

Olio-ohjelmoinnin perusteet Harjoitustyö

ELOKUVATEATTERIN VARAUSJÄRJESTELMÄ

Teemu (522229, tpsalo@utu.fi), Vili (522096, vkstah@utu.fi)
Walteri (520683, wmmynt@utu.fi), Nevil (522687, npsand@utu.fi)

1. Tehtävän kuvaus

Tehtävänä oli tehdä varausjärjestelmä, jossa pystyy varaamaan elokuvalipun eri elokuvaan. Elokuvissa on myös ikärajat, joten oma ikä tulee syöttää varausta tehdessä. Varauksia voi tehdä useampia ja paikkoja voi varata monta kappaletta. Varauksen voi myös poistaa ja järjestelmästä voi tarkastaa, mitä varauksia on tehnyt.

2. Vastuualueet

Walteri - Testausjärjestelyvastaava, 520683 wmmynt@utu.fi

Tehtävänä oli ohjelman testaus eri vaiheissa, löydettyjen ongelmien korjaaminen sekä ohjelman yleisten ominaisuuksien kehittäminen.

Nevil - Käyttöliittymän, luokkien ja vaadittujen ominaisuuksien kehittäminen, 522687 npsand@utu.fi

Tehtävänä oli tehdä ja ohjelmoida suunnitelmassa olevia luokkia ja metodeja. Myös tekstipohjaisen käyttöliittymän tekeminen oli osa tehtävää.

Vili - SQLITE-tietokantavastaava, 522096 vkstah@utu.fi

Tehtävänä oli yhdistää ohjelma ja tietokanta niin, että elokuvaa varatessa tietokannassa olevaan tauluun lisätään uusi varaus.

Teemu - Luokkien ja vaadittujen ominaisuuksien kehittäminen sekä testaus, 522229 tpsalo@utu.fi

Tehtävänä oli luoda luokat ja hyödyntää polymorfismia sekä tehdä ominaisuudet sekä testata niiden toimivuus. Lisäksi lopullinen kaikkien ominaisuuksien ja koko järjestelmän toimivuuden testaus ja bugien korjaus.

3. Ratkaisuperiaate

Pyrimme tekemään ohjelmasta mahdollisimman simppelellä näköisellä, jotta ohjelman funktionaalisuus korostuisi paremmin. Ohjelman on määrä lähennellä oikeaa varausjärjestelmää, joten päätimme lisätä tietojen säilyttämiseksi yksinkertaisen SQLITE-tietokannan. Tietokannassa on kolme taulua: varaukset, elokuvat ja salit. SQLITE antaa mahdollisuuden ohjelman tilan tallentamiseksi.

4. Ohjelman osat

Asiakas.java

Asiakas-luokka sisältää kaksi havainnointimetodia, joilla palautetaan luokan konstruktorissa asetetut arvot nimi sekä ikä.

Elokuva.java

Elokuva-luokka on samanlainen, mutta asetetut arvot ovat elokuvan nimi sekä sen ikäraja.

Jarjestelma.java

Jarjestelma-luokka ottaa ja laittaa tietoja tietokantaan, sekä tekee muutamia tulostuksia, joita näytetään käyttöliittymässä.

Sali.java

Sali-luokka on abstrakti luokka, joka sisältää konstruktorin ja muutaman epäabstraktin metodin.

Sali2D.java ja Sali3D.java

Sali2D ja Sali3D ovat luokkia, jotka perivät luokan Sali. Luokissa ei ole sinällään mitään eroa, mutta jos asiakas tekee varauksen 3D-saliin paikkoja, menee tietokannasta 2 paikkaa/henkilö, kun taas varaus 2D-saliin vie vain yhden paikan.

Varaus.java

Varaus-luokka sisältää varauksessa tarvittavia tietoja, kuten varauksen tehneen asiakkaan tiedot, varattavan elokuvan nimi ja varattavien paikkojen lukumäärä. Luokka sisältää kaikista attribuuteista havainnointimetodit.

Tietokanta.java

Tietokanta-luokka ottaa yhteyden SQLite-tietokantaan, josta haetaan tai jonne asetetaan tietoa. Luokka sisältää useita metodeja, joilla on palautusarvo, jotta tietokannasta saatujen tietojen avulla voidaan muodostaa mm. Olioita. Luokasta tuli hieman sekava, koska aikaisempaa kokemusta SQLiten käytöstä ei ollut. Luokka sisälsi ehkä hieman toisteisia metodeja.

Kayttoliittyma.java

Käyttöliittymä-luokka sisältää kaksi metodia, jotka vastaavat tekstipohjaisen käyttöliittymän toimivuudesta. Molemmat metodit tulostavat asioita. `valikko()` vastaa käyttöliittymän ulkoasusta ja suurimmasta osasta tulostuksista.

Main.java

Main-luokka sisältää ainoastaan main-metodin, jossa luodaan tarvittavat oliot ja käynnistetään käyttöliittymä.

5. Testausjärjestely

Testasimme ohjelmaa eri päätelaitteilla, esimerkiksi eri kannettavilla ja pöytäkoneilla. Projektitiimin sisäisen testailun lisäksi pyysimme opiskelijatovereitamme kokeilemaan ohjelmaa. Näin löysimme ohjelmasta sellaisia ongelmia, jotka eivät sinänsä aiheuta erroria, mutta ovat käyttäjän kannalta huonoja ratkaisuja. Ohjelma ajettiin myös erilaisilla ohjelmilla, esimerkiksi Eclipsellä ja IntelliJ IDEAlla.

6. Liitteet

Harjoitustyön kuvaus: <https://moodle.utu.fi/mod/resource/view.php?id=866550>

Käyttöohje: ks. "Ohjeet ohjelman ajamiseksi.txt" (löytyy GitHubista)