

## Ohjelmistojen mallintaminen, syksy 2010, laskuharjoitus 1

Tämänkertaisten tehtävien tarkoitus on toimia kertauksena muutamiin *Ohjelmoinnin perusteet* -kurssin aiheisiin. Näihin tehtäviin ei ole olemassa mitään absoluuttisia oikeita vastauksia. **Pääasia on, että yrität tosissasi ja teet jotain.** Kurssin luentomateriaali ei auta näiden tehtävien tekemisessä millään tavalla ja tietoa löytyykin lähinnä Ohjelmoinnin perusteiden materiaalista.

Tarkoitus on, että teet tehtävät etukäteen ennen laskuharjoitustilaisuutta.

1. Mikä on olio? Mikä on luokka? Entä attribuutti (toiselta nimeltään oliomuuttuja) ja metodi? Mikä on olion ja luokan keskinäinen suhde? Bonus: mitä tarkoittaa javassa static?
2. Tee yksinkertainen Java-ohjelma, jossa on ainakin 2 eri luokkaa. Yksi luokista on sellainen, että sen oliot sisältävät jonkin muun luokan olioita (esim. kello-olio sisältää viisari-olioita, talo koostuu huoneista, auto koostuu moottorista ja renkaista, ...). Tee luokille joitain metodeita. Tee pieni pääohjelma joka demonstroi luokan käyttöä.
3. Piirrä kuva tai joukko kuvia jotka havainnollistavat mitä edellisessä tehtävässä tekemässäsi ohjelmassa tapahtuu.  
  
Huom: kuva saa olla ihan minkäläinen tahansa, tässä vaiheessa ei oleteta, että osaat piirtää minkään "virallisen" formaatin mukaisia kuvia. Oikeastaan tehtävän tarkoitus onkin saada kaikki opiskelijat piirtämään omanlaisiaan kuvia. Opimme kurssin kuluessa tapoja, miten ohjelmaa voidaan kuvata UML-kielellä. Nyt UML:stä ei tarvitse tietää yhtään mitään.
4. Seuraavalta sivulta alkaen löytyy pätkä Ohjelmoinnin perusteiden laskareista tutusta ohjelmasta. Piirrä kuva tai joukko kuvia jotka havainnollistavat mitä ohjelmassa tapahtuu.
5. Jos joutuisit toteuttamaan Tietojenkäsittelytieteen laitoksen kurssi-ilmoittautumisjärjestelmän, mitä luokkia ohjelmassasi olisi? Tee alustava lista mahdollisista luokista.

Kirjaa ylös kaikki mieleen tulevat luokkaehdokkaat. Älä ole huolissasi ylimääräisistä tai "tyhmistä" luokkaehdokkaista, ne voidaan tarpeen vaatiessa karsia pois. Tähänkään harjoitukseen ei ole olemassa oikeaa vastausta. Nyt ainoastaan kartoitetaan mitä opiskelijoille tulee mieleen, ennen kuin asiaa on oikeastaan edes opetettu yhtään.

```

public class Kioski {
    public Matkakortti ostaMatkakortti(String nimi) {
        Matkakortti uusiKortti = new Matkakortti(nimi);
        return uusiKortti;
    }

    public Matkakortti ostaMatkakortti(String nimi, int arvo) {
        Matkakortti uusiKortti = new Matkakortti(nimi);
        uusiKortti.kasvataArvoa(arvo);
        return uusiKortti;
    }
}

public class Matkakortti {
    private String omistaja;
    private double arvo;
    private int pvm;
    private int kk;

    public Matkakortti(String n){
        omistaja = n;  pvm = 0;  kk = 0;  arvo = 0;
    }

    public void kasvataArvoa(double a){ arvo += a; }

    public void vahennaArvoa(double a){ arvo -= a; }

    public double getArvo(){ return arvo; }

    public void uusiAika(int p, int k){
        kk = k;
        pvm = p;
    }
}

public class Lataajalaite {
    public void lataaArvoa(Matkakortti k, double a) {
        k.kasvataArvoa(a);
    }

    public void lataaAikaa(Matkakortti k, int pvm, int kk) {
        k.uusiAika(pvm, kk);
    }
}

```

```

public class Lukijalaite {
    private double RATIKKA = 1.5;
    private double HKL = 2.1;
    private double SEUTU = 3.5;

    public boolean ostaLippu(Matkakortti k, int tyyppi){
        double hinta = 0;
        if ( tyyppi == 0 ) hinta = RATIKKA;
        else if ( tyyppi ==1 ) hinta = HKL;
        else hinta = SEUTU;

        if ( k.getArvo()<hinta ) return false;
        k.vahennaArvoa(hinta);

        return true;
    }
}

public class Main {
    public static void main(String[] args) {
        Lataajalaite rautatietori = new Lataajalaite();
        Lukijalaite ratikka6 = new Lukijalaite();
        Lukijalaite buss244 = new Lukijalaite();

        Kioski lippuluukku = new Kioski();
        Matkakortti artonKortti = lippuluukku.ostaMatkakortti("Arto");

        rautatietori.lataaArvoa(artonKortti, 3);

        ratikka6.ostaLippu(artonKortti, 0);
        buss244.ostaLippu(artonKortti, 2);
    }
}

```