

Innlevering 2 PGR200

Oppgaven musikk-QUIZ

Hovedformålet med innleveringen var å lage en server som tilbyr en QUIZ tjeneste som klienter kan kontakte, hvor de svarer på spørsmål de mottar og serveren gir tilbakemelding. Dette fortsetter inntil klienten ikke lenger ønsker å få nye spørsmål. Serveren er alltid oppe og mottar opplysninger om quizen fra en database.

Introduksjon

Hensikten var å lage en maven prosjekt som tar i bruk sockets, og JDBC med tanken på god kodekvalitet for å vise hva den kan bidra med i utvikling av prosjektet.

Dere som vurderer prosjektet kan benytte alle maven kommandoer unntatt for deploy som gir feilmeldingen pga at jeg ikke har oppgitt en remote repository.

Altså prosjektet kan valideres, kompiles og ved å bruke kommando package kan dere bygge prosjektet. For å kjøre applikasjonen, først kjør Server klassen, og så Klienten (eller flere).

Prosjektet består av:

Client package med Client klasse.

Database package med DBConnector, DBHandler og DBService.

Quiz package med MusicQuiz og Quiz.

Server package med Server/QuizThread.

Test klasser:

Har tatt i bruk Junit 4.12 for å lage tester. Består av 16 tester alt sammen:

DBConnectorTest:

Sjekker om forbindelse er opprettet, om forbindelsen lukkes osv.

DBHandlerTest:

Sjekker om metoden readProperties() leser konfigurasjon fra properties fil som tiltenkt.

DBServiceTest:

Sjekker om metodene createMusicQuiz() og createMusicQuizList() funker som tiltenkt

MusicQuizTest:

Trenger ikke forklaring.

QuizTest:

Tester flere metoder fra Quiz klassen, blant annet askQuestion, om answersCorecctly() leser riktig svarene, tester om klienter får tilbakemeldingen, og mange flere test metoder som tester for funksjonaliteten.

ServerTest:

Tester om forbindelsen er opprettet.

Kravene som er dekket:

Server : henter sanger og venter på kontakt, starter klienttråd, oppretter streams, lager spørsmål, Sender spørsmål, tar imot svar og gir tilbakemelding, og er alltid oppe.

Client : oppretter streams, tar imot spørsmål og sender svar og avslutter.

Annet : prøvde å gjøre programmet mest mulig robust og SQL injection safe, bruker maven og automatiserte tester.

Tester : Alle tester er grønne :)

Kort beskrivelse av klasser:

Server/QuizThread :

Åpner en server socket på gitt port nummer. Oppretter en client socket for hver forbindelse og passerer den til den nye client threaden. Stotter opptil 10 client threads.

Åpner en input og output stream for gitt klient, tar imot spørsmål og svarer så lenge man vil fortsette å spille. Synchronised statement tillater oss å synkronisere gjennomføringen av de kritiske deler av

programmet. Får poeng for riktig besvarte spørsmål.

Når vi får “Bye” melding fra serveren lukkes den og settes til null slik at nye klienter kan koble seg til serveren. Til slutt luker input,output og socket.

Client :

Åpner en socket på en gitt host og port. Åpner input og output streams. Skriver data til socketen som var åpnet. Lager en thread for å lese fra serveren, leser fra socket til vi får en “Bye” melding fra serveren.

Lukker output og input stream og socket.

DBConnector :

Oppretter forbindelse mot en database.

DBHandler :

Leser konfigurasjonen med hostname, password, database osv. for å opprette en forbindelse mot en database fra en property fil dbinfo.properties

DBService :

Lager en ArrayList med verdiene fra gitt database som skal da brukes for å lage quiz spørsmål.

MusicQuiz :

Klasse med getters og setters.

Quiz :

Først lager en forbindelse mot en database, bruker DBService sin metoder for å lage tabellen og for å lage en liste med verdiene fra gitt tabell. Lager spørsmål, sjekker om de var riktig besvart, og gir tilbakemelding. Henter nye spørsmål tilfeldig og har oversikt over score.

Forutsetninger:

Mulige utvidelser :

Har ikke utvidet tjenesten til å kunne stille flere typer spørsmål.

Har utvidet tjenesten slik at klienten oppnår en score, men den ikke lagres i databasen.