**Klient/server-chatprogram**

**Hvad**:

Jeg har lavet et klient/server-chatprogram i Java, hvor flere klienter har mulighed for at tilkoble sig den samme server og chatte med hinanden igennem serveren.

**Hvordan**:

Programmet er lavet ved hjælp af en Socket og en Port, hvor serveren og klienten kommunikerer igennem. Programmet er multitrådet. Hver gang en ny klient tilkobler sig serveren, opretter serveren en ny tråd til at håndtere denne (ClientHandler). Der kan kun være 5 klienter logger ind ad gangen, hvilket kontrolleres af en FixedThreadPool. Klienterne samles i en liste, der er lavet ved hjælp af Javas indbyggede klasse Vector. Det er smart at bruge Vector fremfor en almindelig ArrayList, når man håndterer tråde, da en Vector-liste automatisk er synkroniseret.

**Design**:

Jeg har selvfølgelig tænkt objekt-orienteret i mit design. Klientprogrammet har en bruger, som jeg har valgt at lægge i en selvstændig klasse. Ud over main-tråden kører der tre tråde i programmet. AccessHandler tager sig af kontakten til serveren (logge ind og chatte), hvor brugeren aktivt skal skrive noget, og ReadThread tager sig af at læse input fra serveren. I et program med flere klienter skal vi kunne modtage input uafhængigt af output og derfor skal vi bruge både en tråd til at skrive og en til at læse. HeartbeatService er en tråd, der opdaterer brugerens heartbeat og sender en standardiseret besked til serveren. Der er en tilsvarende service i serverprogrammet, der tager sig af at smide klienter af serveren, hvis de ikke har sendt et heartbeat inden for 2 minutter.

**Designmønstre**:

Jeg har ikke fået implementeret nogen designmønstre. Man kunne vælge at lave et singleton pattern flere steder, da flere af trådklasserne kun skal instansernes én gang.

**Mangler**:

Jeg har endnu ikke fået implementeret en logfunktion, hvor serveren logger enhver interaktion. Jeg forestillede mig at bruge Javas egen Logger-klasse og opbevare loggene i en txt-fil.

Håndteringen af en sjette klient, der forsøger at logge sig ind, er endnu ikke elegant lavet. Der bør laves et tjek, så klienten får at vide, at serveren er fuld.

Når en klient har skrevet QUIT og på den måde har forladt serveren, nedlægges forbindelsen mellem klientprogrammet og serverprogrammet, men en bruger kan endnu ikke afslutte klientprogrammet eller logge ind igen.

Javadoc forefindes.