

# 1 Studieopgave

Der er løst opgave nr. 2 med Chat programmet.

## 1.1 Skitser problemet

Klient/Server: to processer der taler sammen over et netværk. Dette gøres ofte gennem sockets med TCP forbindelse.

## 1.2 Skitser løsningen

En server startes og den stiller sig i position til at åbne en socket. Herefter kan en klient koble sig op til denne socket. Når dette sker, looper serveren sådan at den igen åbner en ny socket. Dette er muligt da der laves en ny tråd for hver socket der bliver åbnet på server siden. Når der kommer input stream på den socket som en server overvåger, så sender den som en output stream til alle sockets(writers). Disse sockets er delt mellem alle server tråde i en arrayliste.

## 1.3 Tekniske detaljer

Vi kigger på de dele der er netværksspecifikke.

## 1.4 Læringsprocessen

Erfaring med at programmere netværks applicationer.

# 2 Kode

```
1  ...snip...
2  Date date = new Date();
3  String s = "" + nickname + " connected on " + date + "
   \n";
4  writer.write(s);
5  writer.newLine();
6  writer.flush();
7  while(true)//keeps the process alive, waiting for
   input...
8  {
9      outputField.append(reader.readLine() + "\n");
10 }
11 ...snip...
```

```

1  ...snip...
2  reader = new BufferedReader(new InputStreamReader(
3      socket.getInputStream()));
4  writer = new BufferedWriter(new OutputStreamWriter(
5      socket.getOutputStream()));
6  place = writers.size();
7  writers.add(writer);//add to shared list
8
9  ...snip...
10 while(writers.size() > 0){
11     String input = reader.readLine();
12     System.out.println(input);
13     for (int i = 0; i < writers.size(); i++){
14         writer = (BufferedWriter)writers.get(i);
15         writer.write(input + "\n");
16         writer.flush();
17     }
18 }
19 ...snip...

```

```

1  ...snip...
2  public static void main(String[] args){
3      ServerSocket serverSocket = null;
4      Socket socket = null;
5      ArrayList newwriters = new ArrayList();
6      try{
7          serverSocket = new ServerSocket(PORT);
8      } catch (IOException e){
9          System.out.println("Could not create server socket
10             ");
11      }
12      while (true){
13          try{
14              socket = serverSocket.accept();
15          } catch (IOException dk){
16              System.out.println("Could not accept socket");
17          }
18          new Server(socket, newwriters).start();
19      }
20  }
21  ...snip...

```