

# Demo 11 / 24.11

## Tehtävät (mallitentti)

Tentissä ei tarvitse ihan tarkkaan ulkoa muistaa mitä Java:n API-kirjaston aliohjelmiä on käytettävissä. Jos tarvitset jotakin valmista luokkaa tai aliohjelmaa tai metodia, kuvaile mitä kuvittelet sen tekevän ja millaisilla parametreilla. Ohjelmatehtävät pitää dokumentoida JavaDoc:in mukaisesti. Tentissä keräile tehtäviä niin, että niiden maksimipistemäärä on 24 pistettä. Tentissä voi olla rajoitettu mitä tehtäviä ei saa ottaa. Tässä voisi vaatimuksena olla että ainkain toinen tehtävistä 1-2 tai 5 pitää olla valittujen joukossa.

Korpissa on kentät "Mallitenttien tehtävien tekemiseen kulunut aika". Täytä siihen tehtävittäin kauanko sinulta meni. Tämä on avuksi oikean tentin suunnittelua varten. Oikeassa tentissä aikaa on 4 tuntia.

- 1-2. **Tee ohjelma:** Pääohjelma kutsuu aliohjelmaa (funktioita), joka etsii ja palauttaa parametrina viedyn kokonaislukutaulukon 2. suurimman alkion. Aliohjelma EI SAA järjestää taulukkoa. Mitä huomionarvoisia erikoistapauksia voi olla? Määrittele miten ne käsitellään.

```
Esimerkiksi taulukko: 3 1 -7 9 15 8
Aliohjelma palauttaa arvon 9 (ei tulosta)
ja pääohjelma tulostaa:
Taulukon 3 1 -7 9 15 8 toiseksi suurin alkio on 9
```

Taulukon tulostaminen hoidetaan omalla aliohjelmalla joka pitää myös tehdä. Ohjelmassa saa käyttää vain pääohjelman lokaaleja muuttujia, aliohjelman parametrimuuttujia ja aliohjelmien lokaaleja muuttujia. (12p)

3. Piirrä kuva Lineja RPoint olioista ja niiden suhteista kun pääohjelmassa piirretään 3 ylös menevää porrasta.

```
package demo9;
import fi.jyu.mit.graphics.Line;
import fi.jyu.mit.graphics.RPoint;
import fi.jyu.mit.graphics.Window;

public class Portaat {

    public static RPoint porras(Window window, RPoint start) {
        RPoint mid = new RPoint(start.getX(), start.getY()+1);
        RPoint end = new RPoint(start.getX()+1, start.getY()+1);
        window.add(new Line(start, mid));
        window.add(new Line(mid, end));
        return end;
    }

    public static void main(String[] args) {
        Window window = new Window();
        window.scale(0, -1, 10, 10);
        RPoint end = new RPoint(0, 0);
        end = porras(window, end);
        end = porras(window, end);
    }
}
```

```

        end = porras(window,end);
        window.showWindow();
    }
}

```

Line -olion luominen tallettaa viitteet parametreinä vietyihin RPoint -olioihin. (6p)

4. Kerro omin sanoin ja esimerkein mitä tapahtuu aliohjelmakutsun yhteydessä. Käsittele funktiomaista aliohjelmaa, jossa on ainakin muutama parametri. Milloin mikäkin muuttuja syntyy ja milloin häviää. Mikä ero on perusmuuttujilla (boolean, char, int, double) ja oliomuuttujilla. (6p)

5. Kirjoita aliohjelma `poista(jono, kirjain)`, jolle seuraavat ComTest-testit toimisivat. Tyypit joudut miettimään itse. Samoin miten `<-` käytännössä toteutetaan. Kirjoita aliohjelmalle myös pieni pääohjelma, jossa aliohjelman toimintaa testataan.

```

TYYPPI jono <- "kissa istuu puussa"; // alustetaan jono
poista(jono, ' ') === 2; jono.toString() === "kissaistuupuussa";
poista(jono, 'u') === 4; jono.toString() === "kissaistpssa";
poista(jono, 'k') === 1; jono.toString() === "issaistpssa";
poista(jono, 'a') === 2; jono.toString() === "issistpss";
poista(jono, 's') === 5; jono.toString() === "iitp";
poista(jono, 'x') === 0; jono.toString() === "iitp";

```

6. Vastaa seuraaviin muutamien rivien selostuksella ja/tai esimerkillä (kerää max. 6p edestä osatehtäviä):

a) Miten Java-ohjelma käännetään ja ajetaan komentoriviltä? Entä jos ohjelma tarvitsee jonkin kirjaston avukseen? (2p)

b) Laske (laskutoimitukset mukaan) seuraavien binäärilukujen arvo 10-järjestelmän lukuna. Mitä oletuksia teit? (2p):

```

0100 0101
0011 1111
1000 0001

```

c) Oletetaan 8-bittisille negatiivisille kokonaisluville 2-komplementti -esitys. Mikä olisi luvun miinus viisi (-5) bittiesitys tällaisessa järjestelmässä. (1p)

d) Binäärinen desimaaliluku `1001.1110` 10-järjestelmän lukuna. (1p)

e) Pohdi missä mielessä `ArrayList` on parempi kuin tavallinen taulukko. Missä mielessä huonompi? (2p)

f) Mikä ero on `String` ja `StringBuilder` -luokilla? (1p)

g) Paljonko on `34/7` Javassa. Paljonko on `34%7` Javassa. (1p)