

Demo 2 / 20.9

Demotehtävän alussa tai muulla iso **M** ja linkki tarkoittaa että seuraava viittaus on luentomonisteeseen.

Tehtävät

V1. [Ville-tehtävät](#) 2.2, 2.3, 2.4, 6.1, 6.2, 6.3.

1. **M: 24. Lukujen esitys tietokoneessa:** Binäärilukujärjestelmässä on kaksi numeroa, nolla ja yksi. Tätä sanotaan bitiksi (binary digit). Vähiten merkitsevä eli viimeisimmäksi kirjoitettu bitti tarkoittaa lukumäärää yksi, toiseksi vähiten merkitsevä bitti lukumäärää kaksi, kolmanneksi vähiten merkitsevä lukumäärää neljä ja niin edelleen aina kakkosen potensseja jatkaen (8, 16, 32, 64, ...). Kokonaisen binääriluvun ilmoittama lukumäärä saadaan kun summataan nämä yksittäisten bittien ilmoittamat lukumäärät toisiinsa. Esimerkiksi binääriluku 1000101 olisi kymmenjärjestelmässä:

$$1*64 + 0*32 + 0*16 + 0*8 + 1*4 + 0*2 + 1*1 == 69$$

Muunna seuraavat binääriluvut ihmiselle tutumpaan 10-järjestelmään:

```
10010
1001110001
100000000000000000
1010101010101011
100000000000000001
100000000000000010
100000000000000011
```

Nykytietokoneissa on tyypillistä jakaa muisti kahdeksan bitin mittaisiin tallennuspaikkoihin eli tavuihin (byte). Mikä on suurin kokonaisluku, jonka voit tallentaa kahden tavun(eli 16 bitin) kokoiseen tilaan (oletetaan että negatiivisia lukuja ei tarvitse esittää)? Entäpä jos pitää voida esittää positiivisia ja negatiivisia lukuja: mikä silloin on toisaalta pienin ja toisaalta suurin luku, joka kahden tavun kokoiseen tilaan mahtuu?

2. **M: 3. Algoritmit:** Kirjoita suomenkielinen (ei Java) algoritmi eli toimintaohje, jolla kuvaat, miten puhelinluettelosta etsitään tietyn henkilön nimi. Entä millaisella algoritmilla puhelinluettelosta löytyy tietyn numeron omistaja? Kumpi algoritmeista on nopeampi ja miksi? (Lue: <http://users.jyu.fi/~vesal/kurssit/ohj2/moniste/html/m.htm> luku 4).
 3. **M: 3. Algoritmit:** Pitäydy tässä tehtävässä vielä irti Javasta ja muista ohjelmointikielestä. Kuvaile suomen kielellä tai korkeintaan "pseudokoodilla", miten jakaisit seuraavat tehtävät yhtä pykälää tarkemmiksi osatehtäviksi. Jaa sitten vielä kukin osatehtävä seuraavalle tasolle pienemmiksi osatehtäviksi. Käytä jotakin luonnollisia ilmauksia toistolle ja päätöstilanteille ("jos ... niin ... mutta muussa tapauksessa ..."):
- a) Tiskaaminen
 - b) Kahvin keittäminen
 - c) Ohjeet tämän demotehtävän palauttamiseksi

4. **M: 6. Aliohjelmat:** Täydennä alla oleva ohjelma `Portaat.java` toimimaan kuten kommentteissa on sanottu:

```
import fi.jyu.mit.graphics.EasyWindow;

/**
 * Ohjelma piirtää viisi porrasta alkaen origosta päättyen
 * pisteeseen (5,5)
 * @author // Täydennä
 * @version // Täydennä
 */
public class Portaat {

    /**
     * Aliohjelma piirtää ikkunaan yhden nousevan portaan
     * alkaen pisteestä (x,y) ja päättyen pisteeseen (x+1,y+1)
     * <pre>
     * |----- (x+1,y+1)
     * |
     * |
     * |
     * (x,y)
     * </pre>
     * @param window
     * @param x portaan alkupisteen x
     * @param y portaan alkupisteen y
     */
    public static void porras(EasyWindow window, double x, double y) {
        // Täydennä tähän
    }

    /**
     * @param args ei käytössä
     */
    public static void main(String[] args) {
        EasyWindow window = new EasyWindow();
        window.scale(0,0,10,10);
        porras(window,0,0);
        porras(window,1,1);
        // Täydennä tähän
        window.showWindow();
    }
}
```

5. **M: 6. Aliohjelmat:** Lisää edelliseen ohjelmaan aliohjelma `porrasAlas`, ja sen kutsut, niin, että voi hyvällä omalla tunnolla lisätä alkukomentteihin tekstin:

```
* Sitten piirretään laskevat portaat alkaen pisteestä (4,5)
* ja päättyen pisteeseen (9,0)
```

Syntyvä kuva on seuraavan näköinen:

```

      (5,5)
     /  \
    /    \
   /      \
  /        \
 /          \
/            \
(0,0)         (9,0)
```

6. **M: 6. Aliohjelmat:** Tee ohjelma, joka piirtää ikkunaan 4 kappaletta tasakylkisiä kolmioita, joiden korkeus on yksi yksikkö ja leveys kaksi yksikköä. Tee avuksi aliohjelma kolmio, jolle viet parametrina ”sopivat” tiedot.
7. **M: 6. Aliohjelmat:** Kommentoi ja muuta alla oleva UkkoNoa selaiseksi,

```
import fi.jyu.mit.Music.*;

public class UkkoNoa {

    public static void main(String[] args) {
        MidiPlayer mp = new MidiPlayer();

        mp.play("CCCEDDDFEEDD");
        mp.play("C", 0.5);

        mp.play("EEEE");
        mp.play("GF", 0.5);
        mp.play("DDDD");
        mp.play("FE", 0.5);

        mp.play("CCCEDDDFEEDD");
        mp.play("C", 0.5);

        mp.play(" ");
    }
}
```

että pääohjelma on

```
public static void main(String[] args) {
    MidiPlayer mp = new MidiPlayer();

    soitaUkkoNoanKertoSae(mp);
    soitaUkkoNoanSakeisto(mp);
    soitaUkkoNoanKertoSae(mp);

    mp.play("-");
}
```

- B1. Aloita avaamalla Alicen luisteluesimerkki. Eli aloita luistelu-tutorial ja poistu siitä saman tien. Taustalla on puita. Laita kaksi niistä saapumaan luistelijan luo niin, että ensimmäinen saapuu 2 sekunnin aikana ja ”kaatuu” 45 astetta vasemmalle toisen saapumisen jälkeen ja toinen kaatuu sitten 45 astetta oikealle. Tehtävän vastauksena palauta lisäämäsi ”ohjelmarivit” tekstiksi kopioituna tyyliin

```
IceSkater.do simple spin
```

GURU-tehtävät

G1-2 Tutustu <http://users.jyu.fi/~vesal/kurssit/ohj1/graphics/> kirjaston malliohjelmiin CarSample.java ja UkkoSample.java Tee tehtävät 4,5 ja 6 niin, että teetkin luokat Porras, PorrasAlas ja Kolmio, joiden avulla piirät ikkunaan portaat (nyt luonnollisesti for-silmukalla) ja 4 kolmiota. Tee yhdestä kolmiosta pyöriteltävä.