DATANETTVERK: LABOPPGAVE 4

Gjennomført med:

-Kirill Gaydamak

Vi valgte å bruke programmene skrevet i Java. Vi kopierte disse og lagret filene på PC-ene.

1)

Klient program:

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

public class Klientmaskin {

    Run | Debug
    public static void main(String[] argv) throws IOException {

        String sendetekst=" Hello fra klient";
        if(argv.length != 0)
            sendetekst = argv[0];

        Socket klientSocket = new Socket("localhost", 7004);
        PrintWriter ut = new PrintWriter(
            klientSocket.getOutputStream(), true);
        BufferedReader inn = new BufferedReader(
            new InputStreamReader(klientSocket.getInputStream()));

        ut.print(sendetekst + "\r\n");
        ut.flush();

        String svartekst = inn.readLine();
        System.out.println("Svar fra Tjenermaskin: " + svartekst);

        klientSocket.close();


        }
}
```

Server/tjenerprogram:

```java
import java.io.*;
import java.net.*;
public class Tjenermaskin {

Run | Debug
public static void main(String[] args) throws IOException {

    System.out.println("Tjenermaskin er startet.... ");
    ServerSocket tjenerSocket = new ServerSocket(7004);

    while(true){
        Socket koplingSocket = tjenerSocket.accept();

        BufferedReader innput = new BufferedReader(
            new InputStreamReader(koplingSocket.getInputStream()));
        PrintWriter utTilKlient = new PrintWriter(
            koplingSocket.getOutputStream(), true);

        String melding = innput.readLine();
        utTilKlient.print("Flott, du sendte " + melding + "\r\n");
        utTilKlient.flush();
        }

    }
}
```

I denne oppgaven brukes localhost, slik at jeg kjører begge på min maskin. Compiler og kjører disse i terminalen. Fikk forventet resultat:

Klient:

```
alexander@S540:~/Documents$ javac Klientmaskin.java
alexander@S540:~/Documents$ java Klientmaskin
Svar fra Tjenermaskin: Flott, du sendte  Hello fra klient
alexander@S540:~/Documents$
```

Tjener:

```
alexander@S540:~/Documents$ javac Tjenermaskin.java
alexander@S540:~/Documents$ java Tjenermaskin
Tjenermaskin er startet....
Tjener svarer:  Hello fra klient
```

Vi la her til en linje som printer hva som blir sendt tilbake til klienten. Dette slik at det skulle bli lettere å se hva som foregikk.

2)

Kobler meg til Kirill sin PC gjennom å bytte ut localhost med hans IP-adresse. Slik:

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

public class Klientmaskin {

Run | Debug
public static void main(String[] argv) throws IOException {

    String sendetekst=" Hello fra klient";
    if(argv.length != 0)
        sendetekst = argv[0];

    Socket klientSocket = new Socket("158.39.190.40", 7004);
    PrintWriter ut = new PrintWriter(
        klientSocket.getOutputStream(), true);
    BufferedReader inn = new BufferedReader(
        new InputStreamReader(klientSocket.getInputStream()));

    ut.print(sendetekst + "\r\n");
    ut.flush();

    String svartekst = inn.readLine();
    System.out.println("Svar fra Tjenermaskin: " + svartekst);

    klientSocket.close();


    }
```

Følger eller samme fremgangsmåte som i første oppgave. Jeg kjørte tjenerprogrammet, også byttet vi på for å sjekke at det fungerte fra begge sider. Når jeg kjørte tjenerprogrammet fikk jeg dette opp i terminalen

```
alexander@S540:~/Documents$ javac Tjenermaskin.java
alexander@S540:~/Documents$ java Tjenermaskin
Tjenermaskin er startet....
Tjener svarer:  Hello from your darling client
```

Her er teksten spesifisert på Kirill sin maskin. Jeg tar imot, og sender tilbake. (Merk linjen som vi la til i første oppgave er ogsp med her). Slik at man kan se hva tjeneren svarer med.

3)

Forsøkte først å modifisere eksisterende kode slik at både server og klient kan skrive til hverandre. Vi tok utgangspunkt at tekst inn for klient vil være tekst ut for server, og omvendt. Tekst inn for tjeneren, er tekst ut for klienten.

Begge programmene tar først input og skriver denne til konsollen, deretter skrives output inn i konsollen og sendes til server/klient.

Forsøket på å modifisere gjeldende kode gikk ikke helt som forventet. Vi fikk tatt input fra terminalen og sendt dette over, men teksten kom ikke frem til eksempelvis server før tilkoblingen ble avsluttet. Etter litt googling fant vi ut at vi ut at det også var mulig å benytte seg av DataInputStream/ DataOutputStream klassene istedenfor eksempelvis PrintWriter. Vi må også da bruke andre metoder for å skrive/lese input.

Vi la også inn en stop-kondisjon. Altså while-løkken med funksjonaliteten kjører så lenge input i terminalen er “close”.

Klient:

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

public class Client {
Run | Debug
public static void main(String[] args) throws IOException {

    //Input fra terminalen
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    Socket klientSocket = new Socket("localhost", 7004);
    DataInputStream inn=new DataInputStream(klientSocket.getInputStream());
    DataOutputStream out=new DataOutputStream(klientSocket.getOutputStream());


    String input ="";
    String output= "";

    while(!output.equals("close")){

    output= br.readLine();

    out.writeUTF(output);
    out.flush();

    input= inn.readUTF();
    System.out.println("Tjenermaskin sier: " + input);

    }

    klientSocket.close();
    out.close();

    }
}
```

Tjener:

```java
import java.io.*;
import java.net.*;
public class ServerMan {

    Run | Debug
    public static void main(String[] args) throws IOException {

        System.out.println("Tjenermaskin er startet.... ");
        ServerSocket tjenerSocket = new ServerSocket(7004);
        Socket koplingSocket = tjenerSocket.accept();

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        DataInputStream inn=new DataInputStream(koplingSocket.getInputStream());
        DataOutputStream out=new DataOutputStream(koplingSocket.getOutputStream());

        String input ="";
        String output = "";


        while(!input.equals("close")){

        input= inn.readUTF();
        System.out.println("Klientmaskin sier: " + input);

        output= br.readLine();

        out.writeUTF(output);
        out.flush();
            }

        inn.close();
        koplingSocket.close();
        tjenerSocket.close();

        }
```

Jeg brukte localhost og kjørte disse programmene på samme maskin. Programmet fungerte ikke som forventet ved første anledning. I tillegg til overnevnte måtte vi definere strings før while-løkken og heller skrive de over senere. Definerte vi de for første gang inne i selve skrivingen/lesingen kom teksten ikke frem.

```
alexander@S540:~$ cd Documents
alexander@S540:~/Documents$ javac ServerMan.java
alexander@S540:~/Documents$ java ServerMan
Tjenermaskin er startet....
Klientmaskin sier: Hay
Hello
Klientmaskin sier: What is up
Nothing
Klientmaskin sier: Same
```

```
alexander@S540:~/Documents$ javac Client.java
alexander@S540:~/Documents$ java Client
Hay
Tjenermaskin sier: Hello
What is up
Tjenermaskin sier: Nothing
Same
```