id.
- Die Clients empfangen Nachrichten vom Server und geben diese aus (Standardausgabe), bis dieser die Verbindung beendet.
- Der Server versucht eine festgelegte Anzahl Nachrichten an alle verbundenen Clients zu senden und schaltet sich danach ab.
- Wenn ein Client versucht eine zweite Verbindung zum Server aufzubauen, oder ein Client beim Verbindungsaufbau eine ID eines anderen Client benutzt, welcher sich schon mit dem Server verbunden hat, soll der Verbindungsaufbau mit einer entsprechenden Fehlermeldung abgebrochen werden.

Sie müssen lediglich die Klassen Client.java und Server.java implementieren, die für die eigentliche Netzwerkkommunikation verantwortlich sind. Der Aufbau der Klassen ist bereits vorgegeben. Die anderen Klassen dürfen nicht verändert werden. Die Klasse Test.java startet den Server sowie einige Clients, so dass Sie die Funktionsweise ihrer Implementierung überprüfen können.

Nützliche Klassen (nähere Info siehe API-Dokumentation):

- `java.net.ServerSocket` (zum Erstellen eines „lauschenden“ Sockets, statt `socket.bind()`),  
`java.net.Socket`
- `java.io.InputStream`, `java.io.OutputStream`, `java.io.BufferedReader`, `java.io.BufferedWriter`

---

<sup>1</sup>Bitte geben Sie nur die veränderten Dateien ab.

**Hinweis:** In der Musterlösung ist keine Methode länger als 15 Zeilen (die meisten sind deutlich kürzer). Die Lösung für jede Methode ist daher nicht sehr aufwändig. Es geht hier nicht darum kürzeste Lösungen zu entwickeln. Eine Abgabe, die nicht kompiliert, wird mit 0 Punkten bewertet.

### **Achtung!**

**Die Abgabe der Aufgabe erfolgt über ASSESS unter  
<http://ess.cs.uni-dortmund.de/ASSESS/>.**

**Die Abgabe muss in 3er oder 4er Gruppen erfolgen.**