

# Demo 5 / 11.10

## Tehtävät

- V1. Tee [Ville](#)-tehtävät: 4.1, 4.2, 4.3, 4.4, 5.1, 5.4, 9.1
- T1. Jos tarkistat vähintään kahden funktion toiminnan automaattisella testillä ([ComTest](#) ja/tai [JUnit](#)), saat merkitä yhden lisäpisteen.
- M: [15. Toistorakenteet](#):** Ota edellisen kerran guru-tehtävän [LuvutGraafisesti.java](#) mallivastaus ja muuta sitä niin, että `piirraKuva` -aliohjelma piirtää taulukon suurimmat arvot punaisella ympyrällä, pienimmät vihreällä ympyrällä ja loput mustilla ympyröillä.
  - M: [15. Toistorakenteet](#):** Edellistä matkien tai edelliseen lisäten tee toinen (kuormitettu, overloaded) aliohjelma `piirraKuva`, jolle viedään taulukon lisäksi parametrina luku, jota pienemmät arvot piirretään vihreällä ja toinen luku, jota suuremmat arvot piirretään punaisella (yhtäsuuruus mukana). Voitko tehdä tehtävän 1 käyttäen tätä aliohjelmaa? Voiko tätä käyttäen piirtää myös samanlaisen kuvan kuin edellisen kerran Guru-tehtävässä?
  - M: [14. Taulukot](#):** Ota luennolla tehty taulukkomalli: [Taulukot.java](#) ja muuta se sellaiseksi, että siinä on aliohjelma `tulostaYli(taulukko, raja)`, joka tulostaa taulukosta kaikki ne luvut, jotka ylittävät annetun rajan (päättää itse ja hoida kommentoinnilla, että kuuluuko raja tulostettaviin vai ei). Tee vastaava funktio `summaYli(taulukko, raja)`, joka palauttaa rajan ylittävien taulukon lukujen summan.
  - M: [15. Toistorakenteet](#):** Tee aliohjelma `tulostaKertotaulu(n)`, joka tulostaa  $n:n$  kertotaulun `for` -silmukalla. Esimerkiksi `tulostaKertotaulu(3)` tulostaisi:

```
1 * 3 = 3
2 * 3 = 6
3 * 3 = 9
4 * 3 = 12
5 * 3 = 15
6 * 3 = 18
7 * 3 = 21
8 * 3 = 24
9 * 3 = 27
10 * 3 = 30
```

Tee vielä `tulostaKertotauluWhile(n, m)`, joka tulostaa  $n:n$  kertotaulusta  $m$ -riviä. Toteutus `while`-silmukalla. Esimerkiksi `tulostaKertotauluWhile(3, 5)` tulostaisi:

```
1 * 3 = 3
2 * 3 = 6
3 * 3 = 9
4 * 3 = 12
5 * 3 = 15
```

Tee sama vielä `do-while` -silmukalla. Mitä tulostakertotaulu `DoWhile(6, 0)`; tulostaa jos ei tehdä mitään erikoislisäyksiä? Mitä pitäisi lisätä?

5. **M: 14. Taulukot:** Tee funktioaliohjelma `summaa(luvut)`, joka laskee yhteen taulukossa olevat parilliset luvut ja vähentää niistä parittomat luvut. Funktio palauttaa tuon summan. Aloita kirjoittamalla pääohjelma, jossa mietit kuinka aliohjelmaa kutsutaan.
6. Tee funktio(aliohjelma) `alkaaIsolla(sana)`, joka palauttaa `true` jos parametrinä viety merkkijono alkaa isolla kirjaimella. Tee tätä käyttäen ohjelma, joka komentoriviltä käyttäen toimii seuraavasti (vinkki: `Character`-luokka, `args`-taulukko):

```
java demo5.AlkaaIsolla kissa Kana Koira kettu Mato 2Kanaa
Isolla alkaa:
1. argumentti = Kana
2. argumentti = Koira
4. argumentti = Mato
```

Eclipsessä F11 -ajoa varten argumentteja voi antaa seuraavasti:

```
Oletetaan että AlkaaIsolla on jo kerran Eclipsessä ajettu:
Run/Debug Configurations/AlkaaIsolla/Arguments
ja sitten kirjoitetaan halutut parametrit Program Arguments -kohtaan.
```

7. Tarkoitus olisi tehdä ohjelma, joka soittaa ”kahdella” kädellä. Oikealla kädellä soitetaan annetut nuotit kuten [SoitaKappale.java](#) demon vastauksessa. Vasemmalla kädellä soitetaan oktaavia alemmaa vain C-nuotit. ”Kahden käden” soittamisesta voit katsoa mallia Music-kirjaston [Wiki-sivulta](#). Tee funktio

```
public static String vasemmanKappale(String kappale)
```

joka palauttaa merkkijono jossa kaikki muut ISOT kirjaimet paitsi C on vaihdettu -merkeiksi (eli tauoiksi). Uusi ohj1.jar sisältää Music-kirjastosta version, jossa voi tauonkin perässä olla ylennys- tai alennusmerkki ilman että kappale ”hajoaa”. Eli esimerkiksi jono

```
ACCF# => -CC-# (ja -# on vain normaali tauko)
```

Vaihda siis `SoitaKappale`-esimerkissä `MidiPlayer`in tilalle `BufferedMidiPlayer2` ja soita jono `kappale` 1-kanavalla ja 2-kanavaa varten muunna `kappale` em. funktiolla ja soita näin syntynyt jono `kappale2`

```
mp.play(2, kappale2, 0.25, 3);
```

Halutessasi voit valita 2-kanavalle toisen soittimen.

- B1. **M: 14. Taulukot:** Tee funktioaliohjelma `moodi(luvut)`, joka etsii taulukon yleisimmän luvun. Jos eniten esiintyviä on yhtä monta, niin silloin ensimmäisenä vastaan tuleva tulkitaan yleisimmäksi. Mieti ensin tarkasti kuinka itse ratkaisit tämän ongelman (eli mieti algoritmi). Et tarvitse aputaulukkoja välttämättä, vaan tehtävän voi ratkaista muutamalla apumuuttujalla. Tämä on yksi keskiluvuista. Toinen on keskiarvo. Kirjoita tätäkin varten funktioaliohjelma. Kolmas keskiluvuista on mediaani, eli aineiston keskimmäinen alkio. Tätä ei vielä tarvitse tehdä. Mitä mediaanin laskemiseksi pitäisi ensin tehdä? Katso vaikka Wikipediasta keskiluku.

## GURU-tehtävät

G1-2 Tehtävässä 2 voi vielä joku haluta vaihtaa värejä. Parametristat kasvavat tällöin kohtuuttoman pitkeksi. Yksi hyvä ratkaisu on tehdä ympyröiden piirtäjästä olio. Tee tarvittava luokka niin, että alla oleva pääohjelma toimii:

```
public static void main(String[] args) {
    double luvut2[] = {1.9, 3, 2, 2.4, 1.2, 2.6, 3.2, 3.1};
    Window ikkuna = new Window(600, 400);
    ikkuna.scale(0, -1, luvut2.length, 4);
    ikkuna.add(new Axis(100, 100, 0));

    TaulukkoYmpyroiina t2 = new TaulukkoYmpyroiina(luvut2);
    ikkuna.add(t2);
    t2.setMinColor(Color.BLUE).setMaxColor(Color.ORANGE).
        setColor(Color.PINK).setR(0.4).setMin(2).setMax(3);

    ikkuna.showWindow();
}
```