
Database prosjekt

Innledning

Programmet *filmDB* er et java-program ment for å gi en lettere måte å interagere med film databasen. Programmet lar brukere se detaljer om filmer, personer og selskaper, samt legge til nye elementer i databasen. I tillegg er det funksjonalitet for å legge til brukerkommentarer og anmeldelser av filmer og serier.

Klasser i programmet

De ulike klassene i programmet er følgende:

- Main
- UI
- DBConn
- DBController
- BaseController
- PersonController
- UserController
- FilmController
- CategoryController
- FilmCompanyController

Main

Main klassen har statiske attributter som inneholder relasjoner til henholdsvis en instans av alle kontroller klassene. I tillegg har den en relasjon til en UI klasse, som håndterer alt av brukergrensesnitt. Main klassen har en statisk main metode som starter hele programmet, ved å vise frem en hovedmeny og håndtere logikk for å velge hvilke kontrollere som skal brukes basert på brukeren sitt valg.

UI

UI klassen definerer et sett med metoder som kan brukes av kontroller klassene for å kontrollere input/output. UI metodene baserer seg alle sammen på host systemets input/output. Det gjør at alt av brukergrensesnitt er tekst-basert. Viktige metoder i denne klasen er **menu**, som viser en menu og returnerer brukerens valg, og **printItems** som tar inn et resultat fra en spørring og viser frem dette. I tillegg er det definert flere andre metoder for bl.a. bruker input og error meldinger.

DBConn

`DBConn` er en abstrakt klasse som definerer metoder for å koble til, samt lage databasen. Denne klassen brukes av alle kontrollere.

DBController

`DBController` er en klasse som definerer et par metoder for å administrere databasen. Dette er bl.a. å legge inn *fixtures*(eksempel data) og å opprette tabellene i databasen.

BaseController

`BaseController` er en klasse som definerer metoder som vil være felles for alle kontrollerene. Dette er kun `listAllItems()` som henter alle elementer fra den respektive tabellen, og viser disse frem vha. `UI` klassen.

PersonController

`PersonController` er en klasse som definerer metoder for å behandle personer i databasen. Den har en metode for å vise en submeny for operasjoner den definerer. Dette innebærer å vise frem alle personer, legge til nye personer, se detaljer om en bestemt person, og å se alle roller for en skuespiller.

UserController

`UserController` er en klasse som kontrollerer alt av bruker logikk i databasen. Den krever at man har valgt en bruker, som enten kan gjøres ved å ta inn brukernavn, eller å lage en ny bruker. Videre gir denne en meny som lar en liste alle kommentarer og anmeldelser, samt å legge inn en ny kommentarer eller anmeldelse.

FilmController

`FilmController` kontrollerer filmer/serier i databasen. Den har metoder for å liste alle filmer, samt å legge til filmer.

CategoryController

`CategoryController` kontrollerer kategorier. Den har metoder for å vise frem alle kategorier, og å legge til nye.

FilmCompanyController

`FilmCompanyController` har metoder som omhandler filmselskaper. Den gir metoder for å vise og legge til selskaper, i tillegg til å liste opp alle filmselskaper som har produsert filmer innen en kategori.

MusicController

`MusicController` har metoder for å liste musikk, samt legge inn ny filmMusikk.

Usecases

These directions will assume that you have a working program up and running and that you are in the first menu that appears when starting the app. Refer to the README for instructions on how to accomplish this.

All usecases make use of the `Main` and `UI` classes

1. Finne navnet på alle rollene en gitt skuespiller har
 - Enter the people menu by choosing people in the first menu.
 - Choose the option *list all roles*
 - Choose an actor. Charles Dance has, among others, a role in the fixtures

The `PersonController` handles the functionality for this usecase.

2. Finne hvilke filmer en skuespiller opptrer i
 - Choose *people* on the first menu
 - Choose *Select a person* in the people menu
 - Select the ID of a person on the list
 - Select the *show all movies* option

The `PersonController` handles the functionality for this usecase.

3. Finne hvilket selskap som lager flest filmer innen hver sjanger
 - Choose the *companies* option in the first menu

-
- Choose the *View compenies by category* option
 - Choose a category

The `CategoryController` handles the functionality for this usecase.

4. Lag en ny film med regissør, manusforfattere, skuespillere og det som hører med.
 - First add all people that is related to the movie(directors, actors etc. Do not worry about specifying what they have done in the movie, this will be added later) by using the insert new option in the *people* menu
 - Choose the *movies* option in the first menu
 - Choose *insert new*
 - Enter a title
 - Enter a release year
 - Enter the release date in the specified format
 - Enter a brief description
 - Choose actors by entering their ID, and then their role
 - When done adding actors, continue by entering 0 in the choose ID prompt
 - Add directors and writers in the same way
 - Select a category from the list, until all categories have been added
 - Enter n when prompted about the movie being an episode
 - Do the same with the series prompt
 - Choose a company
 - Choose whether or not the movie has been released on video
 - Enter the length of the movie in minutes
 - To add music, go back to the first menu and select *movies*
 - Choose *movies* again
 - Select the movie
 - Select movies
 - Select *Add music*

The `FilmController` handles the majority of this usecase. However, it makes use of the `PersonController`, `CategoryController`, `FilmcompanyController` and `MusicController`. for listing the alternatives.

5. Sett inn en ny anmeldelse av en episode
 - Choose the *movies* option in the first menu
 - Choose *series*
 - Choose the filmID of a series
 - Choose *episodes*

-
- Select a season
 - Enter the id of an episode
 - Choose *rate*
 - Choose *New user*
 - Enter a username
 - Enter rating text
 - Enter rating
 - Too see the rating choose *see ratings*

The `FilmController` handles most of the menu logic, yet the `SeasonController` handles most of the logic for listing the season. For the rating logic, the `FilmController` uses a lot of methods from the `UserController`.