

Meget kort rapport innl. 2 Avansert Java

Av Thomas René Gabrielsen

Eventuell forutsetninger

1. Ha programvare klart for å kjøre javaprojekt
2. Ha en tilgjengelig MySQL server tilgjengelig.
3. Kjør samtlige skript som er vedlagt
4. Ha port 80 tilgjengelig (Skype har til tendens å tro at den har enerett på port 80...)
5. Start server før klient naturligvis.

Kommentarer til løsningen

Først og fremst, dette ble en meget kort rapport. Da fokus har vært på kode og ikke rapportskriving denne gang. Så får rapporten være fokus til mappeinnleveringen!

Kort om klienten:

Den er basert på Stein/Per sin kode om client multithreading. Er gjort litt endringer blant annet er samtlige `.close()` plassert i finally klosulen ved hjelp av en hjelpemetode.

Rekkefølgen på les/skriv til/fra server er spesialtilpasset hvordan serveren kommuniserer, dette er forklart som kommentar i koden.

ServerController

Planen var å følge Model View Controller standarden for serveren, men outputmanageren er ikke begynt på. Ellers har jeg valgt en ServerController som tar imot input av serverbrukeren (admin). Denne input prosessen kjører på hovedtråden, og kan ta imot input som quit, load (laster inn sanger på nytt) og andre fremtidige funksjoner. ServerControlleren starter også en ny tråd som serveren kjører på.

Server

Serveren starter med å laste inn musikk fra databasen for deretter lukke koblingen. Etter dette vil den gå i en mainloop og lytte på nye klienter. Når en ny klient kobler til blir klienten og lista over spørsmål sendt til en ny tråd (en tråd for hver klient). Ideen med å også sende med spørsmålene var at da kan admin oppdaterte denne spørsmålslista med klienter fortsatt kjører, og nye klienter vil bli spurt på den oppdaterte lista. Dette fungerer så vidt jeg kan se.

ClientCommunicator

Er en Runnable og vil kjøres som egen tråd. Det er denne som inneholder selve Quiz-logikken. Jeg tror koden her er ganske lett lesbar. Framgangsmåten er:

1. Tråden velger et random spørsmål fra lista
2. Sender spørsmål til klient
3. Tar imot input fra klient
4. Sammenligner svaret fra klient med svaret til spørsmålet
5. Sender feedback (correct/wrong) til klient
6. Sender nytt spørsmål til klient
7. etc..

Hvis bruker skriver inn "quit" vil loopen avsluttes og poengsummen vil bli lagret.

Question_

Under dette namespace ligger alt som har med selve spørsmålet å gjøre.

En spørsmålsgenerator, som tar inn en liste (som blir generert i DBHandler) og omgjør denne til spørsmål. Har per nå kun to ulike typer spørsmål. Ting som String.format (i lignende format) er noe jeg er vant med fra C#, så var bare et kjapt googlesøk for å finne ut hvordan denne fungerer i Java.

Questions inneholder en liste over Question. Hvert Question har et Answer object i fields. I Answer objektet er det en metode som sjekker om svaret er korrekt.

Planen her var at Questions skulle implementere ArrayList<>, men av en eller annen grunn så funka dette bare ikke... Så må se mer på det.

DBHandler

Inneholder klasser relevant til databaseoperasjoner. Music og Score er to klasser som brukes til DAO interfacet, DBServices inneholder hjelpeklasser som skriver/leser fra databasen mens DBConnection oppretter/returnerer selve koblingen. Jeg tror disse er lett lesbare og lite å kommentere her. Prøvde å fikse clear() men dette fikk jeg ikke til – må se på dette også.

Testing

Meh... Fikk ikke tid. Selv om jeg egentlig hadde ganske god tid.

MySQL scriptene

Inneholder opprettelse av Score og Music tabellene. Samt 10 eksempelsanger.

Eventuell spørsmål

1. Først og fremst så lurer jeg på om det som er skrevet nå har bra eller dårlig oppsett. Når jeg får luket input ut i egen klasse så mener jeg løsningen er nokså komplett. Arbeid deretter blir kun pynting (som clear()).
2. Jeg prøvde å lage clearScores() men problemet her var jo at DBServices objektet allerede var fjerna etter jeg brukte det sist, så da ble det på en måte clear av ingenting. Lurer på om du har noen tips til løsning her?
3. Har du noen tips til hvordan man kan gjøre Questions til en liste i seg selv? Istedefor å inneholde en liste, ser litt bortkastet ut.
4. Vis hvordan Maven kan hjelpe til med utvikling. Har du noen tips til dette med Maven bygging? Hang ikke helt med når Per gjennomgikk dette.
5. Hvis server sender spørsmål til klient så er det ingen måte å fortelle klient at server lukker seg før klient har skrevet svar. Noen tips til dette? En løsning vil jo være en egen tråd som fokuserer på input/output mens en annen klienttråd fokuserer på koblingen til serveren?
6. Når jeg avslutter serveren så er serverkoden allerede inni while(serverIsRunning) loopen. Dette medfører at try { clientSocket = server.accept(); } vil feile en gang når serveren lukkes noe som er estetisk irriterende. Men ser egentlig ingen utveier på dette annet enn å fjerne feilmeldingen fra å printes.