

Loppudokumentti

MovieLib

Jaakko Lukkari

jaakko.lukkari@gmail.com

Ohjelmoinninharjoitustyö

15.5.2010

Helsingin Yliopisto Avoin Yliopisto

Sisällysluettelo

Johdanto.....	1
Käsitteet.....	1
Yleiskuvaus.....	1
Käyttöohje.....	1
Asentaminen.....	1
Sovelluksen käyttö.....	1
Sovelluksen käynnistäminen.....	2
Uuden elokuvan lisääminen.....	2
Elokuvan poisto.....	5
Elokuvan muokkaus.....	6
Katsomiskäyttöliittymässä navigoiminen ja elokuvien toisto.....	7
Päävalikko.....	7
Näyttelijät, ohjaajat, genret ja vuodet -valikko.....	8
Rakenne.....	8
Pääkomponentit.....	8
Malli.....	8
Näkymä.....	11
Käsittelijä/Ohjain.....	15
Pakkaukset ja luokat.....	15
org.vatvit.movelib.....	15
org.vatvit.movelib.controllers.....	15
org.vatvit.movelib.dao.....	15
org.vatvit.movelib.database.....	16
org.vatvit.movelib.models.....	16
org.vatvit.movelib.objects.....	16
org.vatvit.movelib.settings.....	16
org.vatvit.movelib.tests.....	17
org.vatvit.movelib.tools.....	17
org.vatvit.movelib.views.....	17
org.vatvit.movelib.views.ui.....	17
org.vatvit.movelib.views.ui.editor.....	18
org.vatvit.movelib.views.ui.editor.panels.....	18
org.vatvit.movelib.views.ui.menu.....	19
org.vatvit.movelib.views.ui.menu.panels.....	19
org.vatvit.movelib.views.ui.player.....	19
Tietosisältö.....	20
Arkkitehtuurikuvaus.....	20
Kerrokset.....	22
Käyttöliittymäkerros.....	22
Sovelluslogiikkakerros.....	22
Säilytyskerros.....	22
Esimerkit.....	24
Käyttöliittymä.....	24
MovieViewImpl.....	24
MovieEditorViewImpl.....	29
Jatkokehitys.....	30
Ohjeita.....	30
Ideoita.....	31
Testaus.....	31
Yksikkötestaus.....	31

Muu testaus.....	31
------------------	----

Johdanto

Tämä dokumentti käsittelee ohjelmoinninharjoitustyö kurssilla toteutetun MovieLib sovelluksen toimintaa ja rakennetta. Dokumentti sisältää yleiskuvauksen sovelluksesta ja sen toiminnasta, käyttöohjeet ja kuvauksen sovelluksen rakenteesta.

Käsitteet

Alle oleva taulukko sisältää selitykset tässä dokumentissa esiintyvistä käsitteistä.

VLC	VLC media player on avoin ja alustariippumaton mediasoitin
MVC	Sovelluksen suunnittelussa käytetty suunnittelumalli.
TMDb	Yhteisövoimin ylläpidetty elokuvatietokanta. Palvelusta on mahdollista hakea kaikki tiedot liittyen elokuvaan ja sen näytteisiin
SQLite	Pieni, suoraan sovellukseen linkitettävä relaatiokanta. Käytetään monissa eri sovelluksissa. Arvioiden mukaan maailman käytetyin tietokanta.
DAO	DAO eli Data Access Object. Palauttaa pyydettyä tietokannassa olevia tietoja olioina.
Data transfer -objekti (DTO)	Yleisesti DAO:n palauttama tiedon kuljetukseen tarkoitettu objekti, joka ei sisällä omaa toiminnallisuutta.

Yleiskuvas

MovieLib on elokuvien arkistointiin ja arkistoitujen elokuvien katsomiseen tarkoitettu sovellus. Sovellus säilöo käyttäjän lisäämän elokuvan tiedot tietokantaan ja kopioi videotiedoston, tekstitykset ja kansikuvan omaan tietovarastoonsa. Käyttäjä pystyy selaamaan ja katsomaan arkistossa olevia elokuvia kokoruutuun aukeavalla käyttöliittymällä. Elokuvien lisäys, poisto ja muokkaus suoritetaan erillisen käyttöliittymän kautta. Sovelluksen jatkokehitykseen on varauduttu toteuttamalla järjestelmä mahdollisimman modulaarisesti ja toteuttamalla sekä malli että näkymä rajapintojen avulla. Sovelluksen toiminta on videoiden katsomisen osalta riippuvainen libvlc-kirjastosta.

Käyttöohje

Tässä osiossa käsitellään sovelluksen asennukseen ja käyttöön liittyviä ohjeistuksia.

Asentaminen

Sovelluksen käyttö

Tässä osiossa käsitellään sovelluksen toimintoihin liittyviä käyttöohjeita.

Sovelluksen käynnistäminen

Sovelluksen käynnistäminen onnistuu parhaiten käyttämällä movielib.jar -pakettia.

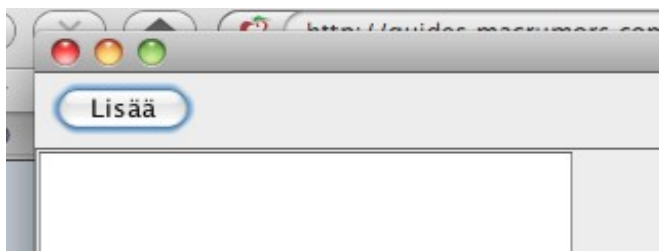
Hallinta: java -jar movielib.jar editor

Katsominen: java -jar movielib.jar

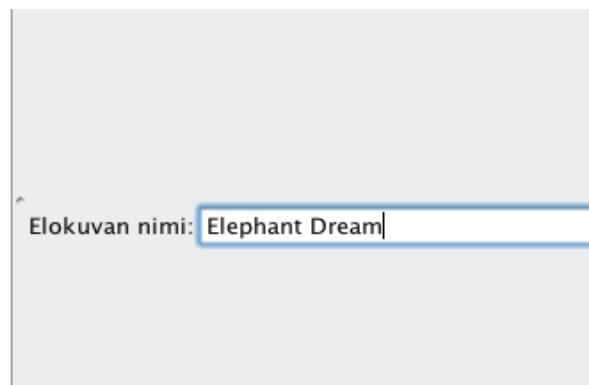
Komento tulee suorittaa komentorivillä kansiossa, johon movielib -sovelluksen tiedostot on purettu.

Uuden elokuvan lisääminen

Käynnistä hallintapaneeli ”Sovelluksen käynnistys” -osiossa mainitulla tavalla.



Kuva 1: Paina sovelluksen vasemmassa yläreunassa olevaa Lisää -painiketta.



Kuva 2: Täytä ”Elokuvan nimi:” - kenttään lisäämäsi elokuvan nimi.



Kuva 3: Valitse ”Hae tiedot” -mikäli haluat hakea elokuvan tiedot käyttäen TMDb -palvelua. Mikäli haluat täyttää tiedot kokonaisuudessaan itse, valitse ”Seuraava”.

Perustiedot

Nimi:

Slogan:

Ohjaaja:

Vuosi:

Trailer (Youtube):

Arvostelu (0.0 – 10.0):

Kuvaus:

Kuva 4: Tarkista ja tarvittaessa muokkaa näkymässä esittettävät tiedot ja paina "Seuraava".

Genret

Animation
Fantasy
Science Fiction
Open Source

Nimi:

Kuva 5: Poista, lisää tai muokkaa haluamiasi genrejä ja paina "Seuraava".

Näyttelijät ja tuotantoryhmä

Proog (voice) - Tygo Gernandt
 Emo (voice) - Cas Jansen
 Author - Andreas Goralczyk
 Author - Bassam Kurdali
 Screenplay - Pepijn Zwanenberg

Nimi:
 Rooli/Työ:
 Näyttelijä: ☐

Kuva 6: Poista, lisää tai muokkaa haluamiesi näyttelijöiden (tai muiden tuotantoryhmään kuuluvien henkilöiden) tietoja ja paina "Seuraava".

Tekstitykset

Otsikko: Suomi
 Kieli: fi
 Tiedosto:

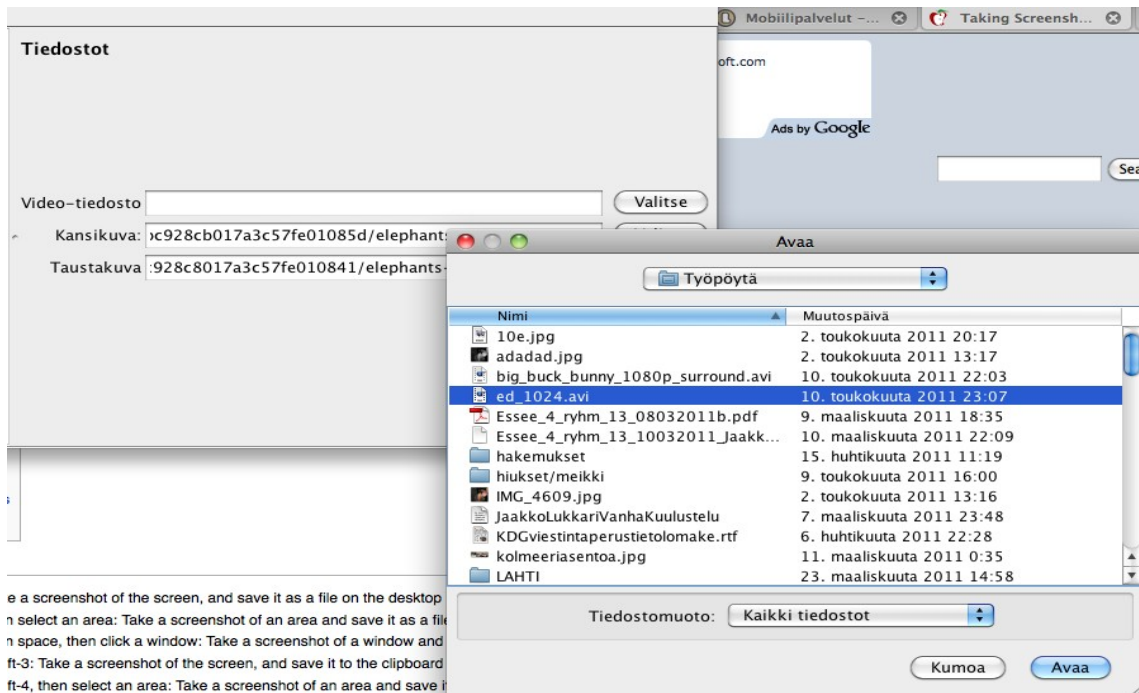
Ava

Elephants.Dream.2006.div...

Nimi	Muutospäivä
elephantsdream-1024-mpeg4-su-ac3.sub	23. joulukuuta 200...
elephantsdream-1920-hd-mpeg4-su-ac3.sub	23. joulukuuta 200...

Tiedostomuoto: Kaikki tiedostot

Kuva 7: Lisää elokuvalle haluamasi tekstitystiedostot ja paina "Seuraava".



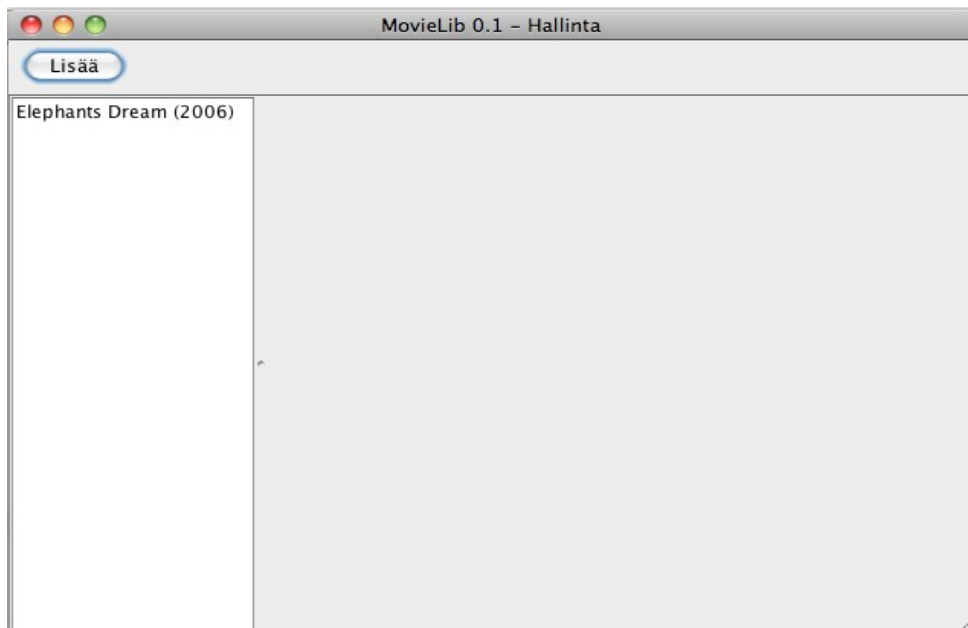
Kuva 8: Aseta elokuvan videotiedosto. Mikäli et ole käyttänyt elokuvan tietojen hakuun TMDb -palvelua, aseta myös ainakin elokuvan kansikuva. Kun olet asettanut elokuvan tiedostot paina ”Valmis” tallettaaksesi elokuvan. Sovellus ilmoittaa, kun tallennus on valmis. Tallennuksen kesto riippu videotiedoston koosta.

Mahdolliset virhetilanteet

Mikäli käyttäjä ei valitse videotiedostoa tai tiedostoa lisäämälleen teksityksille antaa sovellus tallennuksen yhteydessä virheilmoituksen ”(No such file or directory)”.

Elokuvan poisto

Käynnistä hallintapaneeli ”Sovelluksen käynnistys” -osiossa mainitulla tavalla.



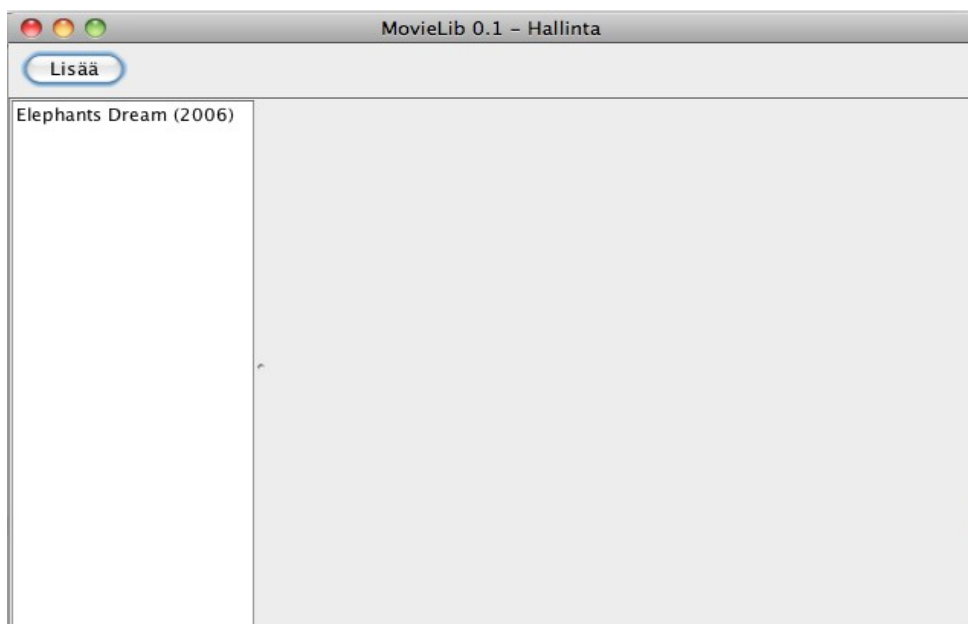
Kuva 9 : Valitse listasta elokuva, jonka haluat poistaa.



Kuva 10: Valitse "Poista". Sovellus ilmoittaa, kun poisto on suoritettu.

Elokuvan muokkaus

Käynnistä hallintapaneeli "Sovelluksen käynnistys" -osiossa mainitulla tavalla.



Kuva 11: Valitse listasta elokuva, jota halua muokata.



Kuva 12: Valitse "Muokkaa" ja nouda "Elokuvan lisäys" -ohjeessa mainittuja ohjeita.

Katsomiskäyttöliittymässä navigoiminen ja elokuvien toisto

Elokuvien katsominen tapahtuu katsomis-käyttöliittymän kautta. Käyttöliittymässä navigoidaan käyttäen nuolinäppäimiä sekä esc- ja enter-painikkeita.

Päävalikko



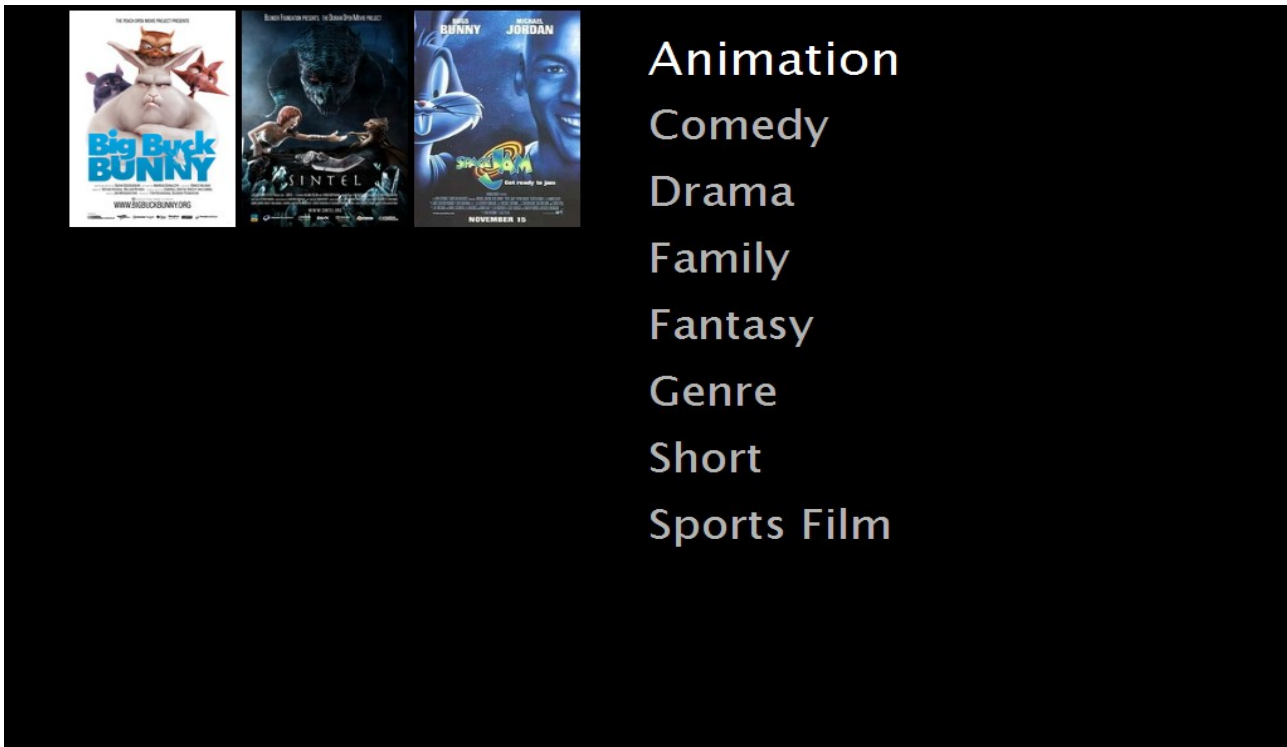
Kuva 13: Päävalikko

Päävalikossa listataan eri tavat joiden perusteella elokuvia pystyy listaamaan. Mikäli käyttäjä esimerkiksi valitsee valinnan "Genret", listataan käyttöliittymään kaikki tietokannassa olevat genret.

Navigointi

Vasemmalle tai ESC	Lopettaa sovelluksen suorituksen.
Oikealle tai ENTER	Siirtyy valittuun näkymään tai mikäli valittuna on kohta "Lopeta", lopettaa sovelluksen suorituksen.
Ylös ja alas	Liikkuminen valikossa.

Näyttelijät, ohjaajat, genret ja vuodet -valikko



Kuva 14: Näyttelijät, ohjaajat, genret ja vuodet -valikko

Esimerkkinä Genret -valikko. Valikossa listataan kaikki tietokannassa olevat vaihtoehdot. Näkymän vasemmassa palstassa näytetään maksimissaan 24 elokuvan ote kategorian sisältämistä elokuvista.

Navigointi

Vasemmalle tai ESC	Paluu takaisin päävalikkoon.
Oikealle tai ENTER	Siirtyy valittuun kategoriaan.
Ylös ja alas	Liikkuminen valikossa.

Elokuvat -valikko



Big Buck Bunny (2008)

Sintel (2010)

Space Jam (1996)

Big Buck Bunny (2008)

7.6 / 10

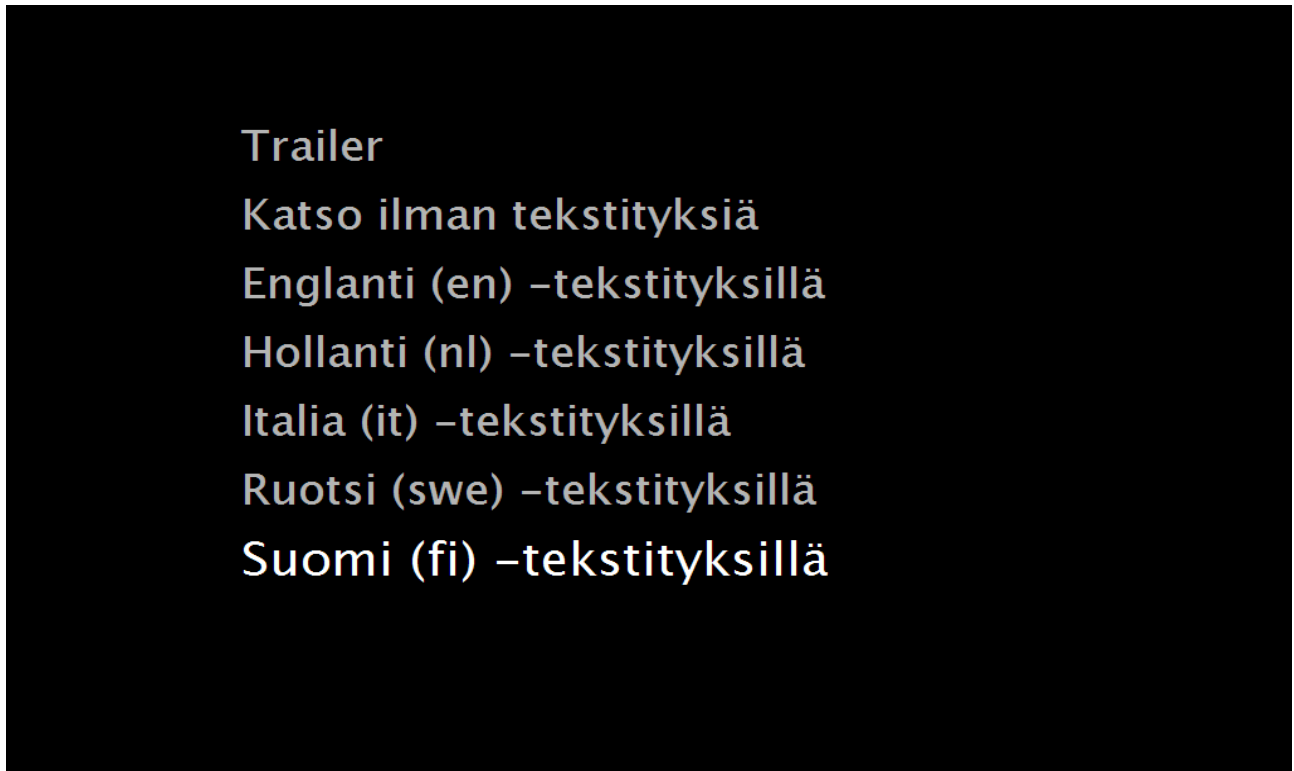
Follow a day of the life of Big Buck Bunny when he meets three bullying rodents: Frank, Rinky, and Gamera. The rodents amuse themselves by harassing helpless creatures by throwing fruits, nuts and rocks at them. After the deaths of two of Bunny's favorite butterflies, and an offensive attack on Bunny himself, Bunny sets aside his gentle nature and orchestrates a complex plan for revenge.

Kuva 15: Elokuvat -valikko

Elokuvat -valikossa listataan joko kaikki elokuvat tai valitun kategorian elokuvat.

Navigointi

Vasemmalle tai ESC	Paluu takaisin päävalikkoon tai kategorian (näyttelijät, ohjaajat, genret ja buoden) listaukseen.
Oikealle tai ENTER	Siirtyminen valitun elokuvan toistovalikkoon.
Ylös ja alas	Liikkuminen valikossa.

Toistoasetusvalikko*Kuva 16: Toistoasetusvalikko*

Toistoasetusvalikossa käyttäjän on mahdollista valita minkä kielisillä tekstityksillä toistaminen tapahtuu. Käyttäjän on myös mahdollista valita haluaako hän katsoa elokuvan esittelyvideon (trailer).

Navigointi

Vasemmalle tai ESC	Paluu takaisin elokuvien listaukseen.
Oikealle tai ENTER	Elokuvan toiston aloitus valituilla asetuksissa.
Ylös ja alas	Liikkuminen valikossa.

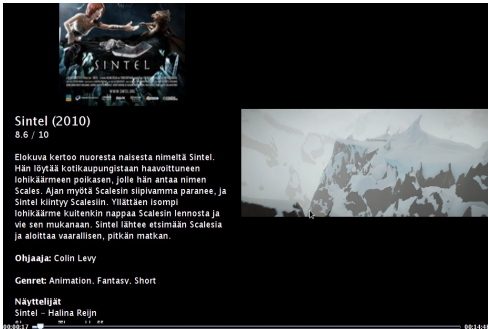
Toistonäkymä



Kuva 17: Toistonäkymä

Elokuvan toistaminen aloitetaan automaattisesti.

Navigointi

ESC	Paluu takaisin toistoasetusvalikkoon.
SPACE	Elokuvan toiston pysäytys tai jatkaminen.
Vasemmalle ja oikealle	Elokuvan kelaus.
ENTER	 <p>Elokuvan tietojen ja keston näyttö tai piilotus.</p>

Rakenne

Rakenteen osalta ollaan pystytty pysymään täysin suunnitteludokumentissa määrätyissä

suunitelmissa. Poikkeuksena mallin metodien määrittelyihin lisätyt poikkeukset.

Pääkomponentit

Tässä osiossa tarkastellaan järjestelmän pääkomponentteja ja niitä edustavia luokkia. Sovellus koostuu kolmesta pääkomponentista, joita ovat malli, näkymä ja ohjain. Sovelluksen käynnistyessä asetuksissa määritetyt tai parametreina annetut malli ja näkymä ladataan sovellukseen tehdyn luokkalataimen avulla. Tämän jälkeen malli ja näkymä annetaan ohjaimelle, joka asetaa näkymälle tarvittavat kuuntelijat. Arkkitehtuurin ja pääkomponenttien osalta noudatetaan MVC -suunnittelumallia.

Malli

Malli on tietorakenneluokka ja toimii sovelluksen tietolähteenä. Mallin on yhteydessä tietokantaan DAO -luokan kautta. Malli sisältää myös hieman normaalista MVC -arkkitehtuurista poiketen sovelluslogiikkaan liittyvää toiminnallisuutta. Tämä siksi, että sovelluksen tietosisältö ja sen hallinta on helposti vaihdettavissa toiseen mallin rajapintaa toteuttavaan luokkaan. Sovelluksen yleisluonteen takia mallin ei myöskään tarvitse informoida ohjainta mallissa tapahtuneista muutoksista.

- **MovieModel** – Rajapinta tietolähteitä varten.
- **MovieModelImpl** – MovieModel -rajapinnan toteuttava luokka, joka hakee ja säilöo elokuvan tiedot SQLite -tietokantaan ja siirtää elokuvaan liittyvät media -tiedostot niille varattuihin kansioihin.

MovieModel -rakenne

```
/**
 * Rajapinta, jonka toteuttavia luokkia voidaan käyttää sovelluksen
 * tietolähteinä.
 */
public interface MovieModel {

    /**
     * Elokuvatietojen kenttiä edustava enumeraali
     */
    public enum Field {
        ID, TITLE, DESCRIPTION, SLOGAN, DIRECTOR, TRAILER_URL, YEAR, RATING, ADDED, GENRE,
        CAST
    }

    /**
     * Kansi ja taustakuvan kokoja edustava enumeraali
     */
    public enum ImageSize {
        SMALL, MEDIUM, LARGE
    }

    /**
     * @return Palauttaa tietolähteen nimen
     */
    public String getModelName();

    /**
     * Poistaa elokuvan tietovarastosta.
     */
}
```

```

*
* @param movie
*      poistettava elokuva
* @return totuusarvo siitä onnistuiko poisto
* @throws MovieDeleteException elokuvan poistossa tapahtunut virhe
*/
public boolean deleteMovie(Movie movie) throws MovieDeleteException;

/**
* Palauttaa id:tä vastaavan elokuvan tiedot
*
* @param id
*      elokuvan id
* @return Movie -luokan ilmentymä täytettynä elokuvan tiedoilla.
* @throws MovieDataRetrieveException elokuvan tietojen haussa tapahtunut virhe
*/
public Movie getMovie(int id) throws MovieDataRetrieveException;

/**
* Elokvien haku.
*
* @param searchField
*      Kenttä, jonka perusteella haku tapahtuu
* @param search
*      Haettava sisältö
* @param limit
*      Vastauksien raja
* @param page
*      Monenneltako sivulta vastaukset halutaan
* @param orderBy
*      Kenttä, jonka perusteella vastaukset järjestetään
* @param desc
*      Järjestetäänkö laskevassa järjestyksessä
* @param includeCast
*      Liitetäänkö vastauksiin myös näyttelijät ja muu tuotantoryhmä
* @param includeGenres
*      Liitetäänkö vastauksiin myös genret
* @param includeSubtitles
*      Liitetäänkö vastauksiin myös tekstitykset
* @return QueryResult instanssi, joka sisältää tiedot käytetyistä hakuohdoista ja haun vastauksen.
* @throws MovieDataRetrieveException elokuvan tietojen haussa tapahtunut virhe
*/
public QueryResult<Movie> getMovies(Field searchField, String search,
    int limit, int page, Field orderBy, boolean desc,
    boolean includeCast, boolean includeGenres, boolean includeSubtitles) throws
MovieDataRetrieveException;

/**
* Lisää elokuvan tietovarastoon.
*
* @param movie
*      Elokuvan tiedot movie -oliossa.
* @param cover
*      Elokuvan kansikuva
* @param backdrop
*      Elokuvan taustakuva
* @return Lisätyn elokuvan id.
* @throws MovieException elokuvan tallennuksessa tapahtunut virhe
*/
public int saveMovie(Movie movie, BufferedImage cover,
    BufferedImage backdrop) throws MovieException;

/**
* Palauttaa elokuvan taustakuvan halutussa koossa

```



```

*
* @param movie
*         halutun elokuvan tiedot
* @param size
*         koko
* @return BufferedImage elokuvan taustakuvasta
*/
public BufferedImage getBackdropForMovie(Movie movie, ImageSize size);

/**
 * Palauttaa elokuvan kannen halutussa koossa
 *
 * @param movie
 *         halutun elokuvan tiedot
 * @param size
 *         koko
 * @return BufferedImage elokuvan kannesta
 */
public BufferedImage getCoverForMovie(Movie movie, ImageSize size);

/**
 * @return onko tietolähteen elokuvia salittu muokattavan
 */
public boolean allowEdit();

/**
 * @return onko tietolähteeseen salittu elokuvien lisääminen
 */
public boolean allowImport();

/**
 * @return onko tietolähteestä sallittu elokuvien poistaminen
 */
public boolean allowDelete();

/**
 * @return kaikki tietolähteessä olevat näyttelijät ja muut tuotantoryhmän
 *         jäsenet
 * @throws MovieDataRetrieveException elokuvan tietojen haussa tapahtunut virhe
 */
public ArrayList<String> getAllCast() throws MovieDataRetrieveException;

/**
 * @return kaikki tietolähteessä olevat ohjaajat
 * @throws MovieDataRetrieveException elokuvan tietojen haussa tapahtunut virhe
 */
public ArrayList<String> getAllDirectors() throws MovieDataRetrieveException;

/**
 * @return kaikki tietolähteessä olevat genret
 * @throws MovieDataRetrieveException elokuvan tietojen haussa tapahtunut virhe
 */
public ArrayList<String> getAllGenres() throws MovieDataRetrieveException;

/**
 * @return kaikki tietolähteessä olevat tuotantovuodet
 * @throws MovieDataRetrieveException elokuvan tietojen haussa tapahtunut virhe
 */
public ArrayList<String> getAllYears() throws MovieDataRetrieveException;
}

```

Näkymä

Näkymä edustaa sovelluksen käyttöliittymiä. Mallin tavoin näkymänä voi toimia mikä tahansa

MovieView -rajanpinnan toteuttava luokka. Sovelluksen määrittelyn mukaisesti toteutettavia näkymiä ovat elokuvien katsomiseen tarkoitettu näkymä ja elokuvien muokkaamisen tarkoitettu näkymä.

- **MovieView** – Rajapinta käyttöliittymiä varten
- **MovieViewImpl** – MovieView rajanpinnan toteuttava luokka, joka on elokuvien katsomiseen tarkoitettu käyttöliittymä.
- **MovieEditorViewImpl** - MovieView rajanpinna toteuttava luokka, joka on elokuvien muokkaamiseen ja lisäämiseen tarkoitettu käyttöliittymä.

MovieView -rakenne

```
/**
 * Rajapinta, jonka toteuttavia luokkia voidaan käyttää sovelluksen
 * näkyminä.
 */
public interface MovieView {

    // KUUNTELIJOIDEN ASETUS
    /**
     * Asettaa kuuntelijan joka tarkkailee sovelluksen sulkupyynnöjä
     *
     * @param listener
     */
    public void setCloseListener(CloseListener listener);

    /**
     * Asettaa kuuntelijan joka tarkkailee elokuvaan liittyviä tapahtumia
     *
     * @param listener
     */
    public void setMovieListener(MovieListener listener);

    /**
     * Asettaa kuuntelijan joka tarkkailee näyttelijöiden/tuotantotiimin
     * tapahtumia
     *
     * @param listener
     */
    public void setCastListener(GroupListener listener);

    /**
     * Asettaa kuuntelijan joka tarkkailee ohjaajien tapahtumia
     *
     * @param listener
     */
    public void setDirectorListener(GroupListener listener);

    /**
     * Asettaa kuuntelijan joka tarkkailee vuosien
     *
     * @param listener
     */
    public void setYearListener(GroupListener listener);

    /**
     * Asettaa kuuntelijan joka tarkkailee genrejen tapahtumia
     *
     * @param listener
     */
}
```

```

public void setGenreListener(GroupListener listener);

// NÄYTÄ

/**
 * Pyytää käyttöliittymää näyttämään parametrinä annetun listan
 * näyttelijöistä ja muista tuotannon jäsenistä
 *
 * @param cast
 *         lista näyttelijöistä tai vst.
 */
public void showCast(ArrayList<String> cast);

/**
 * Pyytää käyttöliittymää näyttämään parametrinä annetun listan genreistä
 *
 * @param genres
 *         lista genrejä
 */
public void showGenres(ArrayList<String> genres);

/**
 * Pyytää käyttöliittymää näyttämään parametrinä annetun listan ohjaajia
 *
 * @param directors
 *         lista ohjaajia
 */
public void showDirectors(ArrayList<String> directors);

/**
 * Pyytää käyttöliittymää näyttämään parametrinä annetun listan
 * tuotantovuosista
 *
 * @param years
 *         lista tuotantovuosia
 */
public void showYears(ArrayList<String> years);

/**
 * Pyytää käyttöliittymää näyttämään parametrinä annetun listan elokuvista
 *
 * @param movies
 *         lista elokuvia
 */
public void showMovies(ArrayList<Movie> movies);

/**
 * Pyytää käyttöliittymää näyttämään parametrinä annetun elokuvan teidot
 *
 * @param movie
 *        okuva
 */
public void showMovie(Movie movie);

/**
 * Pyytää käyttöliittymää näyttämään esikatselmuksen elokuvista joissa
 * näyttelijä tai vst. on
 *
 * @param movies
 *         ote ryhmään kuuluvista elokuvista
 */
public void showCastPreview(ArrayList<Movie> movies);

/**
 * Pyytää käyttöliittymää näyttämään esikatselmuksen elokuvista jotka

```

```

    * kuuluvat genreen
    *
    * @param movies
    *         ote ryhmään kuuluvista elokuvista
    */
    public void showGenrePreview(ArrayList<Movie> movies);

    /**
     * Pyytää käyttöliittymää näyttämään esikatselmuksen elokuvista jotka on
     * ohjannut parametrinä annettu ohjaaja
     *
     * @param movies
     *         ote ryhmään kuuluvista elokuvista
     */
    public void showDirectorPreview(ArrayList<Movie> movies);

    /**
     * Pyytää käyttöliittymää näyttämään esikatselmuksen elokuvista jotka on
     * tuotettu kyseisenä
     *
     * @param movies
     *         ote ryhmään kuuluvista elokuvista
     */
    public void showYearPreview(ArrayList<Movie> movies);

    // Käynnistyts

    /**
     * Kertoo käyttöliittymälle, että kaikki muu on valmista
     */
    public void init();

    // VIRHEET

    /**
     * Näyttää virheilmoitukset
     *
     * @param error
     *         näytettävä virheilmoitus
     */
    public void showError(String error);

    // Lataus
    /**
     * Näyttää käyttäjälle että latautuminen tapahtuu
     *
     * @param reason
     */
    public void showLoading(String reason);

    /**
     * Piilottaa latautumis indikaattorin
     */
    public void hideLoading();

    // Tiedot

    /**
     * @return kyseisen näkymän/ulkoasun nimi
     */
    public String getViewName();

    /**
     * @return valittuna oleva genre
     */

```

```

public String getGenre();
/**
 * @return valittuna oleva näyttelijä tai muu tuotantoryhmän jäsen
 */
public String getCast();
/**
 * @return valittuna oleva ohjaaja
 */
public String getDirector();
/**
 * @return valittuna oleva vuosi
 */
public String getYear();
/**
 * @return valittuna oleva elokuva
 */
public Movie getMovie();
/**
 * @return uudelle elokuvalle valittu kanssi
 */
public BufferedImage getCover();
/**
 * @return uudelle elokuvalle valittu tausta
 */
public BufferedImage getBackdrop();
}

```

Käsittelijä/Ohjain

Käsittelijän/ohjaimen tehtävänä on toimia käyttäjän käskyjen perusteella ja muuttaa mallia ja/tai näkymää niiden mukaisesti. Ohjain ei itsessään sisällä toiminnallisuutta vaan toimii eräänlaisena siltana mallin ja näkymän välillä. Tämä mahdollistaa vaivattomamman jatkokehityksen ja käyttöliittymän tai tietolähteen vaihtamisen.

Ohjaimen julkinen rakenne

```

public MovieController(MovieModel model, MovieView view);
public MovieModel getModel();
public void setModel(MovieModel model);
public MovieView getView();
public void setView(MovieView view);

```

Pakkaukset ja luokat

Tässä kappaleessa käsitellään sovelluksen luokkia ja niiden jakautumista eri paketteihin.

org.vatvit.movieslib

Sovelluksen pääpakkaus.

- **Main**
Sovelluksen toiminnan käynnistävä luokka. Luokka lataa asetuksissa tai argumentteinä määritetyn mallin (MovieModel) ja näkymän (MovieView) ja luo uuden ohjain luokan (MovieController) antaen tämän konstruktorille attribuutteina mallin ja näkymän.

org.vatvit.movielib.controllers

Pakkaus sisältää sovelluksen toimitaa ohjaavan MovieController luokan.

- **MovieController**
Sovelluksen toimintaa ohjaava luokka. Ohjain saa luonnin yhteydessä mallin ja näkymän. Ohjain asettaa näkymälle kuuntelijat, joiden kautta tuleiden tapahtumien perusteella ohjain muokkaa mallia ja päivittää tietoja näkymälle.

org.vatvit.movielib.dao

Pakkaus sisältää tietojen aksessointiin tietokannasta käytettävät luokat.

- **MovieDAO**
Tietokannasta tietoja hakeva ja muokkaava DAO -luokka. Luokan vastuulla on elokuvien tietojen säilöntä.
- **QueryResult<T>**
Geneerinen tietokannan hakuvastaus. Pitää sisällään tiedot käytetyistä hakuehdoista ja hakuvaustauksista.

org.vatvit.movielib.database

Pakkaus sisältää tietokanta yhteyden luomiseen käytettävän DatabaseConnection luokan.

- **DatabaseConnection**
Luo tietokantayhteyden SQLite -tietokantaan. Totettu sigletonina eli palauttaa aina saman yhteyden.

org.vatvit.movielib.models

Pakkaus sisältää sovelluksen mallin rajapinnan, sen toteuttavan luokan ja mallissa käytettävät poikkeukset.

- **MovieDataRetrieveException**
Elokvatietojen haussa tapahtuvia virheitä edustava Exception -luokka. Laajentaa MovieException -luokkaa.
- **MovieDeleteException**
Elokuvan poistossa tapahtuvia virheitä edustava Exception -luokka. Laajentaa MovieException -luokkaa.
- **MovieException**
Geneerinen elokuvaan liittyviä virheitä edustava Exception -luokka.
- **MovieModel**
Rajapinta, jonka toteuttavia luokkia voidaan käyttää sovelluksen tietolähteinä.
- **MovieModelImpl**
Tietokantaa tietovarastonaan käyttävä toteutus MovieModel rajapinnasta. Säilöo elokuvien tiedot tietokantaan ja siirtää media-tiedostot data kansioon.

org.vatvit.movielib.objects

Pakkaus sisältää sovelluksessa käytettävät data transfer -objektit

- **Movie**
Yksittäistä elokuvaa edustava luokka.

- **MovieCast**
Yksittäistä elokuvassa esiintyvää näyttelijää tai muuta tuotantorymään kuuluvaa työntekijää edustava luokka.
- **MovieGenre**
Yksittäistä elokuvan genreä edustava luokka.
- **MovieInfoResult**
Yksittäistä MovieInfoTools luokan TMDb:stä hakemaa elokuvatietoa ja kansi- ja taustakuvaa edustava luokka.
- **MovieSubtitle**
Yksittäistä elokuvan tekstitystä edustava luokka.

org.vatvit.movieslib.settings

Pakkaus sisältää asetusten lataamiseen ja yksittäisten asetusten hakemiseen tarkoitettua luokkaa.

- **SettingsLoader**
Asetusten lataamisen JSON -muotoisesta asetustiedosta tarkoitettu luokka. Luokan avulla asetustiedot voidaan käyttää puumallisia JSON -dokumentteja, jotka mahdollistavat informatiiviset ja helposti hallinnoitavat asetustiedostot.

org.vatvit.movieslib.tests

Pakkaus sisältää sovelluksen toimintaa testaavat yksikkötestit

- **DatabaseConnectionTest**
DatabaseConnection luokan toimintoja testaava yksikkötesti
- **MovieDaoTest**
MovieDao luokan toimintoja testaava yksikkötesti
- **MovieModelImplTest**
MovieModelImpl luokan toimintoja testaava yksikkötesti
- **ToolTest**
Aputyökaluluokkien EqualsTools, FileTools, ImageTools ja MovieInfoTools toimintoja testaava luokka.

org.vatvit.movieslib.tools

Pakkaus sisältää vaihteleviin käyttötarkoituksiin käytettäviä aputyökaluja.

- **EqualsTools**
MovieCast, MovieGenre ja MovieSubtitle ArrayListien vertaamiseen tarkoitettu luokka. Luokan avulla voidaan tarkistaa ovat kahden ArrayList:n sisältämät objektit ja järjestys sama.
- **FileTools**
Tiedostojen kopiointiin tarkoitettu luokka.
- **ImageTools**
Kuvatiedostojen koon muuttamiseen tarkoitettu luokka.
- **MovieInfoTools**
Elokuvien tietojen haku TMDb -palvelusta tarkoitettu luokka. Luokka muuntaa TMDb-muotoiset elokuvatiedot sovelluksen käyttämään muotoon.

org.vatvit.movelib.views

Pakkaus sisältää sovelluksen käyttöliittymään liittyvät luokat.

- **CloseListener**
Käyttöliittymän/sovelluksen sulkupyynnöjä kuunteleva tapahtumakuuntelija.
- **GroupListener**
Elokuva käyttöliittymässä näkyvien ryhmittelyiden ohjaaja, näyttelijät, genret ja vuodet tapahtumia kuunteleva tapahtumakuuntelija.
- **MovieListener**
Elokuvaan liittyviä tapahtumia kuunteleva tapahtumakuuntelija.
- **MovieView**
Rajapinta, jonka toteuttavia luokkia voidaan käyttää sovelluksen näkyminä.

org.vatvit.movelib.views.ui

Pakkaus sisältää MovieView -rajapinnan toteuttavat käyttöliittymä luokat ja niihin liittyvät komponentti luokat.

- **MovieViewImpl**
Elokuvien katsomiseen tarkoitettu kokoruudun käyttöliittymä. Toteuttaa MovieView -rajapinnan. Käyttöliittymä on toteutettu päällekkäin kasautuvista näkymistä. Layout:ina on käytetty CardLayoutia.
- **View**
MovieViewImpl käyttöliittymässä käytettävien näkymien yläluokkana käytettävä luokka. Laajentaa Jpanelia.

org.vatvit.movelib.views.ui.editor

Pakkaus sisältää elokuvien hallintaan tarkoitetun käyttöliittymän ja siihen liittyvät komponentti luokat.

- **MovieEditorViewImpl**
Elokuvien muokkaamiseen ja lisäämiseen tarkoitettu käyttöliittymä. Toteuttaa MovieView -rajapinnan.
- **MovieWizardPanel**
Uuden tai jo tietokannassa olevan elokuvan tietojen muokkaamiseen tarkoitettu paneeli. Paneelin näkymät on toteutettu omina päällekkäin kasaantuvina Jpanelia laajentavina luokkina. Toteutukseen on käytetty CardLayout layoutia.

org.vatvit.movelib.views.ui.editor.panels

- **FileSelectPanel**
Tiedostonvalinta kenttien ja painikkeiden luontia helpottava paneeli. Sisältää tiedostoavaimen avaavan painikkeen ja tekstikentän johon valitun tiedoston polku tulostetaan/asetetaan.
- **MovieCastPanel**
MovieWizardPanelin näkymä, jotka käytetään elokuvan näyttelijöiden ja muiden tuotantotyöntekijöiden tietojen muokkaamiseen ja lisäämiseen.
- **MovieDetailsPanel**

MovieWizardPanelin näkymä, jotka käytetään elokuvan yleistietojen (nimi, ohjaaja, slogan, trailer, vuosi ja kuvaus) muokkaamiseen.

- **MovieFilesPanel**
MovieWizardPanelin näkymä, jotka käytetään elokuvan media-tiedostojen (kansi- ja taustakuva ja videotiedosto) valitsemiseen.
- **MovieGenrePanel**
MovieWizardPanelin näkymä, jotka käytetään elokuvan genrejen muokkaamiseen ja lisäämiseen.
- **MovieInformationSelectPanel**
MovieWizardPanelin näkymä, jotka käytetään tilanteissa joissa elokuvan tietojen haku palauttaa monia eri vaihtoehtoja.
- **MovieNamePanel**
MovieWizardPanelin näkymä, jotka käytetään elokuvan nimen täyttämiseen ja esitietojen hakuun elokuvan nimen perusteella.
- **MovieSubtitlesPanel**
MovieWizardPanelin näkymä, jotka käytetään elokuvan tekstitystiedostojen muokkaamiseen ja lisäämiseen.

org.vatvit.movielib.views.ui.menu

Pakkaus sisältää MovieViewImpl käyttöliittymässä käytettävät näkymät ja komponentit.

- **BasicMenu**
Yksinkertainen valikkonäkymä, joka sisältää vasemmassa palstassa näytettävän infopaneelin sekä oikeassa palstassa näkevät valinta vaihtoehdot. Laajentaa View -luokkaa.
- **MenuList**
Jlist -komponenttia laajentava kokoruututilaa ja näppäimistöohjaista varten optimoitu lista.
- **MenuMessage**
Virheilmoitusten näyttöä varten tarkoitettu näkymä.
- **MovieGroupMenu**
Elokuva ryhmittelyiden listaamista varten tarkoitettu näkymä. Yläluokastaan BasicMenu:sta poiketen infotaulussa näytetään ote ryhmän sisältämistä elokuvista.
- **MovieSettingsMenu**
Elokuvan katsomisasetusten (katsominen ilman tekstityksiä, tekstitysten valinta ja esittelyvideo) valintaan tarkoitettu valikkonäkymä. Laajentaa muista valikoista poiketen suoraan View:iä.
- **MoviesMenu**
Elokuvien listaukseen tarkoitettu valikkonäkymä. Laajentaa BasicMenu:a.
- **PlayInformation**
MovieSettingsMenu toistovalintojen tiedot sisältävä luokka.

org.vatvit.movielib.views.ui.menu.panels

Pakkaus sisältää MovieViewImpl:ssä käytettävien näkymien paneelit.

- **MenuImage**
Paneeli mahdollistaa kuvatiedostojen näytön käyttöliittymässä. Laajentaa Jpanelia.
- **MovieDetails**
Paneeli esittää sille annetun Movie -olion tiedot sekä elokuvan kansikuvan. Laajentaa Jpanelia.
- **MovieInfo**
Elokuvan tietoja ja kansikuvaa esittelevä automaattisesti vierivä paneeli. Laajentaa JScrollPane:a.

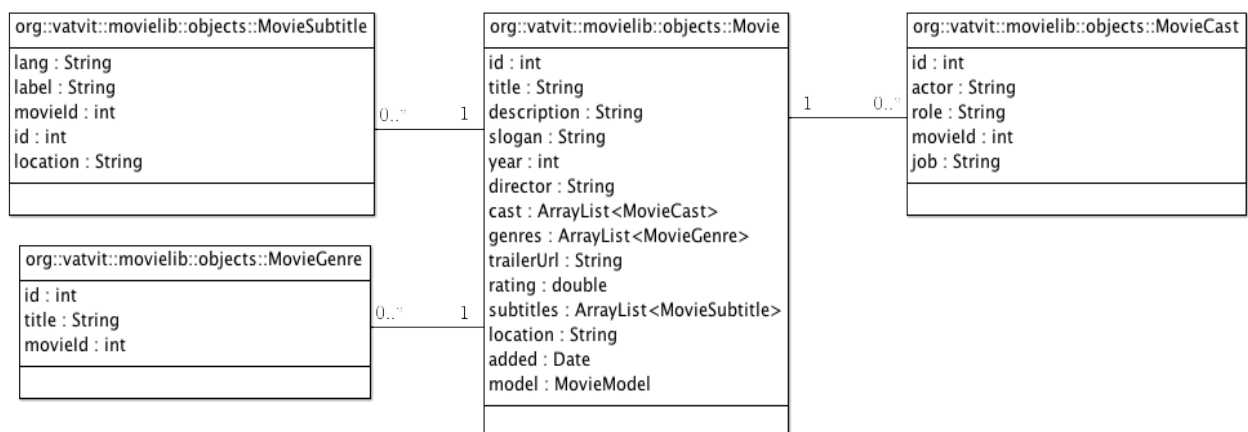
org.vatvit.movieslib.views.ui.player

- **PlayDetails**
Paneeli, joka näyttää elokuvan toistotilanteen (paljonko toistettu ja kokonaiskesto). Paneeli sisältää myös toistotilannetta kuvaavan Jslider:n.
- **Player**
Elokuvan toistamisesta vastaava paneeli. Paneeli sisältää Canvas:n, jolle elokuva tulostetaan, elokuvan tietoja näyttävän MovieInfo paneelin sekä elokuvan toistotilannetta esittävän PlayDetails paneelin.

Tietosisältö

Tässä osiossa käsitellään sovelluksen tietosisältöä.

Sovellus on tietosisällöltään yksinkertainen. Tietokantaan säilötään elokuvan (**Movie**) tiedot, elokuvan näyttelijöiden (**MovieCast**) tiedot, elokuvan luokitukset (**MovieGenre**) sekä elokuvana liitetyt tekstitykset (**MovieSubtitle**).



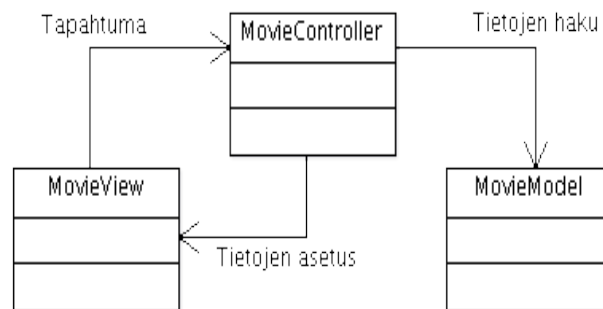
Kuva 18: Tietosisältö

Arkkitehtuurikuvaus

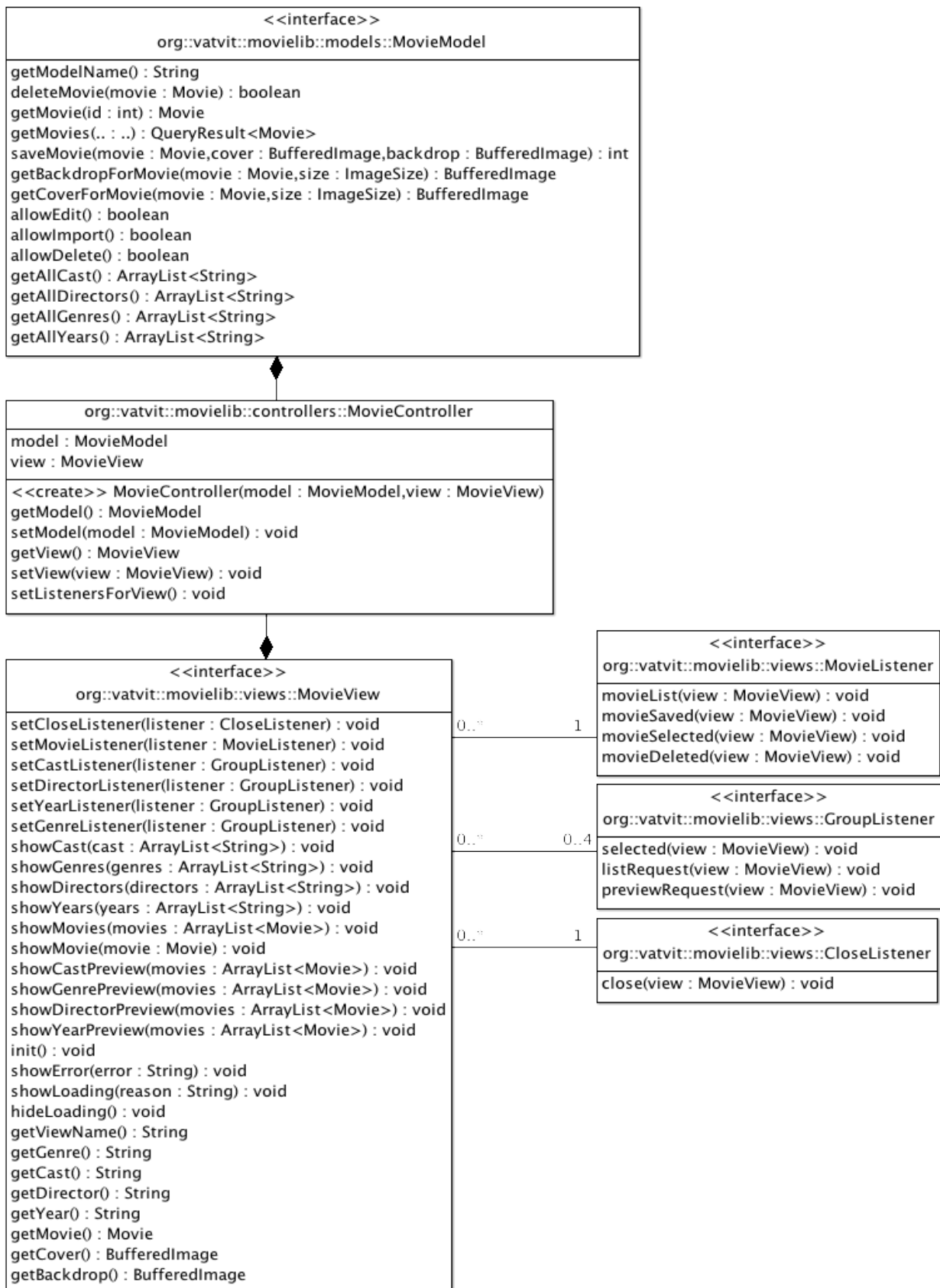
Tässä kappaleessa käsitellään sovelluksen arkkitehtuuria.

Sovelluksen arkkitehtuurimalliksi on valittu MVC. Malli valittiin sen tuoman muokattavuuden ja modulaarisuuden takia. Modulaarisuuden maksimoimiseksi MVC:n toteuttamiseksi valittiin niin sanottu välittäjästrategia (Mediator Strategy), jossa mallin ja näkymän toiminta ei ole sidottu toisiinsa tai sovelluksen muuhun toimintaan vaan tiedonvälitys tapahtuu ohjaimen asettamien

kuuntelijoiden kautta.



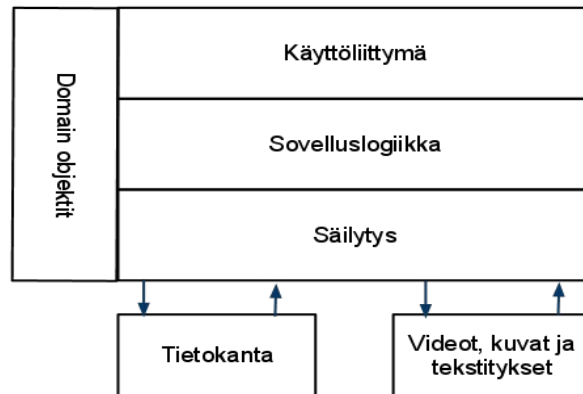
Kuva 19: MVC -arkkitehtuuri



Kuva 20: Luokkakaavio ohjaimen (*MovieController*) ja mallin (*MovieModel*) ja näkymän (*MovieView*) välisistä riippuvuuksista.

Kerrokset

Arkkitehtuurissa on pyritty kerrosmaisuuuteen. Kerrokset hyödyntävät aina alemman kerroksen tarjoamia palveluita. Kerroksisuus parantaa sovelluksen muokattavuutta ja selkeyttää vaihdettavien moduulien rakennetta. Kerroksine rakenne estää myös ylimääräisten riippuvuuksien syntymisen kerrosten välille.



Kuva 21: Arkkitehtuurin kerroskuvaus

Käyttöliittymäkerros

Käyttöliittymäkerros sisältää sovelluksen käyttöliittymät. Käyttöliittymissä tapahtuvat tapahtumat välitetään sovelluslogiikka kerrokselle sovelluslogiikkakerroksen asettamien kuuntelijoiden avulla. Käyttöliittymäkerros koostuu **MovieView -rajapinnasta** ja sen toteuttavista käyttöliittymistä **MovieViewImpl** ja **MovieEditorViewImpl**.

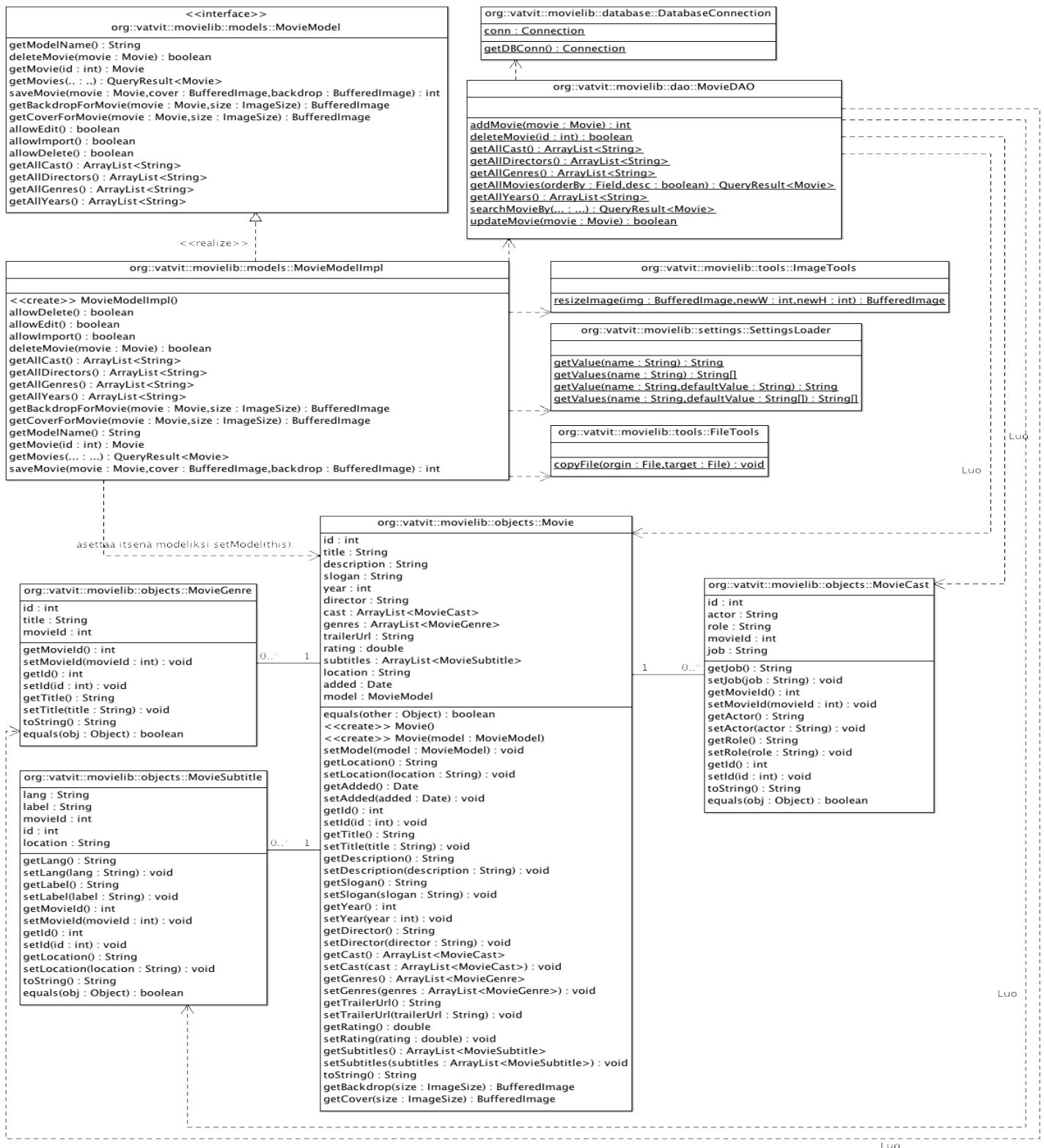
Sovelluslogiikkakerros

Sovelluslogiikkakerros sisältää sovelluksen käyttöliittymän ja säilytyskerroksen yhdistävät toiminnot. Yleisestä käytännöstä poiketen monet sovelluslogiikkakerroksen toiminnot on siirretty säilytyskerroksen toteutettavaksi. Tähän on päädytty, jotta sovelluksen riippuvuus käytetystä tietolähteestä saadaan rajattua säilytyskerrokseen. Sovelluslogiikkakerros koostuu **MovieController -luokasta**.

Säilytyskerros

Säilytyskerros sisältää sovelluksen tietovarastojen hallintaan tarkoitetun toiminnallisuuden. Koska sovelluksen toiminta painottuu suurelta osin sovelluksen tietovarastojen selaamiseen ja hallitsemiseen, toteutetaan säilytyskerroksessa myös monet normaalisti sovelluslogiikkakerroksessa toteutettavat toimenpiteet liittyen elokuvien lisäyksen ja poiston yhteydessä tapahtuviin toimenpiteisiin.

Elokuvan perustiedot, näyttelijät, kategoriat ja tekstityksien tiedot säilötään SQLite -tietokantaan ja media-tiedostot (videot, kannet, taustat ja tekstit) säilötään, joko sovelluksen data-kansioon tai asetuksissa määritettyyn muuhun kansioon. Säilytyskerros koostuu **MovieModel -rajapinnasta** ja sen toteuttavasta **MovieModelImpl**:sta.



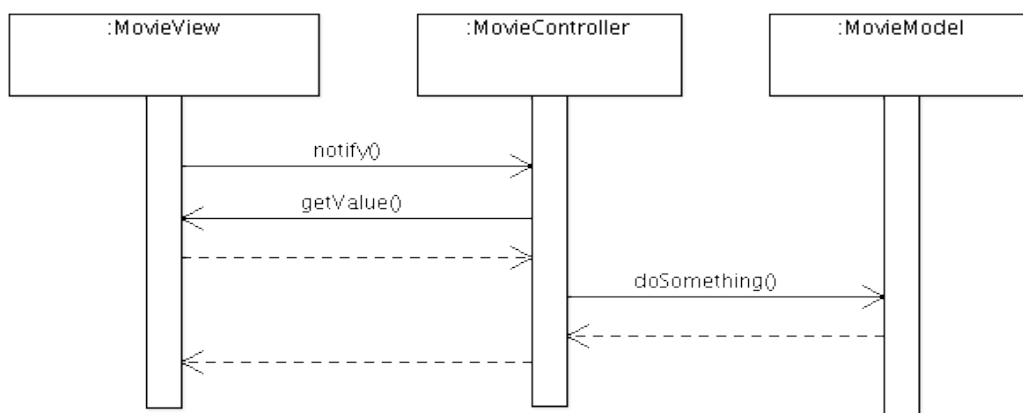
Kuva 22: Mallin rajapinnan implementaation luokkakaavio. Luokkakaaviossa kuvattu vain julkiset metodit ja `getMovie()` -metodi on lyhennetty parametrien suuren lukumäärän takia.

Esimerkit

Tässä osiossa käsitellään sovelluksen toimintaan yleispätevän sekvenssikaavio avulla.

Sovelluksessa käytettävä MVC -arkkitehtuurimalli tekee kaikista sovelluksen toimenpiteistä rakenteeltaan samankaltaisia. Tästä syystä sovelluksen toiminnan kuvaamiseksi riittää yksi sekvenssikaavio.

Sovelluksen käyttöliittymässä (**MovieView**) tapahtuneesta tapahtumasta ilmoitetaan ohjaimelle (**MovieController**) ohjaimen asettaman tapahtumakohtaisen kuuntelijan kautta. Tämän jälkeen ohjain hakee käyttöliittymältä tarvitsemansa tiedot. Saatuja tietoja käyttäen ohjain keskustele mallin (**MovieModel**) kanssa. Mallin palauttamien tietojen perusteella ohjain muuttaa käyttöliittymää vastaamaan mallin tilaa.



Kuva 23: Sekvenssikaavio sovelluksen toiminnasta

Käyttöliittymä

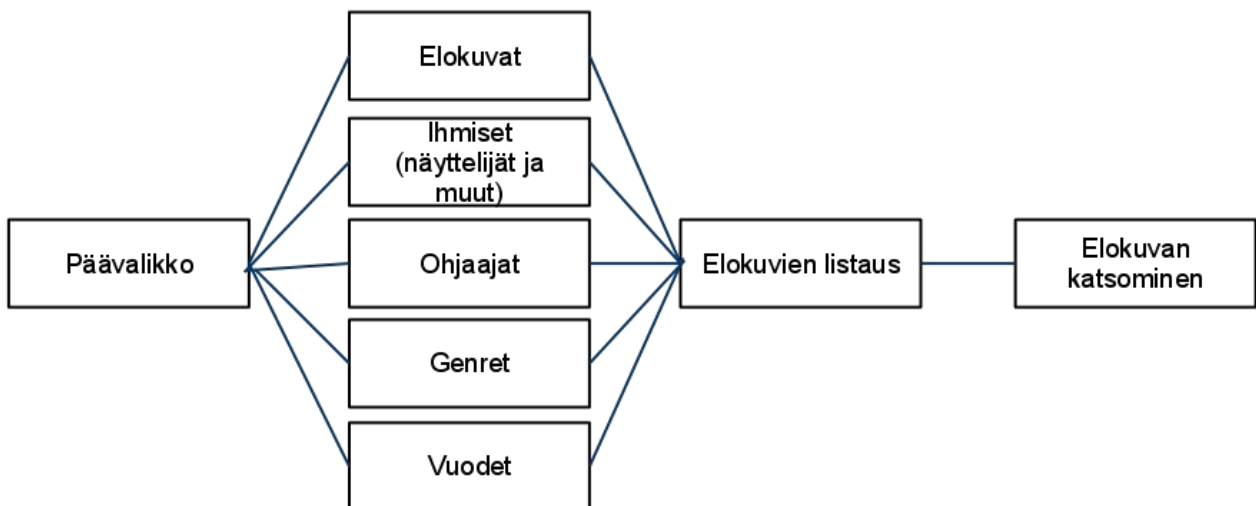
Tässä kappaleessa käsitellään sovelluksen käyttöliittymiä.

Sovelluksen käyttöliittymänä voi toimia mikä tahansa **MovieView -rajapinnan** toteuttava luokka. Sovelluksen määrittelyn mukaisesti sovellukselle on toteutettu yksi elokuvien katsomista varten tarkoitettu käyttöliittymä (**MovieViewImpl**) ja yksi elokuvien muokkaamista ja lisäämistä varten tarkoitettu käyttöliittymä (**MovieEditorViewImpl**).

MovieViewImpl

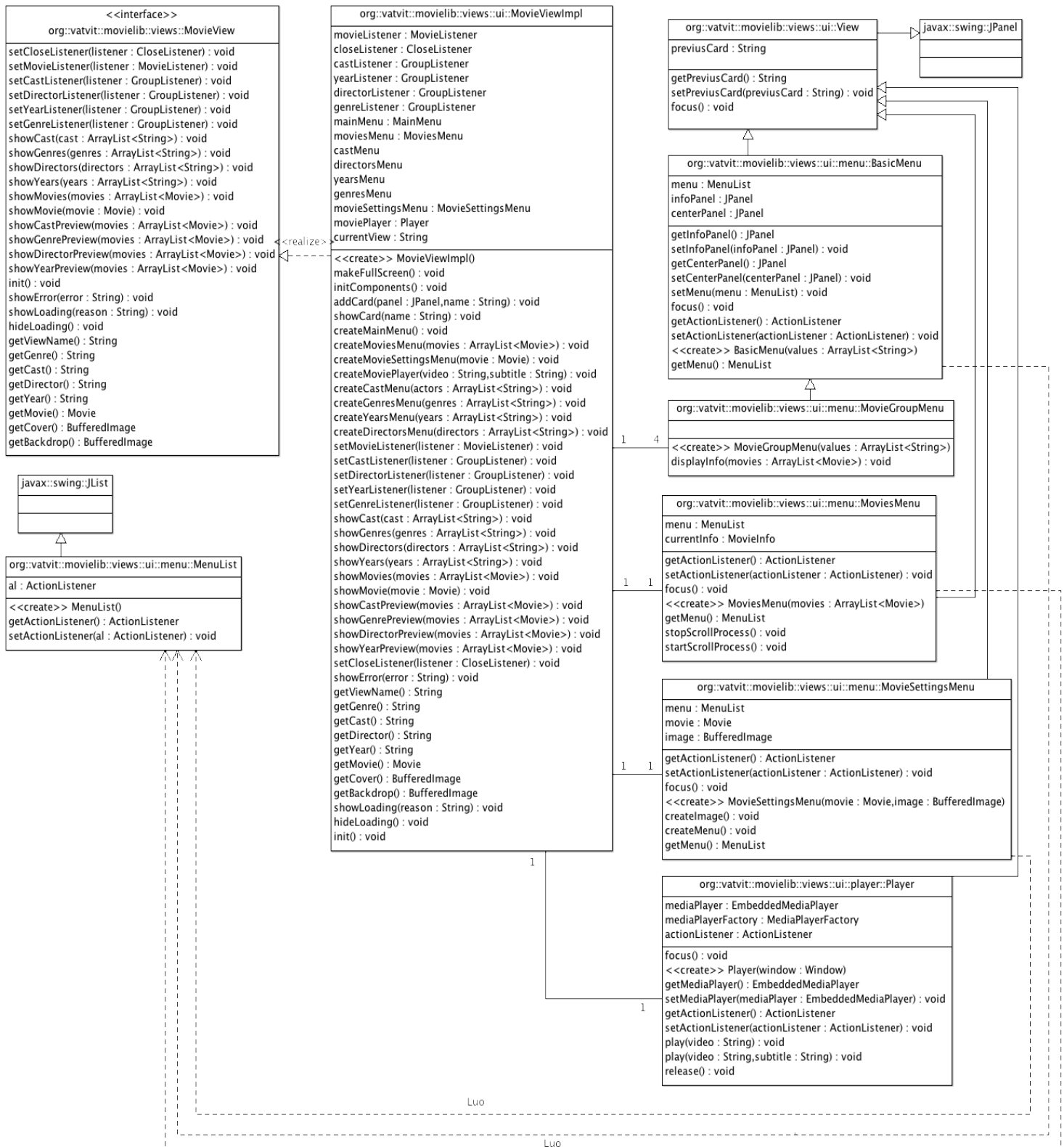
MovieViewImpl on kokoruutuun aukeva elokuvien selaamista ja katsomista varten tarkoitettu käyttöliittymä. **MovieViewImpl** laajentaa Swingin Jframea. Kokoruutu tilassa hyödynnetään Javan uusia kokoruutu -toimintoja ja elokuvien katsomiseen käytetään VLC -kirjasto.

Käyttöliittymän valikoissa liikkuminen tapahtuu nuolinäppäimillä. Valikon kohdan voi valita painamalla oikealle tai ENTER -painiketta. Takaisin palaaminen tapahtuu painamalla vasemmalle tai ESC -painiketta. Valikoissa voi myös liikkua käyttäen hiirtä.



Kuva 24: *MovieViewImpl* -valikkorakenne

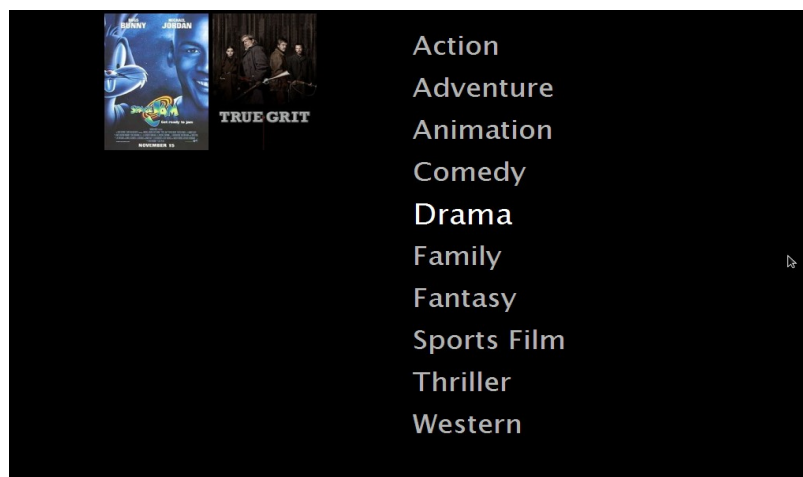
Valikkonäkymien pinoamisessa hyödynnetään **CardLayout**:ia, jonka avulla **Jpanelit** on kasattu päällekkäin niin, että vain haluttu valikko tai näkymä näytetään. Näkymät on jaettu kahteen osaan: lista/menu- ja info -puolleen. Jaossa käytetään **BoxLayout**:ia. Valikoiden keskittämiseen käytetään **GridBagLayout**:ia. Valikko toteutetaan laajentamalla **Jlistiä** ominaisuuksilla, jotka mahdollistavat sen käyttämisen näppäimistön kautta tapahtuvassa navigoinnissa.



Kuva 25: `MovieViewImpl`-käyttöliittymän luokkakaavio. Luokkakaaviossa kuvattu vain käyttöliittymää varten tehdyt komponentit. Swing -komponenttien (`Jpanelin`, `JscrollPane` ja `layoutit`) käyttö jätetty pois kaaviosta.



Kuva 26: Käyttöliittymän päävalikko. Vasemmalle puolelle ladataan valinnan nimeä vastaava kuva.



Kuva 27: Genre-valikko. Vasemmalla puolella näytetään kooste genreen kuuluvien elokuvien kansista. Muiden ryhmien (ohjaajat, näyttelijät ja vuodet) näkymissä käytetään samaa ulkoasua/valikkoa.



Kuva 28: Elokuva-listaus. Oikealla lista ryhmään kuuluvista elokuvista ja vasemmalla elokuvan tiedot. Elokuvan tiedot on asetettu **JscrollPane:n** sisälle. JscrollPane:sta tehdään automaattisesti ylöspäin vierivä omassa **threadissa** suoritettavan prosessin avulla. Tämä mahdollistaa suurienkin tietomäärien näyttämisen näkymässä.



Kuva 29: Elokuvan katsomisasetukset. Valikosta valitaan halutaanko elokuvasta katsoa esittelyvideo eli trailer tai millä tekstityksillä elokuva halutaan katsoa.



Kuva 30: Esittelyvideon katsominen. Näkymällä ei ole eroa elokuvien normaaliin katsomiseen. Videon kelaaminen tapahtuu nuolinäppäimillä ja toistotilasta voi poistua ESC -näppäimellä.

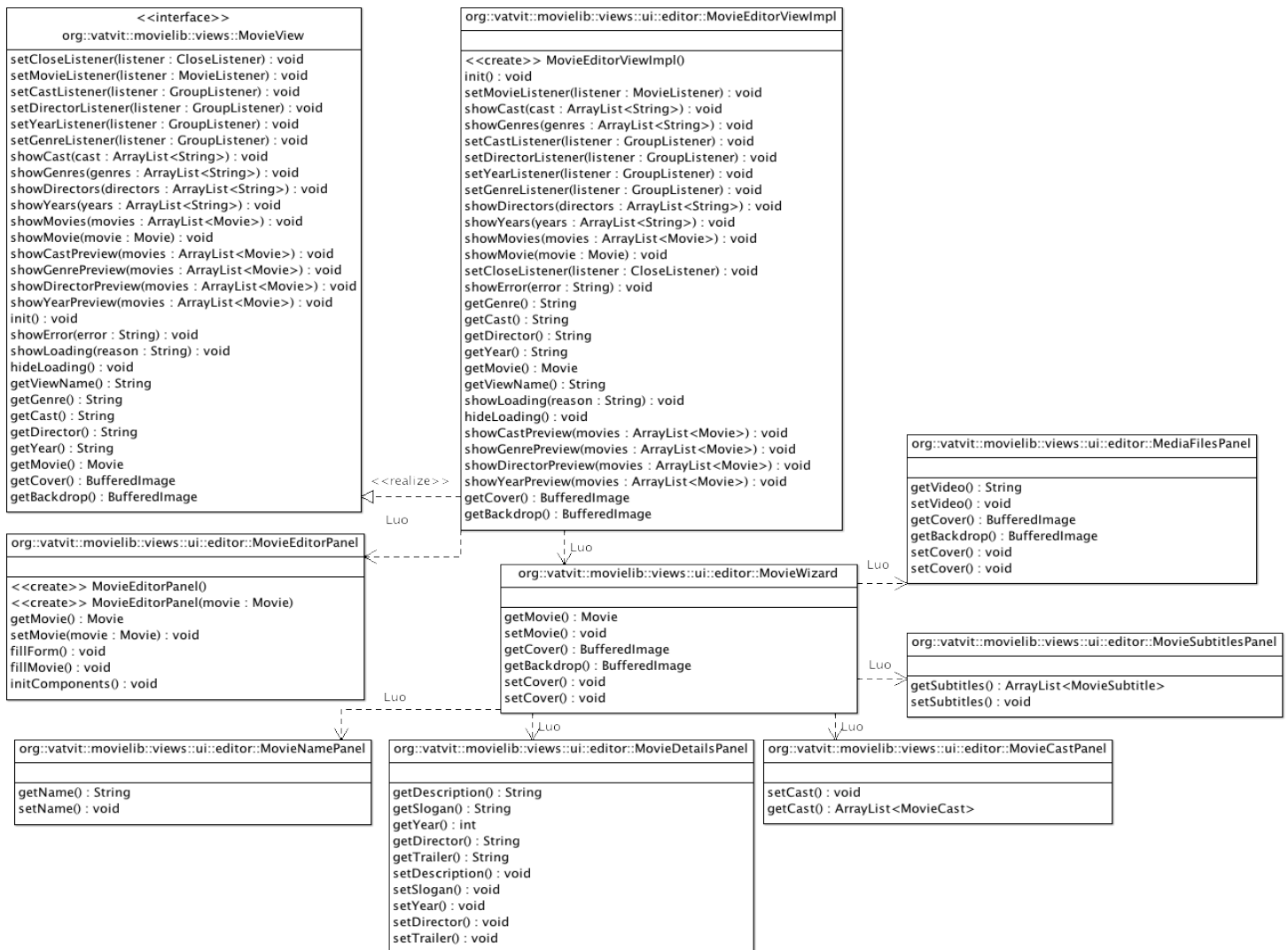
MovieEditorViewImpl

MovieEditorViewImpl on elokuvien lisäämiseen ja muokkaamiseen tarkoitettu käyttöliittymä. Käyttöliittymä koostuu elokuvien listauksesta, valitun elokuvan tietoja esittävästä paneelista ja wizard -tyylisestä uuden elokuvan lisäykseen ja jo tietokannassa olevan elokuvan muokkaamiseen tarkoitettu näkymästä. Elokuvien listaus toteutetaan **Jlist**:llä. Lomake toteutetaan käytten kahta rinnakkaista **FlowLayout**ia ja uusien elokuvien lisäys näkymässä käytetään **CardLayout**ia.

Elokuvien lisäys tapahtuu vaiheissa alkaen elokuvan nimen täyttämällä ja mahdollisilla esitietojen haulla **TMDb** -palvelusta. Koska esitiedoista voi tulla monia eri vastauksia on käyttäjän valittava saaduista vastauksista oikea elokuva. Tämän jälkeen käyttäjä siirtyy täyttämään tai tarkistamaan muut elokuvan perustiedot (slogan, tuotantovuosi, ohjaaja, kuvaus, trailer ja arvostelut). Seuraavaksi käyttäjä tarkistaa listauksen elokuvan näyttelijöistä ja muusta tuotantoryhmästä. Seuraava näkymässä asetetaan tai tarkitetaan elokuvan kansikuva ja taustakuva. Seuraavassa näkymässä asetetaan käyttäjän haluamat tekstitykset elokuvalle. Viimeisessä vaiheessa käyttäjä asettaa videotiedoston.

Uusi	
Elokuvalista	Tietolomake

Kuva 31: Päänäkymän asettelu



Kuva 32: MovieEditorViewImpl -käyttöliittymän luokkakaavio. Luokkakaaviossa kuvattu vain käyttöliittymää varten tehdyt komponentit.

Jatkokehitys

Tässä osiossa käsitellään sovelluksen jatkokehitykseen liittyviä ohjeita ja ideoita. Sovellus on suunniteltu rakenteeltaan mahdollisimman modulaariseksi ja toimimaan niin että sovelluksen malli tai näkymä voidaan vaihtaa muuttamatta itse sovelluksen koodia tai joutumatta kääntämään sovellusta uudelleen.

Ohjeita

Jatkokehittämistä ja koodin kääntämistä helpottaa suuresti Apache Ant -työkalun käyttö. Sovelluksen tiedostot sisältävät build.xml -tiedoston, jonka avulla Apache Ant -osaa kääntää koodin käyttäen lib -kansiossa olevia kirjastoja.

Sovellus koostuu kolmesta osasta; mallista, näkymästä ja ohjaimesta. Näistä malli ja näkymä toteutetaan rajapintojen kautta ja ladataan käynnistyksen yhteydessä sovelluksen Main -luokkaan rakennetun luokkalataimen avulla. Tästä syystä uusien tietolähteiden tai käyttöliittymien

toteuttaminen on helppoa ja suoraviivaista. Esimerkiksi uusi tietolähde lisätään luomalla luokka joka toteuttaa MovieModel -rajapinnan ja lisäämällä uuden luokan tiedot movielib.conf -tiedostoon tai vaihtoehtoisesti ohjaamalla sovellus käyttämään mallia käynnistyksen yhteydessä -m argumentin avulla (esim. java -jar movielib.jar -m org.vatvit.movielib.model.MovieModelImpl).

Malli on myös ainoa Movie -objektien lähde. Tästä syystä sovellukselle toteutettavan mallin yhteydessä myös Movie:ta voidaan laajentaa vastaamaan paremmin uuden mallin tai mahdollisesti uuden käyttöliittymänkin toimintaa.

Ideoita

- Tuki ja näkymät sarjoja varten. Vaatii muutoksia koko sovellukseen.
- Keskitetty videovarasto = videoiden katsominen ja selaaminen suoraan palvelimelta.
- Käyttöliittymä Androidille. Vaatii Androidin omien multimedia ominaisuuksien hyödyntämisen libvlc:n sijaan.

Testaus

Tässä osiossa käsitellään sovelluksen toiminnan testausta. Tarkastelun alla ovat testimenetelmät, tulokset sekä tuloksien uudelleen tuottaminen.

Yksikkötestaus

Muu testaus