
Protokoll

Arbeitsauftrag 1:

- Aufgabenstellung:
 - Erstelle eine Client/Server Anwendung in Java, die für Clients einfache Berechnungen durchführt. Dabei sollte die Addition, Subtraktion, Multiplikation und Division mit Integer-Zahlen ermöglicht werden. Der Client wählt eine Operation aus und schickt dem Server eine Anfrage. Der Server führt diese Anfrage aus und schickt dem Client das Ergebnis zurück.
- Funktionsweise:
 - Nachdem man den Server und den Client gestartet hat, muss man beim Client die IP des Servers und die Portnummer eingeben. Man kann einfach die Entertaste drücken, wenn man IP und Portnummer nicht verändert. Danach kann man den Benutzernamen eingeben, den man dann auch in der Konsole des Servers sehen kann. Nach dem Ganzen, kann man dann den Taschenrechner betätigen, indem man die erste Zahl, die zweite Zahl und die Rechenoperation in der Konsole des Clients eingibt. Das Ergebnis kann dann in der Konsole des Servers sehen.

```
"C:\Program Files (x86)\Java\JDK\Java\jdk8\bin\java" ...  
Verbindung zum Server...
```

```
Läuft der Server auf dem Pc, Entertaste drücken oder IP des Servers eingeben:  
Port Nummer eingeben oder Entertaste drücken:  
Benutzernamen eingeben: Hannes
```

- IP, Portnummer und Benutzernamen eingeben beim Client.

```
Verbindung erfolgreich!  
Geben Sie bitte die erste Zahl ein:4  
-----  
Du hast 4 eingegeben.  
-----  
Geben Sie bitte die zweite Zahl ein:5  
-----  
Du hast 5 eingegeben.  
-----  
Geben Sie bitte die Rechenoperation ein: +  
-----  
Du hast + eingegeben.  
-----  
-----Ergebnis findet Ihr beim Server!-----  
-----
```

- Taschenrechner im Client.

```
"C:\Program Files (x86)\Java\JDK\Java\jdk8\bin\java" ...  
Neuer Client verbunden:Hannes
```

Dein Ergebnis:9

I

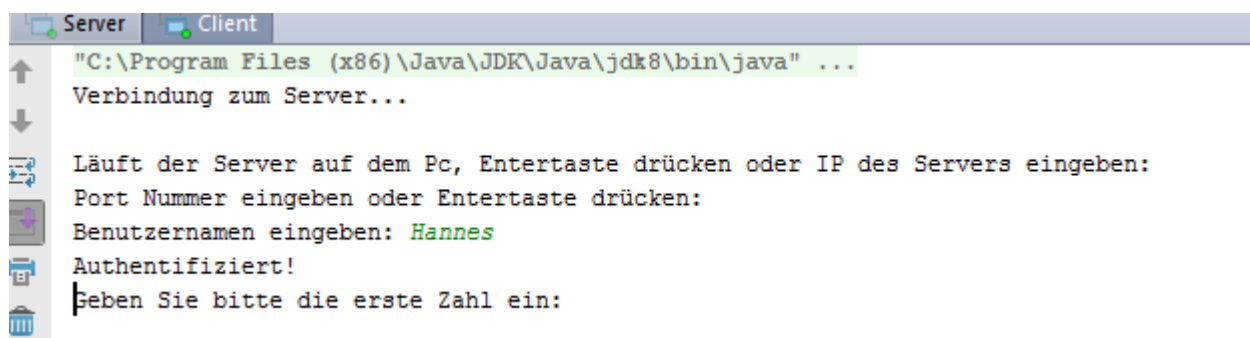
➤ Benutzername und Ergebnis beim Server.

- Note: 9
- Probleme:
 - Ich hatte keine Probleme, diese Aufgabe zu lösen.

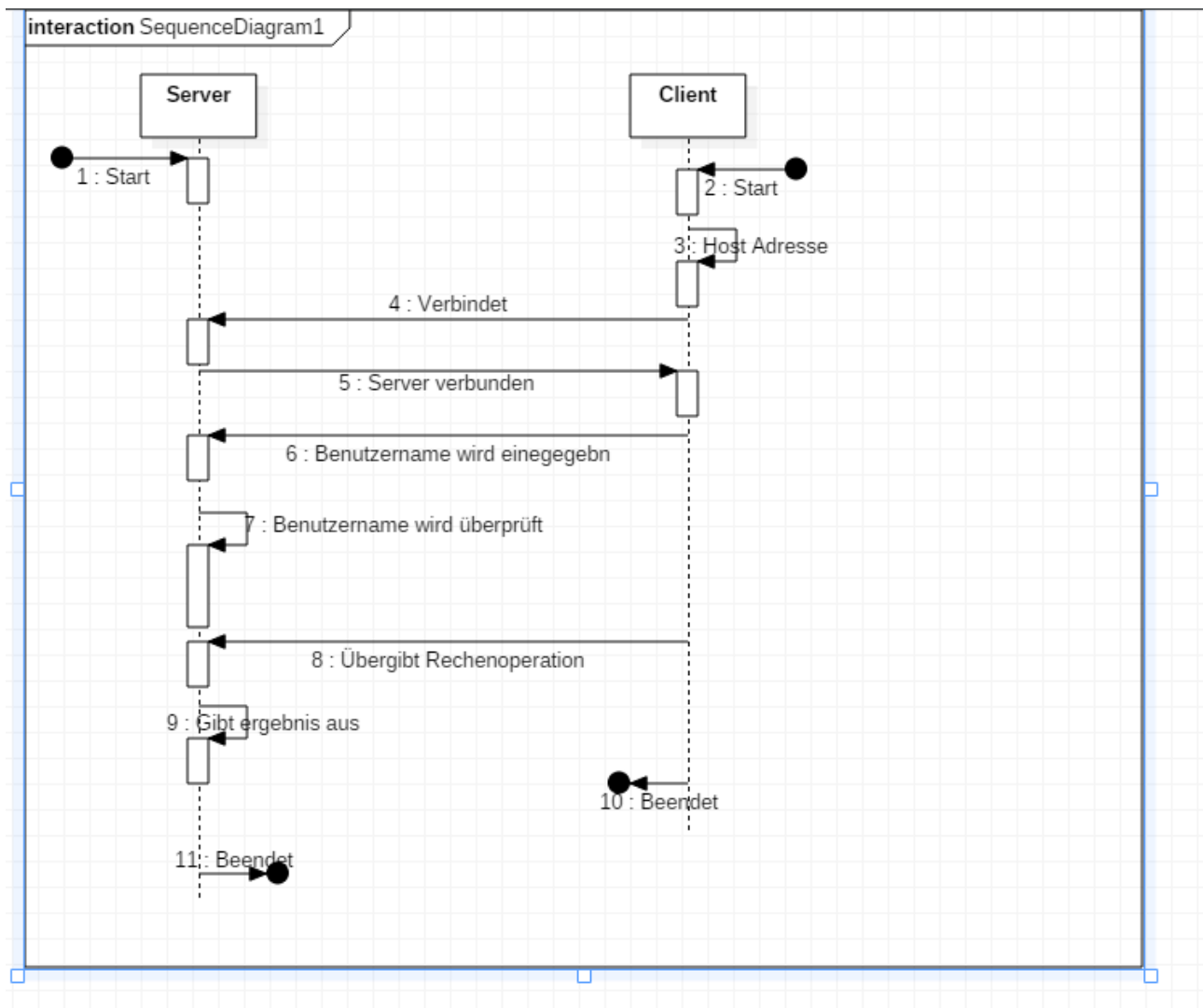
Arbeitsauftrag 2:

- Aufgabenstellung:
 - Aufgabe 1 soll mit deiner Benutzerauthentifizierung erweitert werden. Der Client muss sich vor der Anfrage authentifizieren. Nur bei erfolgter Authentifizierung ist es dem Client möglich den Dienst zu nutzen. Überlege dir hierfür einen Ansatz und erweitere das Protokoll von Aufgabe1 dementsprechend.
- Funktionsweise:
 - Zum Verbinden bleibt es gleich, wie beim ersten Arbeitsauftrag. Dazu kommt, dass sich nur 3 Benutzer anmelden können, um den Taschenrechner zu betätigen, nämlich: Hannes, Maximilian und Ulrich.

```
-----  
ArrayList<String> AuUser = new ArrayList<>(); //AuUser -> Authentifizierte User  
  
AuUser.add("Hannes"); //Authentifizierte User  
AuUser.add("Maximilian");  
AuUser.add("Ulrich");
```



➤ Ausgabe der Konsole.



➤ Zustandsdiagramm

- Note: 7/8
- Probleme:
 - Ich wusste am Anfang nicht wirklich, wie ich das Programmieren soll, aber habe es dann mithilfe von Klassenkameraden geschafft. Ein anderes Problem war, dass ich einen NullPointerException im Programm hatte, aber später auch gelöst habe.

Arbeitsauftrag 3:

- Aufgabenstellung:
 - Erstelle eine Client/Server Anwendung in Java, die den Download eine Webseite ermöglicht. Der Client spezifiziert die URL und sende sie an den Server. Der Server downloadet die Seite, speichert sie als html-File lokal ab und schickt sie an den Client. Welchen Vorteil hat dieses Vorgehen und welche Aufgabe (als Netzwerkkomponente) erfüllt der Server?
- Funktionsweise:
 - Nachdem sich der Client wie üblich mit dem Server verbunden hat, kann man dann eine beliebige Webseite in der Konsole eingeben. Man muss aber davor „<https://>“ eingeben, damit es funktioniert. Im Ordner des Arbeitsauftrages 3, befindet sich eine readMe - Datei, wo beschrieben wird woher ich die Datei Downloader und Server-exit her habe.

Läuft der Server auf dem Pc, Entertaste drücken oder IP des Servers eingeben:

Port Nummer eingeben oder Entertaste drücken:

Benutzernamen eingeben: *Hannes*

Verbindung erfolgreich!
Url eingeben mit <https://>

<https://www.google.com>

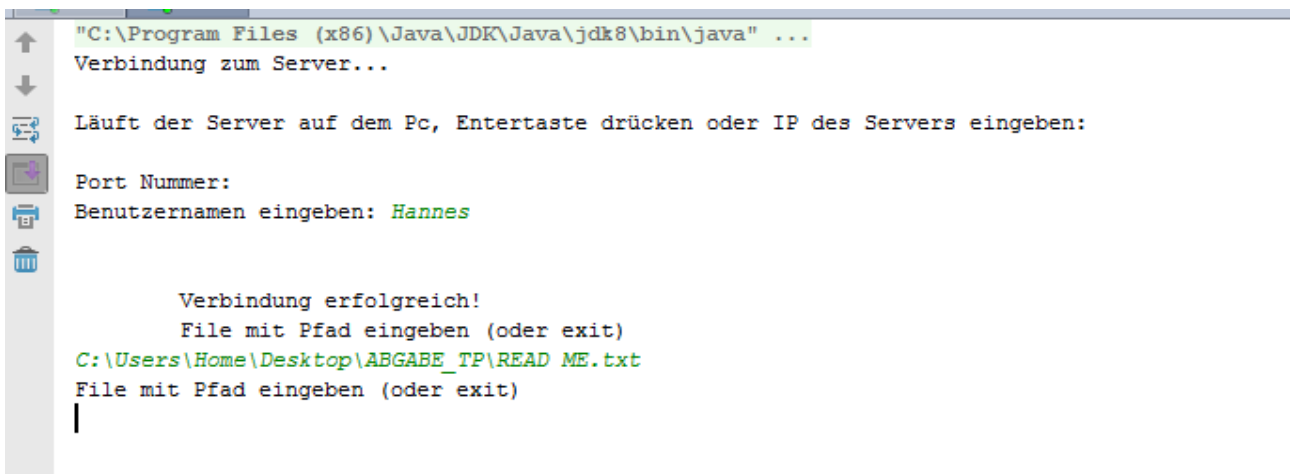
|

➤ Url eingeben.

- Note: 7
- Probleme:
 - Ich hatte einen Fehler, den ich nicht finden konnte, aber schlussendlich lösen konnte. Der Fehler war, dass ich ein ObjectOutputStream zu viel hatte und es mir nicht aufgefallen.

Arbeitsauftrag 5:

- Aufgabenstellung:
 - Realisiere eine Client/Server Anwendung in Java, die einen File-Transfer vom Server zum Client ermöglicht. Der Client kann irgendeinen Pfad zu einer Datei oder einem Ordner angeben der dann vom Server zum Client übertragen wird. Dabei sollen auch parallele Downloads möglich sein.
- Funktionsweise:
 - Die Verbindungsart zwischen Client und Server ändert sich zum ersten Mal, denn der Client kommuniziert jetzt mit der Klasse ServerThread.java. Die Anmeldung bleibt aber immer gleich. Nachdem man sich „angemeldet“ hat, kann man den Pfad der Datei eingeben. Wie auch im Arbeitsauftrag 3, verwende ich den Downloader, denn ich vom Internet habe.



```
"C:\Program Files (x86)\Java\JDK\Java\jdk8\bin\java" ...
Verbindung zum Server...

Läuft der Server auf dem Pc, Entertaste drücken oder IP des Servers eingeben:

Port Nummer:
Benutzernamen eingeben: Hannes

Verbindung erfolgreich!
File mit Pfad eingeben (oder exit)
C:\Users\Home\Desktop\ABGABE_TP\READ ME.txt
File mit Pfad eingeben (oder exit)
|
```

- Note: 7
- Probleme:
 - Im Programm gibt's ein Problem, nämlich kopiert das Programm nicht den Inhalt der Dateien, es übernimmt nur das Dateiformat (.txt, .jpg). Ich hatte dann leider zu wenig Zeit um mich mit diesen Problem auseinander zu setzten.

Arbeitsauftrag 6:

- Aufgabenstellung:
 - Realisiere eine Client/Server Anwendung in Java. Der Server stellt zwei Funktionen (getRandom() und getTime()) zur Verfügung, die eine lange Berechnung simulieren sollen. Für diese Aufgabe sollten zwei verschiedene Serverversionen erstellt werden. Eine sequenzielle Version, die nur einen Thread verwendet und somit die Clients nacheinander abarbeitet und eine parallele Version die mehrere Threads verwendet und die Clients parallel abarbeitet. Teste beide Server und dokumentiere deine Beobachtungen. Welche Einschränkungen und welches Leistungsvermögen haben die Server?
- Funktionsweise:
 - „Anmeldung“ bleibt wie immer gleich, nur dass die Klasse Server.java nicht ausführbar ist. Eine Klasse Start.java kommt dazu, was den Server startet. Zum Arbeitsauftrag, habe ich 2 verschiedene Versionen programmiert: Eine mit sequentieller Thread und eine mit paralleler Threads. Der Unterschied ist, dass beim sequentiellen die Threads hintereinander ausgeführt werden und beim parallelen parallel. Der Parallele funktioniert besser, da man nicht warten muss bis der vorherige Thread beendet wurde.

```
Läuft der Server auf dem Pc, Entertaste drücken oder IP des Servers eingeben:  
Port Nummer eingeben oder Entertaste drücken:  
Für getRandom() [1] oder für getTime() [2]  
1  
Wert 15.0  
Für getRandom() [1] oder für getTime() [2]  
2  
Wert 2017/03/05/11:24:41  
Für getRandom() [1] oder für getTime() [2]  
|
```

➤ Beide Programme sehen identisch aus.

- Note: 8 und 8
- Probleme:
 - Ich bin nicht wirklich auf Probleme gestoßen.

Arbeitsauftrag 7:

- Aufgabenstellung:
 - Erweitere Aufgabe 6 so, dass der Server einen sauberen Shutdown durchführt. D.h. das alle bereits gestarteten Anfragen sollten korrekt beendet, aber keine neue mehr gestartet werden. Alle auftretenden Exceptions sollten dabei korrekt behandelt werden. Verwende dabei das Interface ExecutorService.
- Funktionsweise:
 - Dieses Programm unterscheidet sich nur vom Arbeitsauftrag 6, dass es eine Shutdownfunktion hat. Man schreibt „Shutdown“ in der Konsole, und der Client beendet sich.

Verbindung zum Server...

Läuft der Server auf dem Pc, Entertaste drücken oder IP des Servers eingeben:

Port Nummer eingeben oder Entertaste drücken:

getRandom() [1] oder getTime() [2] oder Shutdown

Shutdown

Programm beendet

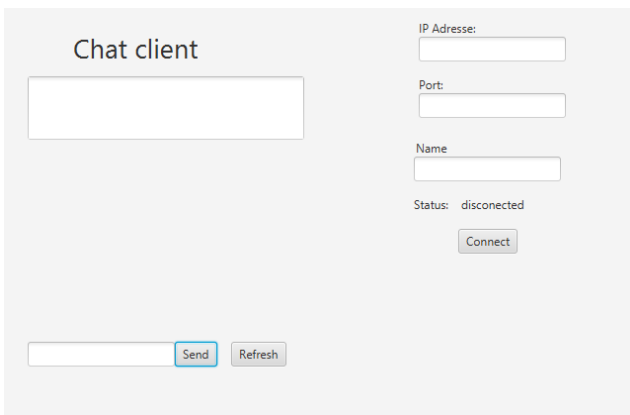
Process finished with exit code 0

➤ Der Client beendet sich.

- Note: 8
- Probleme:
 - Ich hatte nicht wirklich Probleme.

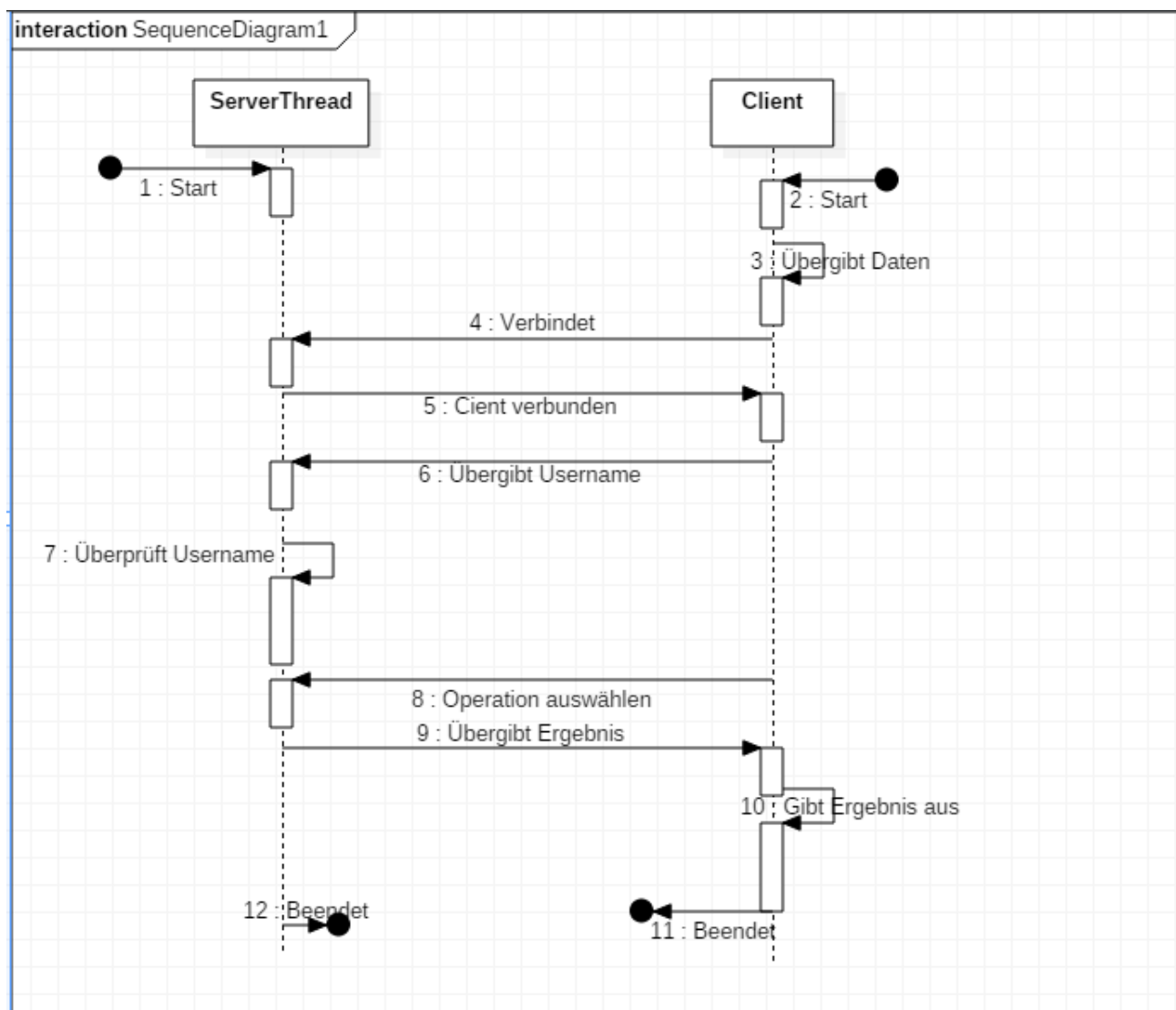
Arbeitsauftrag 8:

- Aufgabenstellung:
 - Realisiere eine Client/Server Anwendung in Java, die einen Chat zwischen mehreren Clients ermöglicht. Verwende dabei UDP-Datagramme.
- Funktionsweise:
 - Da es mir nicht zeitlich nicht gereicht hat, habe ich bei diesem Arbeitsauftrag ein Programm hergenommen, dass ich schon in der Vergangenheit programmiert habe. Dieses Programm hat eine graphische Oberfläche und man muss die Main 2 mal starten, damit man kommunizieren kann.



➔ Hier verändert sich zu den anderen Programmen die Portnummer, sie lautet 6666.

- Note: 8
- Probleme:
 - Ich weiß nicht ganz genau, ob das vom Arbeitsauftrag gemeint ist, aber ich denke das sollte passen.

Zustandsdiagramm:

- Für die meisten Programme gilt dieser Zustandsprotokoll

Fehlerbericht:

- Arbeitsauftrag 4:
 - Ich habe diesen Arbeitsauftrag leider nicht lösen können, da ich auf mehreren Problemen gestoßen bin, unter anderem habe ich zu wenige Kenntnisse um eine graphische Oberfläche für das Spiel zu programmieren.
- Arbeitsauftrag 9:
 - Ich hatte nicht mehr genügend Zeit und zu wenige C Kenntnisse um diesen Arbeitsauftrag zu lösen.
- Arbeitsauftrag 10:
 - Ich hatte nicht mehr genügend Zeit und zu wenige C Kenntnisse um diesen Arbeitsauftrag zu lösen.
- Arbeitsauftrag 11:
 - Ich hatte nicht mehr genügend Zeit und zu wenige C Kenntnisse um diesen Arbeitsauftrag zu lösen.

- Arbeitsauftrag 12:
 - Ich hatte nicht mehr genügend Zeit und zu wenige C Kenntnisse um diesen Arbeitsauftrag zu lösen.