Übungsblatt 7 (Musterlösung)

Das Protokoll eines Witze-Dienstes

Sie haben in der Vorlesung das Protokoll eines Witze-Servers entwickelt (siehe Protokoll.pdf). Dieses Protokoll soll als TCP-Dienst implementiert werden.

1. Implementierung eines TCP-Witze-Servers

Erzeugen Sie im bereits vorhandenen Package „sockets“ eine Klasse JokeServer. Übernehmen sie den Code aus ihrem SimpleServerTCP (oder aus der Musterlösung).

a) Implementieren Sie nun das Protokoll, das wir in der Vorlesung entwickelt haben, indem Sie Ihren SimpleServer entsprechend erweitern.
b) Der Server soll auf dem Port 6666 laufen.
c) Testen Sie Ihren Server mit Telnet. Benutzen Sie dazu die Software Putty.
   - Falls Putty nicht installiert ist, können Sie eine „putty.exe“ von http://www.putty.org/ herunterladen.
     - Einstellungen:
       - “RAW” (nicht Telnet)
       - Ihre eigene IP ist 127.0.0.1 oder “localhost”
       - Port = 5555
d) Testen Sie den Server eines anderen Studenten mit Telnet
   - Die IP eines Rechners kann man mit dem Befehl „ipconfig“ ermitteln (unter Linux/Mac: „ifconfig“)

Tipps:

- Der Sever sollte auf der Kommandozeile mit „`System.out.printf(…);`“ ausgeben was alles passiert, z.B.:
  - Verbunden mit Client
  - Empfangen: Hallo
  - Gesendet: Hallo
  - Empfangen: . . .
- Man kann einen empfangenen String mit dem Befehl „`split(";");`“ zerteilen. Im Beispiel wird der String an allen Stellen geteilt die ein Semikolon enthalten. Der Befehl gibt ein Array von Strings zurück. Beispielsweise könnte man sich zur Vereinfachung folgende Methode definieren (Achtung, sie müssen alle Methoden als „static“ deklarieren, da wir hier nicht objektorientiert arbeiten wollen):

```
public static String[] getLine(Scanner inFromClient) {
	String line = inFromClient.nextLine();
	System.out.printf("Empfangen: %s\n", line);
	return line.split(";");
}
```

- Definieren sie sich ein Witze und Spruch-Array (auch static):

```
public static String[] witz = { "Erster Witz", "Zweiter Witz", Dritter Witz" };
```

- Lagern Sie ggf. auch die Fehlermeldungen in eine (statische) Methode aus.

Lösung

2. Implementierung eines TCP-Witze-Clients

Erzeugen Sie im bereits vorhandenen Package „sockets“ eine Klasse JokeClient. Übernehmen sie den Code aus ihrem SimpleClientTCP (oder aus der Musterlösung).

a) Implementieren Sie nun das Protokoll, das wir in der Vorlesung entwickelt haben, indem Sie Ihren SimpleClient entsprechend erweitern.
b) Der Client soll für den Benutzer folgende Features anbieten:
   - Der Benutzer wird gefragt, mit welchem Server er sich verbinden möchte (IP-Adresse).
   - Der Benutzer wird gefragt, ob er einen Witz oder einen Spruch möchte, oder ob er den Client beenden will. Dabei soll man hintereinander mehrere Witze/Sprüche vom selben Server abrufen können.
c) Testen Sie Ihren Client mit Ihrem eigenen Server.
d) Testen Sie Ihren Client mit den Servern anderer Studenten.

Tipps:

- Die selben Tipps wie beim Server.

Lösung

```
package sockets;

import java.io.*;
import java.net.*;
import java.util.*;

public class JokeClient {
	public static void main(String args[]) throws Exception {
		System.out.printf("Client gestartet\n");

		Scanner inFromUser = new Scanner(System.in);
		System.out.printf("Bitte IP-Adresse des Servers eingeben: ");
		String serverIP = inFromUser.nextLine();
		Socket socket = new Socket(serverIP, 5555);
		protocol(socket, inFromUser);
		System.out.printf("Client beendet\n");
		inFromUser.close();
	}

	public static String[] getLine(Scanner inFromServer) {
		String line = inFromServer.nextLine();
		System.out.printf("Empfangen: %s\n", line);
		return line.split(" ");
	}

	public static void protocol(Socket socket, Scanner inFromUser) throws Exception {
		PrintStream outToServer = new PrintStream(socket.getOutputStream());
		Scanner inFromServer = new Scanner(socket.getInputStream());

		String[] header;
		String data;
		int witzNummer = 1;
		int spruchNummer = 1;

		// Begrüßung
		outToServer.printf("Hello 0 Andreas\r\n");
		header = getLine(inFromServer);
```

```
                while (true) {
                        System.out.printf("Witz (w), Spruch (s), Spruch des Tages (t) oder Ende (e): ");
                        String auswahl = inFromUser.nextLine();

                        if (auswahl.equals("w")) {
                                outToServer.printf("Witz %d /\r\n", witzNummer++);
                        } else if (auswahl.equals("s")) {
                                outToServer.printf("Spruch %d /\r\n", spruchNummer++);
                        } else if (auswahl.equals("t")) {
                                outToServer.printf("Sdt 0 /\r\n");
                        } else if (auswahl.equals("e")) {
                                break;
                        } else {
                                continue;
                        }
                        header = getLine(inFromServer);
                        int lines = Integer.parseInt(header[1]);
                        for (int i = 1; i <= lines; i++) {
                                data = inFromServer.nextLine();
                                System.out.printf("Empfangen: %s\n", data);
                        }
                }

                // Verabschiedung
                outToServer.printf("BYE 0 /\r\n");
                header =

                                getLine(inFromServer);

                outToServer.close();
                inFromServer.close();
                socket.close();
        }
}
```

```
package sockets;

import java.io.*;
import java.net.*;
import java.util.*;

public class JokeServer {

        public static String[] witz = { "Was ist gelb und kann nicht schwimmen?\r\nEin Bagger!",
                        "Das ist ein personalisierter Witz fuer _",
                        "Was passiert, wenn man cola und bier zusammen trinkt? Man colabiert.",
                        "Was ist suess und klebrig und schwingt von Baum zu Baum…? ein Tarzipan.",
                        "Sagt ein Skelett in der Kneipe: \"Bitte ein Bier und einen Aufwischlappen\"",
                        "Geht ein Skelett zum Arzt. Sagt der Arzt: \"Sie kommen aber reichlich spät.\"",
                        "Fragt die eine Gans die andere: \"Glaubst Du an ein Leben nach Weihnachten?\"",
                        "Treffen sich zwei Hellseher. Meint der Erste:\r\n\"Dir geht’s gut und wie geht’s
mir?\"",
                        "Steht ein Schwein vor einer Steckdose: \"Na Kumpel, wer hat dich denn
eingemauert?\"",
                        "Was sagt die Holzwurmmutter abends zu ihren Kindern?\r\n\"Husch, husch ins
Brettchen\"!",
                        "Ein Beamter zum anderen: Ich weiß gar nicht, was die Leute gegen uns haben, wir tun
doch nichts!",
                        "Sie: \"Du bist immer anderer Meinung als ich!\"\r\nEr: \"Zum Glück, sonst hätten wir
ja beide Unrecht.\"",
                        "Der Unteroffizier zu den Rekruten: \"Männer, ihr müsst dem Feind immer fest ins Auge
sehen – Krause was starren Sie mich so an?\"" };

        public static String[] spruch = { "Was du heute kannst besorgen, das verschiebe nicht auf morgen",
                        "Wer früher stirbt ist länger tot", "Das ist ein personalisierter Spruch fuer _",
                        "Es gibt viel zu tun, warten wirs ab!", "Früh aufstehen ist der erste Schritt in die
falsche Richtung!",
                        "Wer seine Träume verwirklichen will, muss erstmal aufwachen!",
                        "Wer morgens zerknittert aufsteht, hat am Tag die besten Entfaltungsmöglichkeiten!" };

        public static void main(String args[]) throws Exception {
```

```
                System.out.printf("Server gestartet\n");
                ServerSocket serverSocket = new ServerSocket(5555);

                while (true) {
                        Socket socket = serverSocket.accept();
                        System.out.println("Verbunden mit Client!");
                        protocol(socket);
                }
        }

        public static void fehler(Scanner inFromClient, PrintStream outToClient, Socket socket, String[] line)
{
                System.out.printf("Client spricht falsches Protokoll: %s\n", Arrays.deepToString(line));
                outToClient.printf("FEHLER;Falsches Protokoll;\r\n");
                inFromClient.close();
                outToClient.close();
        }

        public static String[] getLine(Scanner inFromClient) {
                String line = inFromClient.nextLine();
                System.out.printf("Empfangen: %s\n", line);
                return line.split(" ");
        }

        public static int countLines(String s) {
                int lines = 1;
                for (int i = 0; i < s.length(); i++) {
                        if (s.charAt(i) == '\n') {
                                lines++;
                        }
                }
                System.out.println("Lines: " + lines);
                return lines;
        }

        public static void protocol(Socket socket) throws Exception {
                Scanner inFromClient = new Scanner(socket.getInputStream());
                PrintStream outToClient = new PrintStream(socket.getOutputStream());

                String line[];
                String clientName;

                line = getLine(inFromClient);
                if (line.length < 3 || !line[0].equals("Hello")) {
                        fehler(inFromClient, outToClient, socket, line);
                        return;
                }

                outToClient.printf("Hello 0 /\r\n");
                clientName = line[2];
                while (true) {
                        line = getLine(inFromClient);
                        if (line.length < 3 || (!line[0].equals("Witz") && !line[0].equals("Spruch") &&
!line[0].equals("Sdt")
                                        && !line[0].equals("Bye"))) {
                                fehler(inFromClient, outToClient, socket, line);
                                return;
                        }
                        if (line[0].equals("Bye")) {
                                outToClient.printf("Bye 0 /\r\n");
                                outToClient.close();
                                inFromClient.close();
                                socket.close();
                                return;
                        } else {
                                Integer nummer = Integer.parseInt(line[1]) - 1;
                                if (!line[0].equals("Sdt") && nummer < 0) {
                                        fehler(inFromClient, outToClient, socket, line);
                                        return;
                                }
                                if (line[0].equals("Witz")) {
                                        String w = witz[nummer % witz.length];
                                        int linesofwitz = countLines(w);
                                        outToClient.printf("Answer %s /\r\n", linesofwitz);
```

```
                                outToClient.printf("%s\r\n", w.replace("_", clientName));
                            } else if (line[0].equals("Spruch")) {
                                String s = spruch[nummer % spruch.length];
                                int linesofspruch = countLines(s);
                                outToClient.printf("Answer %s /\r\n", linesofspruch);
                                outToClient.printf("%s\r\n", s.replace("_", clientName));
                            } else {
                                String s = spruch[0];
                                int linesofspruch = countLines(s);
                                outToClient.printf("Answer %s /\r\n", linesofspruch);
                                outToClient.printf("%s\r\n", s.replace("_", clientName));
                            }
                        }
                    }
                }
            }
```

Viel Erfolg !!!