

## Ohjelmistojen mallintaminen, syksy 2010, laskuharjoitus 4

1. Toisissa laskareissa oli tehtävänä laatia käyttötapausmalli Ohjelmoinnin harjoitustyön tuntikirjanpitoa tukevaan järjestelmään.

Etsi luokkakandidaatit luennollakin esitettyä käsiteanalyysiä käyttäen, eli kerää tehtäväkuvauksessa esiintyvät substantiivit. Karsi sen jälkeen ylimääräiset kandidaatit, eli attribuutit, synonyymit ja asiaan kuulumattomat.

2. Tee alustava kohdealueen eli määrittelyvaiheen luokkakaavio tuntikirjanpitojärjestelmästä. Etene esim. samaan tapaan kuin luennon kampaamoesimerkissä.
3. Oletetaan, että HKL-tehtävään (ks. edellisen viikon laskarit) liittyy seuraava pääohjelma.

```
public class Main {
    public static void main(String[] args) {
        Paivamaara paiva = new Paivamaara(23,11);
        Lataajalaite rautatietori = new Lataajalaite( paiva );
        Lukijalaite ratikka6 = new Lukijalaite( paiva );
        Lukijalaite bussi244 = new Lukijalaite( paiva );

        Kioski lippuLuukku = new Kioski();

        Matkakortti artonKortti = lippuLuukku.ostaMatkakortti("Arto");
        Matkakortti heikinKortti = lippuLuukku.ostaMatkakortti("Heikki");

        rautatietori.lataaArvoa(arttonKortti, 3);

        boolean onnistuiko = ratikka6.ostaLippu(arttonKortti, 0);
        onnistuiko = bussi244.ostaLippu(arttonKortti, 2);

        onnistuiko = ratikka6.onkoVoimassa(heikinKortti);
    }
}
```

Kuvaa tapahtumat sekvenssikaaviona. Jos yhteen kaavioon uhkaa tulla liikaa tavaraa, jaa kaavio useampaan osaan.

4. Seuraavalla sivulla on annettu pieni huoneistojen lämpötilojen hallintaa simuloiva Java-ohjelma. Takaisinmallinnetaan ohjelmakoodi UML:ksi. Piirrä ohjelmaa kuvaava luokkakaavio sekä sen main-metodin suoritusta kuvaava sekvenssikaavio.
5. Palaamme toisista laskareista tutun LAL-järjestelmän pariin. Tehtävänä on etsiä LAL:in luokkakandidaatit samaan tapaan kuin tehtävässä 1. Eli kerää substantiivit ja karsi ylimääräiset.
6. Tehdään alustava kohdealueen luokkakaavio osalle LAL-järjestelmää. Keskitetään vain neljään ensimmäiseen tekstikappaleeseen (eli *Palkkaamme*

*palvelukseemme työntekijöitä ... ei tule enää mukaan*) ja otetaan luokka-kaavioon niiden joukosta löytyviä asioita.

Ota edellisen tehtävän luokkalistasi ja karsi niistä ne luokat, jotka kuuluvat nyt tarkasteltavan osan ulkopuolelle. Mieti jäljelle jääneiden luokkien yhteyksiä sekä attribuutteja.

Älä pyri täydelliseen ratkaisuun. Tärkeintä on, että ongelmakenttä rupeaa pikkuhiljaa jäsentymään.

```
public class Simulator {
    public static void main(String[] args) {
        Room r = new Room();
        Thermostat t = new Thermostat(r);
        t.stabilizeAt(20);
    }
}

class Room {
    private int temperature = 0;
    int getTemperature() { return temperature; }
    void addTemperature(int t) { temperature = temperature + t; }
}

class Thermostat {
    private Sensor sensor;
    private Heater heaterA;
    private Heater heaterB;
    Thermostat(Room r) {
        sensor = new Sensor(r);
        heaterA = new Heater(r);
        heaterB = new Heater(r);
    }
    void stabilizeAt(int degrees) {
        int measured = sensor.getReading();
        while ( measured < degrees ) {
            heaterA.heatTheRoom();
            heaterB.heatTheRoom();
            measured = sensor.getReading();
        }
    }
}

class Sensor {
    private Room room;
    Sensor(Room r) { room = r; }
    int getReading() { return room.getTemperature(); }
}

class Heater {
    private Room room;
    Heater(Room r) { room = r; }
    void heatTheRoom() { room.addTemperature(5); }
}
```