

## Demo 4 / 8.2

### Pisteytyksestä

Pisteet aina niin, että yksi tehtävä on yksi piste (jos tehty ja ohjelmatehtävä koneella testattu). Jos samassa tehtävässä on kaksi tehtävänumeroa, esim. 2\*3, niin silloin siinä on kaksi tehtävää, eli pisteet 1+1. Jos yhden pisteen tehtävässä on kohdat a ja b, niin vastaavasti yksi kohta on 0.5 pistettä. Jos tehtävässä on vaikka 4 selvästi eri funktiota tai kohtaa, niin silloin tehtävän voi ajatella koostuvan 4 osasta. Esimerkiksi tehtävässä 7\*8 on käsketty tehdä luokka, 2 funktiota ja testi joten sen 2 pistettä voi ajatella koostuvan 4\*0.5 pistettä. Syöttöä vaativissa tehtävissä saa käyttää Ali.jar:in funktioita.

Ja muistakaa että Java-ohjelmien on käännyttävä ja toimittava kuten pyydettiin sekä sisällettävä Javadoc-tyyliset kommentit, jotta sen saa merkitä täyden veroisena.

Tehtävissä pyydetty testit voit kirjoittaa toistaiseksi joko:

1. itse tehdyllä pääohjelmalla, ks: [Etusuku2.java](#) tai [Poista.java](#)
2. Junit-testinä, ks: [PoistaTest.java](#)
3. ComTest -testinä, ks: <https://trac.cc.jyu.fi/projects/ohj2/wiki/ohj2ComTest> ja [Poista.java](#)

### Tehtävät

- 1\*. Kirjoita Java-ohjelma, joka toimisi seuraavasti:

```
C:\OMA>java Ika[RET]
Kerron sinulle paljonko täytät tänä vuonna!
Anna syntymävuotesi>1978[RET]
Täytät tänä vuonna 30 vuotta.
```

- 2\*3. Kirjoita funktio, joka palauttaa suorakulmaisen kolmion hypotenuusan pituuden kun parametrina on kateettien pituus. Kirjoita testi. Kirjoita vielä funktio `kolmion_ala`, joka laskee vastaavista tiedoista suorakulmaisen kolmion pinta-alan. Testaa tämäkin funktio. Muista että funktiot eivät tulosta mitään!
4. Kirjoita seuraavat funktiot tai aliohjelmat sekä testit, jolla testataan niiden toimivuus.

- a) Funktio `int etumerkki(double x, double eps)` joka palauttaa

```
-1 mikäli x<-eps,   eps on aina >= 0
0  mikäli -eps<=x<=eps
1  mikäli x>eps
```

- b) Funktio `demopisteet` jolle viedään opiskelijan tekemien demopisteiden summa ja demotehtävien maksimimäärä ja funktio palauttaa sitten Ohjelmointi 2-kurssin [taulukon](#) mukaiset demopisteet.

- 5\*. Tutki pöytätestin avulla mitä ovat muuttujien arvot seuraavassa ohjelmassa kunkin lauseen suorittamisen jälkeen. Mitä ohjelma tulostaa?

```

/**
 * Mitä ohjelma tulostaa??
 * @author Vesa Lappalainen
 * @version 1.0, 19.01.2003
 */
public class Alisotk2 {

    private static int i(StringBuffer s) {
        return Integer.parseInt(s.toString());
    }
    private static void set(StringBuffer s,int i) {
        s.delete(0, s.length()).append(""+i);
    }

    /* 01 */ private static StringBuffer b; private static int c;
    /* 02 */
    /* 03 */ private static void s_1(StringBuffer a, int b)
    /* 04 */ {
    /* 05 */     int d;
    /* 06 */     d = i(a);
    /* 07 */     c = b + 3;
    /* 08 */     b = d - 1;
    /* 09 */     set(a,c - 5);
    /* 10 */ }
    /* 11 */
    /* 12 */ private static void a_2(int a, StringBuffer b)
    /* 13 */ {
    /* 14 */     c = a + i(b);
    /* 15 */     { int c; c = i(b);
    /* 16 */     a = 8 * c; }
    /* 17 */     set(b,175);
    /* 18 */ }
    /* 19 */
    /* 20 */ public static void main(String[] args) {
    /* 21 */     StringBuffer a = new StringBuffer("4"); int d=9;
    /* 22 */     System.out.println(" " + a + " " + b + " " + c + " " + d);
    /* 23 */     b=new StringBuffer("3"); c=2; d=1;
    /* 24 */     s_1(b,c);
    /* 25 */     a_2(d,a);
    /* 26 */     s_1(a,3+d);
    /* 27 */     System.out.println(" " + a + " " + b + " " + c + " " + d);
    /* 28 */ }
}

```

6. Mitä ovat muuttujien arvot seuraavien ohjelmanpätkien jälkeen? Sisennä (ja selvennä) ohjelmanpätkät "oikein"! Huomaa että alla on kahdessa sarakkeessa 4 toisistaan riippumatonta koodinpätkää.

<pre> /* 1 */ a=4; b=2; c=0; if ( a-b != 0 ) c++; a=1; b=3; </pre>	<pre> /* 2 */ a=4; b=2; c=0; if ( c != 0 )     c--; else b=1; a = b-2; </pre>
<pre> /* 3 */ a=4; b=2; c=0; if ( a &gt; 0 ) if ( b &gt; 1 )     c=9; </pre>	<pre> /* 4 */ a=4; b=2; c=0; if ( c++ != 0 ) if ( a-- != 0 ); else b-=3; else c+=8; </pre>

```

else if ( c == 1 ) ;      |
else b = 0;              |
a = 3;                   |

```

- 7\*8. Kirjoita luokka `Henkilo`, jossa on (attribuutteina) sukunimi, etunimi ja syntymävuosi. Kirjoita metodi `kysy`, jossa kysytään henkilön tiedot olion attribuutteihin. Kirjoita metodi `tulosta`, joka tulostaa henkilön tiedot (ks. [monisteen luku 9.2](#)). Kirjoita myös `main`-metodi jossa luodaan ainakin yksi henkilö ja kutsutaan noita metodeja. Tällä kertaa tämä ei vielä tarvitse automaattista testiä. Pyri alla olevan kaltaiseen pääohjelmaan:

```

public static void main(String[] args) {
    Henkilo hlo = new Henkilo();
    Henkilo aku = new Henkilo("Aku", "Ankka", 1934);
    hlo.tulosta();    aku.tulosta();
    hlo.kysy();      hlo.tulosta();
    aku.kysy();      aku.tulosta();
}

```

jolloin tulostus olisi:

```

0
Aku Ankka 1934
Henkilön etunimi () >Matti[ret]
Henkilön sukunimi () >Meikä[ret]
Henkilön syntymävuosi (0) >1970[ret]
Matti Meikä 1970
Henkilön etunimi (Aku) >[ret]
Henkilön sukunimi (Ankka) >[ret]
Henkilön syntymävuosi (1934) >1935[ret]
Aku Ankka 1935

```

- B1. Katso Ohjl-kurssin [Wikistä](#) miten käytetään Graphics.jar-kirjastoa. Ota malliohjelma [UkkoSample.java](#) ja muuta se sellaiseksi, että ainakin yksi ukko pyörittää jokaista jalkaansa ja kättänsä.

- G1-4. Katso [Demo 3:n guru-tehtävässä G1-2](#) käsiteltyä tiedostoa `users.html` ja tiedostoa [files.txt](#). Kirjoita ohjelma, joka tarkistaa jokaiselta henkilöltä että hänellä on kussakin vastaustiedostossaan ainakin oma nimensä. Jos nimeä ei ole, `files.txt`:tä ei ole tai `files.txt`:ssä mainittua tiedostoa ei ole, henkilölle lähetetään sähköposti, jossa mainitaan asiasta. Sähköposti lähetetään tekemällä Unix-konetta varten seuraavanlainen komentojono `postita`:

```

#!/bin/sh
mailx -s Demol vesal@jyu.fi <<EOF
Tiedostossa kavely.txt ei ole nimeäsi.
Tiedostoa guru2.cpp ei löydy.
EOF
mailx -s Demol mattim@jyu.fi <<EOF
Tiedostoa files.txt ei löydy.
EOF
mailx -s Demol teppot@jyu.fi <<EOF
Tiedostoa t9.cpp ei löydy.
EOF

```

`postita` ajettaisiin (siis demossa ei tarvitse ajaa) sitten sopivassa Unix/Linux koneessa.

Jos Linuxissa ei ole komentoa `mailx` voidaan vaikka tehdä alias.

Myöskään WWW:stä ei tarvitse osata lukea, vaan riittää osata lukea normaalista tiedostojärjestelmästä. Jos `users.html` on hakemistossa `demo1` (esim. `demo09\demo1`), niin henkilön `files.txt` on alihakemistossa käyttäjänimi (eli esim. `demo09\demo1\vesal\files.txt`). Vastaavasti vastaustiedostot ovat suhteessa tähän hakemistoon. Oletetaan että jos käyttäjän tunnus on `vesal`, niin hänen sähköpostiosoitteensa on `vesal@jyu.fi` (oikeasti koko postiosoite on käytössä ja se ehkä tuleekin `users.html`:ään)

**Vinkki:** Käytä "hajota ja hallitse" -tekniikkaa. Teet mm. aliohjelmiä joita kutsutaan:

```
kasitteleKaytaja("vesal",virhetiedosto);  
if ( onkoTiedostossa("kavely.txt","Vesa Lappalainen") ) ...
```

Lisäideoita katso [Demon 3](#): mallivastauksesta [LueUsers.java](#).