

Client-/Server-Anwendung “Wetterdienst” mit StreamSockets

Sebastian Schmid S0543196

Übung 2 Verteilte Systeme

Aufgabenstellung

Zu erstellen waren ein Client-Server Protokoll mit Implementierung.,

Dabei sollte vom Client ein Datum an den Server gesendet werden. Der Server sollte Temperaturwerte zum gegebenen Datum in einer CSV-Datei nachschlagen, min, max und avg berechnen und an den Client zurücksenden.

Das gesendete solle dann vom Client im Terminal angezeigt werden.

Lösung

```
1 private void cliUI() throws IOException {
2     Boolean done = false;
3     Scanner reader;
4
5     reader = new Scanner(System.in);
6     System.out.print("Bitte Datum Eingeben [YYYY-MM-DD] \n\t>>> ");
7     String userInput = reader.next();
8
9     this.send(userinput);
10    System.out.println(this.rcv());
11 }
```

```
1 public void run() throws IOException {
2     while (true) {
3         Socket connectionSocket = welcomeSocket.accept();
4
5         String query = this.rcv(connectionSocket);
6         String answer = this.generateRcvData(query);
7         this.send(answer, connectionSocket);
8         connectionSocket.close();
9     }
10 }
```

Sockets

- Client

```
1 import java.net.Socket;
2
3 /* Aufbau */
4 clientSocket = new Socket("localhost", 6789);
5
6 /* Senden */
7 outToServer.writeBytes(msg + '\n');
8
9 /* Empfangen */
10 inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
```

- Client verwendet java.net.Socket
- Senden und Empfangen als BitStreams.

- Server

```
1 import java.net.ServerSocket;
2 import java.net.Socket;
3
4 /* Aufbau */
5 welcomeSocket = new ServerSocket(6789);
6
7 /* Lauschen */
8 Socket connectionSocket = welcomeSocket.accept();
9
10 /* Empfangen */
11 BufferedReader inFromClient =
12     new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));
13
14 /* Senden */
15 DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());
16 outToClient.writeBytes(msg);
```

- Server verwendet java.net.ServerSocket (Lauschen) und java.net.Socket (Senden)
 - Da TCP muss der Datenstrom vom Server beendet werden (Socket.close())
 - Auch hier muss die Umwandlung zwischen Bitstreams und Strings erfolgen
-

CSV

```
1 private void filter(String l, String pattern) {
2     if (l.contains(pattern)) {
3         filteredLines.add(l.split(","));
4     }
5 }
```

- Hier erfolgt ein einfaches Überführen der passenden Zeilen in eine neue Datenstruktur
-

Protokolle

- Übertragungsprotokoll → Mehrzeiliger Stringblock
- Datumskonvention → YYYY-MM-DD