

Aihe: B Varausjärjestelmä

Teemme harjoitustyöksemme huonevarausjärjestelmän hotellille.

Ratkaisuperiaate:

Implementoidaan ainakin viisi luokkaa, jotka ovat luokka graafiselle käyttöliittymälle, kokonaista hotellia mallintava luokka, yhtä huonetta mallintava luokka, luokka mallintamaan varausta ja luokka tietokannan manipulaatiota varten. Lisäksi luodaan kaksi Huone-luokan perivää, joista toinen mallintaa perushuonetta ja toinen luksuhuonetta, sekä muutamia virheluokkia. Implementoidaan myös mahdollisesti graafista käyttöliittymää varten luokkia.

Käyttöliittymä-luokka siis mallintaa ohjelman graafista käyttöliittymää. Luokkaan luodaan ainakin metodi `kaynnistaKayttoliittyma()`, joka muodostaa käyttöliittymän ja asettaa sen näkyväksi.

Käyttöliittymä-luokan avuksi tehdään todennäköisesti luokkia esim. käyttöliittymän napeille, tekstiruuduille, objektilaatikoille yms., jotta koodi olisi selvempää. Nämä luokat perivät Javan Swing-kirjaston luokkia.

Hotelli-luokkaan implementoidaan metodit `onkoTaynna(Date aloitus_paiva, Date lopetus_paiva)`, jonka avulla voidaan tutkia, onko hotelli varattu jonain tiettynä päivänä täyteen, ja `haeVapaat(Date aloitus_paiva, Date lopetus_paiva)`, jonka avulla voidaan etsiä vapaat huoneet. Hotelli-luokalla tulee olemaan myös attribuutti `huone_maara`.

Huone-luokalla tulee olemaan `huone_id` attribuutti. Sen aliluokilla `Perushuoneella` ja `Luksuhuoneella` tulee olemaan attribuutteja liittyen huoneen ominaisuuksiin.

Varaus-luokan attribuuteilla pystytään yksilöllistämään varaukset ja myöskin tutkimaan, onko jokin huone varattu tiettynä aikana. Varaus-luokkaan implementoidaan metodit `muutaVarausta(int varaus_id, Date aloitus_paiva, Date lopetus_paiva)`, jonka avulla voidaan muuttaa varauksen tietoa, `poistaTietokannasta()`, joka poistaa varauksen tiedot tietokannasta, sekä `tiedotTietokantaan()`, joka vie varauksen tiedot tietokantaan.

Tietokanta-luokan avulla hallitaan tietokannassa olevia tietoja. Sillä tulee olemaan metodi, jolla yhdistetään SQLite:n JDBC Driverin avulla. Lisäksi luodaan metodit tietokannan olemassa olon tarkistamista varten sekä tietokantatiedoston luomista varten. Kaikki huoneen varaukset voidaan hakea `haeVaraukset(intt huone_id)`-metodilla ja jokin tietty varaus `haeVaraus(int varaus_id)`-metodilla. Itse tietokanta tule olemaan muotoa (Integer unique primary key `varaus_nro`, Integer `huone_id`, String `varaaja`, String `varaus_alku`, String `varaus_loppu`, boolean `onko_luksus`).

Ohjelmaa varten implementoidaan myös virheluokat `EiKyseistäHuonettaPoikkeus`, `EiKyseistäVaraustaPoikkeus` ja `EiTietokantaaPoikkeus`.

UML-kaavio

