

# Demo 3 / 31.1

## Tehtävät

Viitteissä tyyliin "Luvun 8.5" noudatetaan luentomonisteen numerointia. Demoissa saa käyttää vapaasti aliohjelmanpaketteja [ohj2/ali](#). Hakemistossa on kirjaston käyttöohje. Java-ohjelmien testit saa tehdä aina myös [ComTestiä](#) tai [JUnittia](#) käyttäen, eli ei ole pakko kirjoittaa pääohjelmää jos sitä ei erikseen vaadita. Mitä palautetaan? Itse tehdyt tiedostot. Eli Javsta .java. Jos testit tehdään ComTestillä niin se generoi toisen \*Test.java tiedoston joka siis ei ole itse kirjoitettu ja tätä EI palauteta. Mutta jos testit tehdään Junitilla ja ITSE kirjoitetaan tuo testitiedosto, niin silloinse toki palautetaan. \*.class -tiedostoja EI MISÄÄN NIMESSÄ palauteta.

1. Kirjoita algoritmi joka tarkistaa onko merkkijono sama kuin kysymysmerkkejä sisältävä merkkijono (? = mikä merkki vaan). Kysymysmerkki tarkoittaa siis mitä tahansa yhtä kirjainta. TDD: keksi lisää erilaisia testattavia asioita.

```
jono      maski
"Kissa"   "K?ss?"   => samat
"Kiss"    "K?ss?"   => ei samat
```

2. TDD: Java-toteutus ja testipääohjelma edelliselle algoritmille. Testin saa tehdä myös ComTestiä käyttäen.

```
public static boolean onkoSamatKys(String jono, String maski);
...
    if ( onkoSamatKys("Kissa","K?ss?") ) ...
```

- 3\*. Kirjoita algoritmi, jolle annetaan kaksi merkkijonoa ja joka poistaa toisesta jonosta kaikki toisen jonon esiintymät. Lopputulokseen ei saa jäädä enää yhtään poistettavan merkkijonon esiintymää (vihje: tee "runsaasti" apualiohjelmia):

```
jono      poista      tulos
Catcat    at              Cc
Paatti    at              Pi
Puatit    at              Puit
```

TDD: keksi lisää esimerkkejä jotka pitää testata.

Kokeile osaatko tehdä Java-toteutusta jossa on automaattinen testi (vapaaehtoinen +1 bonuspiste, eli jos algoritmi JA Java-toteutus, voit merkitä yhteensä kaksi pistettä. Pelkällä Java-toteutuksella vain yksi piste. Algoritmi pitää silloinkin olla!).

- 4\*. Toteuta luokka, jolla kuvataan päivämäärä. Kirjoita ainakin sopiva muodostaja ja metodi toString, jolla päivämäärä saadaan merkkijonoksi. Luonnollisesti testit tavalla tai toisella. Katso mallia: [Aika.java](#).
- 5\*. Suunnittele ja kirjoita Java-ohjelma, jossa on dialogi (frame) huoneesta mitattujen tietojen syöttämiseksi ja sitten näiden perusteella lasketaan ja näytetään huoneen pinta-ala ja tilavuus. Toteuta ohjelma luennon esimerkin ja [monisteen 8.5 luvun](#) mukaisesti aliohjelmia

käyttäen . Katso malliksi [Mittakaava.java](#).

- 6\* Demokerran 2 mallivastauksessa [Etusuku2.java](#):ssa on kaksi samankaltaista aliohjelmää. Muuta nämä yhdeksi aliohjelmaksi vaihdaAlkuLoppu (sopiva määrä parametrejä, mieta mitä) ja toteuta sitten alkuperäiset kaksi aliohjelmää yhden rivin aliohjelmina jotka kutsuvat vaihdaAlkuLoppu.
- 7\*. Näytä kuvan avulla (piirrä kuva kunkin sijoituksen jälkeen uudelleen) mitä ovat muuttujien arvot seuraavien sijoitusten jälkeen (kun muuttujat ovat sijoittuneet muistipaikkoihin kuten kuvassa). Piirrä kuvaan myös mihin viitteet b, c ja p loogisesti aina "osoittavat".

ks. [Viitteet.java](#):

```
int a;
StringBuffer b = new StringBuffer("1");
StringBuffer c = new StringBuffer("2");
StringBuffer p;
a = 19;
p = b;
p.append("0");
p = c;
p.append(a);
c = b;
c.append("3");
```

	a = 19	p = b	p.app(0)	p = c	p.app(a)	c = b	c.app(3)
100	a						
104	900   b						
108	940   c						
112	??    p	o+-	o+-	o+-	o+-	o+-	o+-

Keko:

900	1						
940	2						

8. Kirjoita Java-ohjelma, joka lukee yhden merkkijonon (rivin) ja tulostaa toisen merkkijonon siten, että merkkijonon 1. ja 2. sana ovat aakkosjärjestyksessä. Tulostetaan aina vain kaksi sanaa, vaikka syötetyssä merkkijonossa olisi useampikin sana. Toteutus mielellään funktion `jarjesta_1_2`-avulla (eli kirjoita ko. funktio, vrt [Etusuku2.java](#)). TDD: Kirjoita "kaikki mahdolliset" testattavat tapaukset. Jos teet "automaattisen" testin, ei tarvitse kysyä merkkijonoa. Merkkijonon saa ottaa ja näyttää myös graafisessa käyttöliittymässä jos haluaa.
- B1. Kirjoita algoritmi (vrt. tehtävä 1) joka tarkistaa onko merkkijono sama kuin mahdollisesti YHDEN (tai ei yhtään, mutta ei enempää, jos saa olla monta \*, niin asia vaikeutuu oleellisesti) \*-merkin sisältävä jono (\*:n kohdalla voi siis olla miten monta, myös 0, ja mitä merkkiä tahansa)

```

jono      maski
"Kissa"   "K*a"    => samat
"Kissa"   "K*i"    => ei samat
"Kissa"   "K*"    => samat
"Lintu"   "K*"    => ei samat

```

Mikäli kuitenkin haluat kokeilla useamman tähden toteutusta, niin vinkiksi yksi sana: **rekursio**.

## G1-2 Hae tiedosto [users.html](#):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<base target=m1>
</head>
<body>
93 <a href="vesal/index.html?r=">Vesa Lappalainen</a> = 8.5<br>
94 <a href="mattim/index.html?r=">Matti Meikäläinen</a> = 8<br>
95 <a href="teppot/index.html?r=">Teppo Teikäläinen</a> = 9<br>
</body>
</html>

```

```

ja tee sitten ohjelma, joka lukee tiedoston ja tulostaa sen muodossa:
93 Vesa Lappalainen = 8.5 - vesal
94 Matti Meikäläinen = 8 - mattim
95 Teppo Teikäläinen = 9 - teppot

```

eli riisuu pois kaikki rivit, joissa ei ole linkkiä ja linkkiriveistä tulostetaan em. tiedot em. muodossa

## Java vinkkejä

### Miten merkkijono luetaan:

```

String s = Syotto.kysy("Anna jono");
System.out.println("Jono oli: \"" + s + "\"");

```

### Tiedosto luetaan:

```

import java.io.*;

/**
 * Luetaan tiedosto ja tulostetaan se näytölle.
 * @author Vesa Lappalainen
 * @version 1.0, 25.01.2002
 */
public class TiedostonLukeminen {

    public static void main(String[] args) throws IOException {
        BufferedReader f = null;
        try {
            f = new BufferedReader(new FileReader("TiedostonLukeminen.java"));
        } catch (FileNotFoundException ex) {
            System.out.println("Tiedosto ei aukea!"); return;
        }

        try {

```

```
        String rivi;
        while ( ( rivi = f.readLine() ) != null ) {
            System.out.println(rivi);
        }
    } catch (IOException ex) {
        System.out.println("Vikaa tiedostoa lukiessa!"); return;
    } finally {
        f.close();
    }
}
}
```

Merkkijonon muuttaminen numeroksi:

```
String s = "123";
int i = Mjonot.erotaInt(s,0);
```