

## Ohjelmistojen mallintaminen, syksy 2011, laskuharjoitus 5

**huom:** laskarit pidetään salissa B221

Kaikki alla olevat tehtävät tehdään etukäteen. Paikanpäällätehtävät tehtävät löytyvät laskarisivulta <https://wiki.helsinki.fi/pages/viewpage.action?pageId=76261353>

1. Terraariossa on monia erilaisia eliöitä. Eliöt ovat joko kasveja tai liikkuvia eliöitä. Kukin eliö kuuluu pysyvästi johonkin lajiin. Osa lajeista on toistensa vihollisia. Yhdellä lajilla voi olla useita eri vihollislajeja. Liikkuvat eliöt muodostavat laumoja. Laumaan kuuluu useita eliöitä, mutta eliö voi kerrallaan kuulua vain yhteen laumaan. Liikkuvat eliöt toteuttavat liikkumisominaisuuden. Liikkumisominaisuus säätelee eliön liikkumista terraariossa. Myös laumat toteuttavat oman liikkumisominaisuutensa joka vaikuttaa jokaisen lauman sisältämän eliön liikkeeseen. Mallinna terraario luokkakaaviona.
2. Osoitteesta <https://wiki.helsinki.fi/pages/viewpage.action?pageId=78616773> löytyy kuvaus sekä ohjelmakoodi aikoinaan Ohjelmoinnin jatkokurssilla käsitellyyn Maidon elämää -tehtä- väsarjaan.  
Takaisinmallinna koodi luokkakaaviona.
3. Kuvaa sekvenssikaaviona mitä Maidon elämässä tapahtuu kun seuraava koodi suoritetaan:

```
Maatila maatila = new Maatila("Esko", new Navetta(new Maitosailio()));
// voit ajatella edellisen sekvenssikaaviossa seuraavanlaiseksi:
// Maitosailio maitosailio = new Maitosailio();
// Navetta navetta = new Navetta(maitosailio);
// Maatila maatila = new Maatila( navetta );

Lypsyrobotti robo = new Lypsyrobotti();
maatila.asennaNavettaanLypsyrobootti(robo);

Lehma lehma1 = new Lehma();
Lehma lehma2 = new Lehma();

maatila.lisaaLehma(lehma1);
maatila.lisaaLehma(lehma2);

maatila.eleleTunti();

maatila.hoidaLehmat();

System.out.println(maatila);
```

4. Ohjelmoinnin jatkokurssin viikon 4 neljännessä tehtävässä toteutettiin luokat CDLevy, Kirja ja Laatikko. Viimeisen kohdan jälkeen kaikki näistä toteuttivat rajapinnan Talletettava. Oletetaan että luokkia käyttää seuraava main-metodi:

```
public class Main {
    public static void main(String[] args) {
        Laatikko laatikko = new Laatikko(10);

        laatikko.lisaa( new Kirja("Kent Beck", "Test Driven Development", 0.5) );
        laatikko.lisaa( new CDLevy("Pink Floyd", "Dark Side of the Moon", 1973) );
    }
}
```

```

        Laatikko isoLaatikko = new Laatikko(20);

        isoLaatikko.lisaa(laatikko);

        System.out.println( isoLaatikko );
    }
}

```

Kuvaa sekvenssikaaviona mitä tapahtuu kun main suoritetaan. Piirrä myös koodia kuvaava luokkakaavio.

- tehtävänanto osoitteesta:  
<http://www.cs.helsinki.fi/u/wikla/ohjelmointi/jatko/s2011/harjoitukset/4/>
- Java-koodi (esimerkkiratkaisu, ilmestyy 28.11. klo 23.59):  
<https://wiki.helsinki.fi/display/s2011paja/ohja+viikko+4>

5. Laajennetaan toissaviikolla aloitettua monopolin luokkamallia. Ota pohjaksi oma toissaviikoinen ratkaisusi tai esimerkkivastaus.

Voit olettaa, että kyseessä on tietokonepelinä toteutettava monopoli.

Seuraavassa asioita, jotka pitäisi tulla mallissa esille.

Ruutuja on useampaa eri tyyppiä:

- aloitusruutu
- vankila
- sattuma ja yhteismaa
- asemat ja laitokset
- normaalit kadut (joihin liittyy nimi)

Monopolipelin täytyy tuntea sekä aloitusruudun että vankilan sijainti.

Jokaiseen ruutuun liittyy jokin toiminto.

Sattuma- ja yhteismaaruutuihin liittyy kortteja, joihin kuhunkin liittyy joku toiminto.

Toimintoja on useanlaisia. Ei ole vielä tarvetta tarkentaa toiminnon laatua.

Normaaleille kaduille voi rakentaa korkeintaan 4 taloa tai yhden hotellin. Kadun voi omistaa joku pelaajista. Pelaajilla on rahaa.

Tässä vaiheessa ei vielä liitetä peliin metodeja. Niiden aika tulee seuraavalla viikolla.

6. Ohjelmoinnin jatkokurssilla on tutustuttu sangen hyödylliseen luokkaan ArrayList. ArrayList-luokan olioiden voi ajatella olevan automaattisesti itseään kasvattavia vaihtelevanmittaisia taulukoita, joihin voidaan säilöä muita olioita. Lyhyesti ilmaistuna ArrayList-oliot ovat "olio-säiliöitä".

Etsi Java-API:sta (<http://download.oracle.com/javase/6/docs/api/>), ArrayList-luokka.

Tutkitaan ArrayListin ominaisuuksia. Tällä kertaa ei kiinnitetä huomiota ArrayListin metodeihin vaan sen sijaintiin Java-API:n luokkahierarkiassa. Eli mikä on ArrayList:in yläluokka, mitä sisarluokkia (eli ylluokan muita aliluokkia) sillä on, mikä on ArrayListin ylluokan ylluokka, mitä serkkuluokkia ArrayListilla on, jne ...

**Erityisesti on syytä selvittää mitä "sopimuksia" eli rajapintoja ArrayList mainitsee toteuttavansa? Mitä noissa rajapinnoissa luvataan?**

Piirrä tilanteesta luokkakaavio. Jos kaavio kasvaa liian isoksi, älä tunge kaikkea kuvaan. Metodiniimiä ei kannattane luokkakaavioon ainakaan montaa merkitä.

ArrayList ja muut oliosäiliöt liittyvät läheisesti kurssin Tietorakenteet aihepiiriin. Ammattimaisessa Java-ohjelmoinnissa tämä erilaisten "oliosäiliöiden" luokkaperhe on erittäin hyödyllinen. Kuten huomataan, on erilaisia säiliöluokkia todella suuri määrä. Joskus onkin haastavaa löytää parhaiten omiin tarpeisiin soveltuva luokka.

Ohjelmoinnin jatkokurssin viikon 2 tehtävässä 5.2 tutustuttiin hieman ArrayListin tapaan toimivaan HashSet-luokkaan. **Miten HashSet-mukaan sijoittuu Javan luokkahierarkiaan ArrayList:in suhteen? Mikä selittää sen, että HashSet:iä ja ArrayList:iä voi käyttää monin paikoin toistensa tilalla? Mitä eroavaisuuksia löytyy?**