

VJEŽBA 5: Soketi (2. dio, programski jezik Java)

Zadatak ove vježbe je istovjetan zadatku iz prethodne vježbe, samo ga je potrebno realizirati koristeći programski jezik Java (a ne C).

Zadatak. U nastavku ovog dokumenta dani su .java dokumenti u kojima se nalazi programski kod za servera i klijenta. Vaš zadatak je modificirati dani kod na slijedeći način:

1. klijent serveru šalje neki broj (tipa float);
2. server kvadrira primljeni broj;
3. server kvadrirani broj šalje klijentu;
4. klijent ispisuje primljeni rezultat.

Objašnjenje programskog koda: TcpClient.java

```
// naredba "import" u Javi je analogna naredbi "#include" u C-u
import java.net.*;    // ukljuci sve klase definirane u paketu java.net (za rad sa soketima)
import java.io.*;     // ukljuci sve klase definirane u paketu java.io (za osnovne IO, tj. input
                      // output, operacije)

public class TcpClient // ime glavne klase je TcpClient
{
    // public: main() je prva metoda koja se poziva iz Java okruzenja i zato mora biti public
    //          (funkcije u Javi se nazivaju metode)
    // static: Java okruzenje mora biti u stanju pozvati main() metodu bez da stvara instancu
    //          klase, pa zato mora biti static
    // void:    main() ne vraca nista
    // throws IOException: kazemo kompajleru da metoda main() moze baciti IO iznimke
    public static void main(String args[]) throws IOException
    {
        // otvori soket klijenta prema serveru adria.fesb.hr koji slusa na portu 50076
        Socket s1 = new Socket("adria.fesb.hr", 50076);

        // ispisi poruku ako je soket uspjesno kreiran
        System.out.println("Klijent se spojio na port 50076!");

        // dohvati soketov tzv. input file handle i output file handle
        InputStream s1In = s1.getInputStream();    // dohvati podatke koji se salju soketu
        OutputStream s1out = s1.getOutputStream(); // salji podatke ovom soketu

        // dis = data input stream (da bi se podaci mogli primiti na ovaj soket)
        BufferedReader dis = new BufferedReader(new InputStreamReader(s1In));
        // dos = data output stream (da bi se podaci mogli slati sa ovog soketa)
        DataOutputStream dos = new DataOutputStream(s1out);

        try
        {
            if (args.length > 0) // ako je korisnik unio broj za kvadriranje
            {
                // ovdje umetnite kod za slanje broja serveru (koristite
                // dos.writeChars() za slanje podataka serveru, i dis.readLine() za
                // primanje podataka sa servera)
            }
        }
        catch (Exception e)
        {
            // uhvati neku nepoznatu iznimku (i nemoj srusiti program)
        }
    }
}
```

```

        // procitaj i ispisi String kojega nam posalje server
        String st = new String(dis.readLine());
        System.out.println(st);

        // na kraju zatvori konekciju i izadji
        dis.close();
        dos.close();
        s1In.close();
        s1out.close();
        s1.close();
    }
}

```

TcpClient.java (analogno i za TcpServer) kompajlirate i izvršavate na slijedeći način:

```

javac TcpClient.java // kompajliraj
java TcpClient      // izvrši

```

Objašnjenje programskog koda: TcpServer.java

```

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

public class TcpServer
{
    public static void main(String args[]) throws IOException
    {
        ServerSocket s = new ServerSocket();

        // povezi soket servera sa hostom i portom na kojemu ce slusati
        s.bind(new InetSocketAddress ("adria.fesb.hr", 50076));
        System.out.println("Soket kreiran na portu 50076!");

        while(true) // beskonacna petlja (server ceka konekcije)
        {
            System.out.println("Cekam na novu konekciju...");

            Socket s1 = s.accept();

            OutputStream s1out = s1.getOutputStream();
            DataOutputStream dos = new DataOutputStream(s1out);
            InputStream s1In = s1.getInputStream();
            BufferedReader dis = new BufferedReader(new InputStreamReader(s1In));

            // ovdje umetnite kod za primanje broja za kvadriranje i slanja rezultata
            // klijentu (koristite dis.readLine() za primanje podataka, i dos.writeChars()
            // za slanje podataka klijentu)

            dos.close();
            dis.close();
            s1out.close();
            s1.close();
        }
    }
}

```

Napomena za TcpServer.java: podaci koje TcpServer primi biti će tokenizirani, tj. rastavljeni na tokene. Na primjer, ako je klijent poslao serveru string "23", server će to primiti kao "\02\03\0". Da bi se string "\02\03\0" mogao konvertirati u float, najprije je potrebno izbrisati sve znakove "\0" iz njega. To se

radi na slijedeći način:

```
String novistring = stariString.replaceAll("\0", ""); // izbriši znakove "\0"  
float primljeniBroj = Float.parseFloat(noviString); // konvertiraj string u float
```