

Innlevering 1 PGR200

Introduksjon

Hovedformålet med innleveringen var å lage en Maven prosjekt som tar i bruk JDBC og testing med tanken på god kodekvalitet.

Prosjektet består av 5 klasser:

1. DBClient
 - a. Klassen med main metoden som kjører programmet.
2. DBConnector
 - a. Klassen som oppretter konneksjon med MySQL Database.
3. DBHandler
 - a. En klasse som inneholder noe hjelpemetoder til våre hovedmetoder og til klienten.
4. DBService
 - a. Klassen som inneholder hovedmetoder copyFile() and showTable() som tilbyr funksjonaliteten som var kravet for dette prosjektet.
5. FileIO
 - a. En klasse som har oppgaven til å lese filer og inneholder hjelpemetoder readFile() og getTableHeader().

I tillegg er det 5 test klasser:

1. DBConnectorTest
2. DBHandlerTest
3. DBServiceTest
4. FileIOTest
5. ShowTableTest

Jeg føler at disse 5 test klassene var viktige å teste fordi de inneholder metoder som har behov for automatiserte tester med god testdekning.

Kravene som var dekket:

Prosjektet var laget i Maven med java 8. Har hovedmetoder copyFile() og showTable() i tillegg til hjelpemetoder. Alle metodene og hovedfunksjonene til programmet var testet. Prøvde å gjøre programmet mest mulig SQL injection safe og robust.

Etter at jeg har fått tilbakemeldingen på første innleveringen, har jeg lagt til/endret følgende:

1. Har ryddet opp i metodene, har lagt til følgende nye metoder og klasser:
 - a. DBHandler

- i. doDatabaseWork()
 - ii. readProperties()
 - iii. prepareForImport()
- b. DBService
 - i. dropTables()
- c. FileIO

2. Jeg har skyllet ut hvordan man snakker med databasen på i sitt eget lag, og har tatt i bruk properties fil.

3. Har endret koden slik at eksempel filen kan utvides med flere attributter.

4. Endret format på showTable() slik at den vises som du anbefalte i tilbakemeldingen.

5. Har slettet redundant kode i klienten.

6. Filnavn kan nå settes som en property.

7. Har forkortet metodene og lagt til flere nye.

8. Endret src/target i pom for java versjon.

9. Utbedret robusthet, input til public metoder sjekkes for null.

10. Flere tester, tester ryddes opp ved hjelp av setup og teardown.

Kort beskrivelse av klasser og metoder

Har utelatt noe små metoder som for eksempel releaseConnection() fordi de er mer eller mindre selvforklarende.

DBService

«Hoved» Klasse som inneholder hovedmetoder som tilbyr funksjonalitet som var kravet for dette prosjektet.

copyFile(String fileName, String tableName)

Setter inn data fra leste filen og oppretter en database tabell med disse verdiene.

Vector sqlColumnOrder holder oversikt over kolonner, og statement kjøres når data er lest. Jeg brukte flere hjelpemetoder her: readFile() prepareforImport(tableName, data, sqlColumnOrder) og createTable().

createTable(String fileName, String tableName)

Oppretter en table i database. En spørring med CREATE TABLE er forberedt og statement er klar til å utføres.

showTable(String tableName)

Skriver ut alle rader og kolonnenavn fra en database tabell til console.

En spørring med SELECT * FROM tableName er forberedt og utføres, deretter henter data fra tabellen med getMetaData() og skriver ut alle verdier til console.

DBClient

DBClient kjører applikasjonen. Alt som skjer når programmet kjøres vises i console vinduet. Når applikasjon kjøres, får brukeren 2 alternativer til å koble seg mot en database.

Case 1: Ved å skrive 1 i console vil brukeren kunne velge brukernavnet og passordet selv for å koble seg mot en database.

Case 2: Andre alternativ er skrive 2 for å bruke default login.

Case 3: Ved å skrive 3 vil programmet avsluttes.

Etter manual/ default inlogging med hjelpemetode dbLogin() vil programmet kjøre 2 hovedmetoder som var kravet for dette prosjektet: copyFile og showTable, ved hjelp av hjelpemetode doDatabaseWork().

DBConnector

DBConnector oppretter forbindelse mot en database.

FileIO

Klassen som leser filer. Har metoder:

readFile(string fileName)

Leser ut verdiene og setter dem til tilsvarende tokens, som settes da i en HashMap record. Alle records er lagt til en Vector data som inneholder alle verdiene fra filen og som klassen skal returnere etter at metoden er kjørt.

getTableHeader(fileName)

Hjelpemetode. Leser en fil og oppretter arrays for kolonne navn, datatype og datasize.

DBHandler

Har ulike oppgaver, hensikten med klassens metoder er å ha litt bedre oversikt av koden i prosjektet.

doDatabaseWork(dbService, filePath, tableName)

Kjører copyFile() og showTable() i klienten.

readProperties()

Leser en properties fil som er lagt til i mappen. Brukes i klienten.

Test Klasser

DBConnectorTest

getConnectionTest() sjekker om forbindelse opprettes riktig.

isConnectionClosing() sjekker om forbindelse lukkes som tiltenkt.

getManualConnectionTest() sjekker om forbindelse med parametre funker som tiltenkt.

DBServiceTest

copyFileFailTest() sjekker om `copyFile()` method funker som tiltenkt, her sjekker jeg om metoden registrerer når filen som ikke eksisterer er oppgitt.

CopyFileTest() sjekker om forventet verdier er lagt til en database tabell ved å sammenligne dem.

FileIOTest

getTableHeaderColumnTest() sjekker om `getTableHeader()` leser riktig linjer fra en fil for å kunne passere dem som kolonnenavn når vi oppretter en ny tabell.

getTableHeaderTest() sjekker om metoden funker som tiltenkt, her sjekker jeg om metoden registrerer når filen som eksisterer/ikke eksisterer er oppgitt.

readFileTest() sjekker om metoden leser filen ved å asserte at verdiene er ikke null.

ShowTableTest

showTableTest() sjekker om `showTable()` viser utskriften i console riktig ved å sammenligne første forventet verdien.

DBHandlerTest

readPropertiesTest() sjekker om `readProperties()` funker ved å asserte at verdiene som er lest er ikke null.

Forutsetninger

Kunne noe vært gjort bedre?

Første versjonen av innleveringen inneholdte flere forutsetninger som var nevnt i dokumentasjonen, og alt dette har jeg nå endret og utbedret hele koden. Jeg har også tilpasset koden etter tilbakemeldingen som jeg har fått fra veilederen.