

Demo 12 / 7.4 klo 16:15-18:00

Huom! Palautus tiistaina 7.4 klo 14 mennessä. Demotilaisuus luennon poikkeuksellisesti luennon jälkeen.

Taas vähintään kahteen vastaukseen ComTest- tai Junit -testit.

1-2 Muuta ohjelman (ks. demo 11 vastaukset.) [ValuuttaMuunnos.java](#) tietorakenne Valuutat TreeMap-rakenteeksi.

3. Toimiiko [ValuuttaMuunnos.java](#) (ks. demo 11 vastaukset) siten, että sitä voidaan käyttää seuraavasti ja jos voi niin miksi:

```
Määrä ja valuutta>10 s[RET]
10.00 Skr on 6.00 mk.
Määrä ja valuutta>2[RET]
2.00 Skr on 1.20 mk.
Määrä ja valuutta>[RET]
Kiitos!
```

4-6. Kirjoita ohjelma joka toimisi seuraavasti (kiinnitä huomiota siihen, että komentovalikoimaa voidaan ohjelmaan lisätä "kohtuullisen" helposti.

```
Anna komento>Ynnää 3 4 5[RET]
Tulos on 12.
Anna komento>Isoksi kana[RET]
kana isona on KANA
Anna komento>palindromiko saippuakivikauppias[RET]
Sana "saippuakivikauppias" on palindromi!
Anna komento>Ynnää 2 1[RET]
Tulos on 3.
Anna komento>Yn 2 1[RET]
Tulos on 3.
Anna komento>Yh 2 1[RET]
En tunne komentoa YH!
Anna komento>[RET]
```

Mallivastauksen Java-versioon tuli mm.

```
public interface KomentoRajapinta {
    boolean suorita(String parametrit);
}

public static class Ynnaa implements KomentoRajapinta { ...
public static class Apua implements KomentoRajapinta { ...
public static class Palindromiko implements KomentoRajapinta { ...
public static class Isoksi implements KomentoRajapinta { ...

... luokkia Komento ja Komennot (sis. mm. metodin: etsi) ...

public static void main(String[] args) {
```

```

Komennot komennot = new Komennot();
Apua apua = new Apua("komento.txt");

komennot.add(new Komento("?", apua));
komennot.add(new Komento("apua", apua));
komennot.add(new Komento("+", new Ynnaa()));
komennot.add(new Komento("ynnää", new Ynnaa()));
komennot.add(new Komento("isoksi", new Isoksi()));
komennot.add(new Komento("palindromiko", new Palindromiko()));

String s;

while ( true ) {
    s = Syotto.kysy("Anna komento");
    if ( s.equals("") ) break;
    komennot.tulkitse(s);
}
}

```

7*. Vaikket tekisikään tehtävää 4-6, niin lisää siihen [Help.java](#) käyttäen avustus. Eli jos et ole tehnyt tehtävää 4-6, niin kirjoita pääohjelma joka ? painamisella tulostaa avustuksen käyttäen [Help.java](#) metodeja. Tutki [ali-](#) hakemistosta [Help.java](#). Kirjoita myös alkeellinen avustustiedosto.

8. Kirjoita luokka Kello jossa on metodi tarkista(String jono), joka tarkistaa onko parametrina viety merkkijono oikeata muotoa kellonajaksi. Jos on vikaa, niin funktio palauttaa merkkijonona vian, muuten null-viitteen. Aloita testin tekemisellä.

B1-2 Kirjoita ohjelma, joka kyselee nimiä kunnes annetaan tyhjä rivi ja tulostaa tämän jälkeen nimet aakkosjärjestyksessä. Nimet syötetään muodossa "*Aku Ankka*", mutta siitä huolimatta ne lajitellaan siten, että "*Iines Ankka*" < "*Hannu Hanhi*", eli sukunimen mukaan.

G1-4 Hevosen hyppely. Hevonenhan voi hypätä shakkilaudalla 2 yhteen suuntaan ja yhden toiseen suuntaan. Tee ohjelma joka etsii hevoselle reitin 5x7 laudalla niin, että hevonen käy tasan yhden kerran jokaisessa ruudussa. Vinkkejä: Lauta -luokka, jossa on "ruutu" jokaiselle paikalle. "Ruutuun" tallennetaan monesko hyppy tulee tähän ruutuun. 0 on vapaa ruutu. "Siirtovektori" hevosen mahdollisille hypuille. Tehtävässä etsitään rekursiivista ratkaisua (backtracking, <http://en.wikipedia.org/wiki/Backtracking>), mutta ongelmalle on myös heuristinen lineaarinen algoritmi, ks. esimerkiksi <http://web.telia.com/~u85905224/knight/eWarnsd.htm>.

K1-2 Alla pieni pala Korpista. Poista turhat olion luonnit. Liittykö mitenkään varauksiin?

```

/**
 * Method that is needed when getting reservationids from tables,
 * compressed values (etc. 1,2,3) are separated to different rows in
 * String array
 * @param arrayWithCommaValues array that has ids separated with commas
 * in one row
 * @return String array in which comma separated values are separated
 */
public static String[] getSeparatedArray(String[] arrayWithCommaValues)

```

```
{
    List resList = new ArrayList();
    String[] reservations2 = new String[0];
    if (arrayWithCommaValues != null) {
        for (int i = 0; i < arrayWithCommaValues.length; i++) {
            String[] tmpReservations = arrayWithCommaValues[i].split(",");
            if (tmpReservations != null) {
                for (int j = 0; j < tmpReservations.length; j++) {
                    resList.add(tmpReservations[j]);
                }
            }
        }
        reservations2 = new String[resList.size()];
        for (int i = 0; i < resList.size(); i++) {
            reservations2[i] = (String) resList.get(i);
        }
    }
    return reservations2;
}
```

GURU -tehtävät

Guruillekin suosittelen tuota 4-7 tehtävää.

Kurssin loppuosa ja jatko

Tämä demo (12) on viimeinen kurssin 8 op:n Java-osuutta koskeva demo. Loput demot (C1 ja C2) ovat 1 op:n C++ "kurssissa" (TIEP112). Lisäksi WWW-osuus (JSP - Java Server Pages) on 1 op:n TIEP113 kurssissa.

Vaiheessa 6 pitää olla tiedostojen lukeminen ja kirjoittaminen sekä luokkien välinen yhteistyö käyttöliittymäluokan puolella.

Vaiheessa 7. täytyy olla valmiina ainakin (sovittava ohjaajan kanssa henkilökohtaisesti)

- päätteeltä luku (joillakin oikeellisuustarkistuksilla)
- jokin etsiminen tai vastaava (esim. valikoiva tulostus) jossa koko tietorakenne täytyy käydä valikoiden läpi
- jokin oma (selvästi itse tehty) toiminto malliohjelmaan nähden (esim. mallissahan tulostus ei toimi)
- KOMMENTIT ainakin jokaisen tiedoston alussa ja kunkin metodin alussa JavaDocin mukaisesti. Laittakaa muuten kunkin tiedoston alkukommentteihin myös tekijöiden email, jos pitää olla yhteydessä.
- käännöksessä ei saa tulla yhtään virhettä eikä varoitusta
- tiedostojen alkukommenteissa on lueteltava ne asiat, jotka kustakin tiedostosta on tekemättä. Sen tiedoston alussa, jossa on pääohjelmakin, on kerrottava mitkä ohjelman osat eivät toimi