

## Chat Server Obligatorisk Aflevering

Jeg har udviklet en Chat Program hvor jeg har brugt Threads/Threadspools til at håndtere flere task af gangen.

Min Protokol er følgende:

1. Start server med Threadpools(åbner 5 porte)
2. Start en eller flere Clienter(max 5 af gangen) og vælg en ledig port.
3. Skriv username(Brugt RegEx patteren for at holde Username kravne)
4. Servern godkender username og connecter Client til Server
5. Clienterne har mulighed for at skrive i Consollen som derefter vil bliver vist i serveren.
6. Alt hvad Serveren modtager bliver skrevet ind i en ChatLog.txt, hvilket indeholder dato/tidspunkt og metode/classenavn, samt Username og Besked.
7. Når Client er færdig med at Chatte så kan de skrive "stop" hvorefter den pågældende port vil blive ledig.
8. Serveren får besked på at porten er lukket, og så en ny Client kan Connecte på samme port.
9. Hvis Serveren lukker vil alle Clienter få en fejlmeddeles og lukke ned.

I mit program har jeg valgt at benytte mig af Runnable hvilket er Javas indbygget Interface for håndtering af Threads. Derudover har jeg benyttet mig af Socket/InputStream/outputstream samt ServerSocket til at håndtere connection samt kommunikation mellem Server og Client.

I min Client kasse har jeg lavet 3 metode som holder styr på hver sit ting.

Den første metode er Port() hvilket returnere et portnr som Klienten tilslutter til serveren.

Derudover har jeg også en UserName() metode som håndtere diverse krav til brugernavnet. Til dette har jeg valgt at benytte mig af et regex pattern:

```
^[a-zA-Z.\_-]{1,12}$
```

Dette gøre at Username max må have 12 Char samt kun "-" og "\_" som special symboler.

Til sidst har jeg en Chat() metode hvilket snakker sammen med server klassen via Socket, DataOutputStream og DataInputStream. I denne metode anvender jeg også et Regex pattern:

```
^[0-9a-zA-Z.\_-!~æøåÆØÅ]{1,250}$
```

Dette pattern holder styr på at der max må være 250 tegn, samt ÆØÅ.

Til fejlhåndtering har jeg benyttet mig af Try/Catch til at håndtere fejl i min kode:

```
IOException – Bruges til at håndtere BufferedReader.  
ConnectException – Bruges til at håndtere Sockets  
InputMismatchException – Bruges til at håndtere forket Scanner input.
```

#### Konklusion:

Jeg har formået at lave et virkende program hvor man via Threads kan tilslutte flere Clienter til en Server samt have en log til at holde styr på alle chat beskeder. Det eneste jeg ikke fik lavet var en Connection mellem alle Clienter så hver tilsluttet Client kan se hinandens beskeder, hvilket på nuværende tidspunkt kun kan ses gennem Server.

From client to server. The user name is given by the user. Username is max 12 chars long, only letters, digits, '-' and '\_' allowed. – UDFØRT

From server to client. Client is accepted. – UDFØRT

From server to client. Client not accepted. Duplicate username, unknown command, bad command or any other errors. – UDFØRT

From client to server. From server to all clients. First part of message indicates from which user it is, the colon(:) indicates where the user message begins. Max 250 user characters. - UDFØRT

From client to server. Client sends this heartbeat alive every 1 minute. – IKKE UDFØRT

From client to server. Client is closing down and leaving the group. – UDFØRT

LIST <> From server to client. A list of all active user names is sent to all clients, each time the list at the server changes. – IKKE UDFØRT