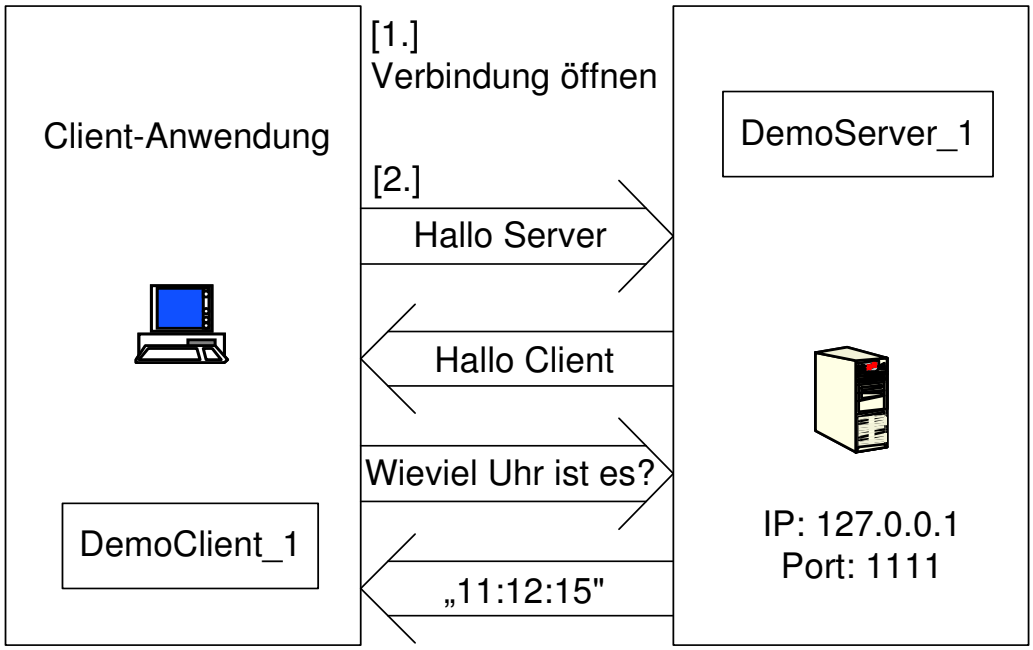
Beispiel 1: Ein einfacher Server

In diesem Beispiel soll ein einfacher Server nach folgendem Schema realisiert werden:



Anwendung:

DemoServer_1:
- zuerst auf der eigenen Maschine starten
- der Server wartet auf Anfragen
- wenn eine Anfrage hereinkommt, beantwortet er sie
- ist die Anfrage abgearbeitet, wartet er auf die nächste

DemoClient_1:
- starten
- macht eine Anfrage an der Server gemäss obigem Kommunikationsschema
- hat er die Antwort erhalten, wird das Programm beendet

Verzeichnis-Struktur:
..\Uebungen\SocketKommunikation // Arbeitsverzeichnis
..\Uebungen\SocketKommunikation\src\beispiel1 // Quellcode
..\Uebungen\SocketKommunikation\class\beispiel1// Bytecode

Aufgaben:

1. Überlegen Sie sich, wie sie die Anwendung realisieren wollen. Erstellen Sie dazu ein Klassendiagramm.
2. Realisieren Sie zuerst den Client. Testen Sie ihn gegen die zur Verfügung gestellte Serverklasse.
3. Was passiert, wenn Sie im DemoClient_1 die Befehle try{...}catch herausnehmen und eine Port-Nummer angeben, die nicht existiert?
4. Realisieren Sie den Server selber.
5. Schreiben Sie einen DemoClient_1b, der den Server eines Ihrer Kollegen anfragt und testen Sie ihn.
6. Sobald der Server auf Ihrer Maschine gestartet ist, läuft er immer. Kontrollieren Sie, wie viel Resourcen ihrer Maschine er belegt.
7. Zeichnen Sie mit dem Packetizer auf, was bei einer Anfrage über das Netz geht.

Kompilation und Laufenlassen

Working directory: ..\Uebungen\SocketKommunikation

a) Von Hand:

```
>javac -classpath class -d class src\beispiel1\*.java
>java -classpath class beispiel1.DemoServer_1
```

b) im Eclipse