

Tentti / 18.12

Vastaa yhteensä **neljään** tehtävään (huomaa että tehtävissä voi olla useita alakohtia), joista yksi on tehtävä 5. Voit siis valita kolme tehtävistä 1–4 ja tehtävä **5 on pakollinen**. Vastaa **JOKAISEN tehtävänumeron vastaus erilliselle konseptipaperille** (joka tehtävälle eri tarkastaja). Jos vastaat kaikkiin viiteen, tarkastajilla on oikeus valita 4 huonointa vastausta. Tenttiaika 4 tuntia.

1. Java-koodi

1a). Selvitä perustellen, mitä alla oleva ohjelma tulostaa. (3 p):

```
public class Hypercool {
    public static void main(String[] args) {
        int a = 1;
        int b = 1;
        System.out.print("1");
        for (int i = 0; i < 10; i++) {
            System.out.print(" " + a);
            int d = a;
            a += b;
            b = d;
        }
    }
}
```

1b) Käyttäjätunnusgeneraattori. Täydennä alla oleva aliohjelma teeTunnus. Huom! Koodi on ”pala” isompaa ohjelmaa. (3 p)

```
/**
 * Muuttaa kirjaimen käyttäjätunnukseen kelpaavaksi.
 * Kirjain muutetaan pieneksi ja skandeista poistetaan pisteet.
 * Tuntemattomat kirjaimet ja mahdolliset välimerkit korvataan
 * 1-merkillä.
 * @param c muutettava kirjain
 * @return käyttäjätunnukseen kelpaava kirjain
 * @example
 * <pre name="test">
 *   muutaKirjain(' ') === '1';
 *   muutaKirjain('2') === '1';
 *   muutaKirjain('A') === 'a';
 *   muutaKirjain('B') === 'b';
 *   muutaKirjain('a') === 'a';
 *   muutaKirjain('Ä') === 'a';
 *   muutaKirjain('Ö') === 'o';
 *   muutaKirjain('ö') === 'o';
 * </pre>
 */
public static char muutaKirjain(char c) {
    final String mitka = "ääö";
    final String miksi = "aao";
    char lc = Character.toLowerCase(c);
    int i = mitka.indexOf(lc);
    if ( i >= 0 ) return miksi.charAt(i);
    if ( lc < 'a' ) return '1';
}
```

```

        if ( lc > 'z' ) return '1';
        return lc;
    }

/**
 * Aliohjelma muuttaa nimen käyttäjätunnukseksi.
 * Käyttäjätunnus muodostetaan yhdistämällä käyttäjän
 * etunimen ensimmäinen kirjain ja sukunimen ensimmäiset
 * seitsemän merkkiä, jolloin käyttäjätunnuksesta tulee 8 merkkiä pitkä.
 * Jos sukunimessä ei ole seitsemää merkkiä, sukunimi otetaan mukaan
 * kokonaisuudessaan.
 * Kirjaimet ä, ö ja å korvataan kirjaimilla a, o ja a.
 * Käyttäjätunnus sisältää ainoastaan pieniä kirjaimia.
 * @param etunimi muutettavan nimen etunimi
 * @param sukunimi muutettvan nimen sukunimi
 * @return nimien perusteella muodostettu käyttäjätunnus
 * @example
 * <pre name="test">
 *   teeTunnus("Martta","Hänninen")   === "mhannine"
 *   teeTunnus("Åke","Ek")             === "aek"
 *   teeTunnus("Björn","Björnström")  === "bbjornst"
 * </pre>
 */
public static String teeTunnus(String etunimi, String sukunimi) {
    /// TÄYDENNÄ TÄMÄ...
}

```

2. Matriisin transpoosi

Matriisin (= 2-ulotteinen taulukko) transpoosi saadaan, kun alkuperäisin matriisin rivit muutetaan sarakkeiksi ja päinvastoin.

Esimerkkejä:

```

5  2  -8  2  7          5  1
1  4  -3  12 27      transpoosi  2  4
                                   -8 -3
                                   2  12
                                   7  27
-----
2  5          2  -4
-4 10      transpoosi  5  10

```

- 2a) Kirjoita matriisin transpoosin muodostamisesta algoritmi suomeksi. (2 p)
- 2b) Toteuta algoritmisi Java-kielillä. Tee siis (funktio)aliohjelma, joka palauttaa parametrina saamansa matriisin transpoosin. Matriisi annetaan ja palautetaan 2-ulotteisena taulukkona. Javadoc-kommentteja ei poikkeuksellisesti tarvitse kirjoittaa. (4 p)

3. Java-kielen rakenteesta

Vastaa alla oleviin kysymyksiin lyhyesti.

- 3a) Miten Java-ohjelmalle viedään parametreja komentorivillä ja miten käyttäjän antamat parametrit saadaan selville ohjelmakoodissa? (1p)
- 3b) Mitkä ovat oliotietotyyppisten ja alkeistietotyyppisten muuttujien erot ja yhtäläisyydet? (1p)
- 3c) Mitä pitää ottaa huomioon eri tietotyyppisiä yhtäsuuruusoperaattorilla vertailtaessa? (1p)
- 3d) Mitä tarkoittaa ylivuoto ja kuinka se pitää ottaa huomioon? (1p)
- 3e) Minkä tietotyyppin valitsisit seuraaville tiedoille? Perustele vastauksesi. Anna muuttujille myös nimi. (1p)
- i. mansikoiden määrä kiloina
 - ii. henkilötunnus
 - iii. vuosiluku
 - iv. onko kurssilla tilaa
- 3f) Mitä eroa on String- ja StringBuilder-olioilla? (1p)

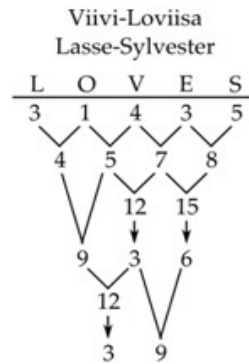
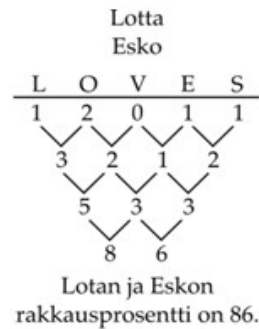
4. Yleistä Java-tietoa

Vastaa alla oleviin kysymyksiin lyhyesti (max. sivu/vastaus)

- 4a) Mitä on kommentointi ja miksi koodia kannattaa kommentoida? Anna esimerkki Javadoc-tyylisestä kommentista aliohjelmaan joka tutkisi onko sana palindromi (sama sana luki sen kumminpäin tahansa) vai ei. (2p)
- 4b) Mitä toistorakenteita Java-kielessä on ja miten ne eroavat toisistaan? Anna jokaiseen mielekäs käyttökohde. (2p)
- 4c) Kuinka Java-koodi muuttuu suoritettavaksi ohjelmaksi? Mikä on Java-virtuaalikoneen tehtävä? Mitä hyötyä Javan virtuaalikoneesta on? (2p)

5. Rakkausprosentti

- Alla esimerkki rakkausprosentin laskemisesta, eli LOVES sanan alle kirjoitetaan montako kertaa kukin kirjain löytyy tutkittavista nimistä. Sitten lukuja lasketaan pareittain yhteen uudelle riville niin kauan, että jäljellä on kaksi numeroa. Yli 9 olevat luvut muutetaan yksinumeroiseksi luvuksi summaamalla luvun numerot keskenään.



Viivi-Loviisan ja Lasse-Sylvesterin
rakkausprosentti on 39.

- Varsinainen tehtävä: Lisää alla olevaan ohjelmaan puuttuva aliohjelma ja täydennä ne kohdat, joissa lukee TÄYDENNÄ siten, että syntyy toimiva rakkausprosentin laskeva ohjelma. Laita vastauspaperiin T ja numero siitä, monettako "aukkoa" täydennät (aukkojen arvot arvostelussa eivät ole yhtä suuria) (6p)

- Vapaaehtoinen lisäbonus: Keksi kaksi sanaa, joilla saat maksimaalisen prosentoin (1p).

```
package tentti09;
```

```
import java.util.Scanner;
```

```
/**
```

```
 * Ohjelma kysyy kaksi nimeä ja laskee nimien perusteella
 * kaksinumeroisen rakkausprosentin. Rakkausprosentti lasketaan
 * laskemalla "LOVES"-sanan jokaisen kirjaimen lukumäärät kummassakin
 * nimessä. Esimerkiksi nimissä "Lotta" ja "Esko" on yhteensä
```

```
 * <pre>
```

```
 * - 1 kpl L-kirjainta
 * - 2 kpl O-kirjaimia
 * - 0 kpl V-kirjainta
 * - 1 kpl E-kirjainta
 * - 1 kpl S-kirjainta.
```

```
 * </pre>
```

```
 * Suur- ja pienaakkosilla ei ole merkitystä.
```

```
 * Kirjaimien lukumäärät summataan vierekkäisten kirjaimien lukumäärien
 * kanssa yhteen, jolloin jäljelle jää neljä lukumäärää.
```

```
 * Jos lukumäärien summasta tulee kaksi- tai useampinumeroinen,
 * niin silloin luvun numerot summataan yhteen
```

```
 * (esimerkiksi luvusta 12 tulee 3).
```

```
 * Vierekkäisten kirjaimien lukumäärien summaamista jatketaan kunnes
 * jäljellä on kaksi lukua. Nämä kaksi lukua muodostavat rakkausprosentin.
```

```
 * @author Ohjl tuntiovet
```

```
 * @version 16.12.2009
```

```
 */
```

```
public class Rakkauslaskuri {
```

```
    /** Sana jonka kirjaimiin verrataan */
```

```
    public static final String RAKKAUSSANA = "LOVES";
```

```
    /**
```

```
     * Laskee yhteen luvun numerot
     * @param luku tutkittava luku
     * @return luvun numeroiden summa
     * @example
```

```

* <pre name="test">
*   summaaNumerot(0) === 0
*   summaaNumerot(12) === 3
*   summaaNumerot(123) === 6
*   summaaNumerot(78) === 15
* </pre>
*/
public static int summaaNumerot(int luku) {
    int summa = 0;
    int n = luku;
    while ( n > 0 ) {
        summa += n % 10;
        n = n / 10;
    }
T1   return TÄYDENNÄ;
}

/**
* Muuttaa luvun yksinumeroikseksi laskemalla luvun numerot
* yhteen niin kauan että tulos on yksinumeroinen.
* @param luku tutkittava luku
* @return luvun numeroiden summa (aina yksi numero)
* @example
* <pre name="test">
*   yksiNumeroinenSumma(1)   === 1;
*   yksiNumeroinenSumma(12)  === 3;
*   yksiNumeroinenSumma(123) === 6;
*   yksiNumeroinenSumma(78)  === 6;
* </pre>
*/
public static int yksiNumeroinenSumma(int luku) {
T2   int tulos = luku;
    while ( tulos > 9 ) tulos = summaaNumerot(TÄYDENNÄ);
    return tulos;
}

/**
* Laskee taulukon vierekkäiset luvut yhteen ja palauttaa
* uuden taulukon, jossa on vierekkäisten lukujen summa.
* Jos summan luvusta tulee kaksi tai useampinumeroinen,
* sen numerot lasketaan yhteen kunnes se on yksinumeroinen.
* @param a laskettava taulukko
* @return vierekkäisten summa (yhtä lyhyempi kuin a)
* @example
* <pre name="test">
*   Arrays.toString(laskeVierekkaisetYhteen(new int[]{1}))      === "[]";
*   Arrays.toString(laskeVierekkaisetYhteen(new int[]{1,1}))    === "[2]";
*   Arrays.toString(laskeVierekkaisetYhteen(new int[]{2,3,5}))  === "[5, 8]";
* </pre>
*/
public static int[] laskeVierekkaisetYhteen(int[] a) {
T3   int[] a2 = new int[a.length - 1]; // Lukujen määrä vähenee aina yhdellä.
    for (int i = 0, j = 1; j < a.length; i++, j++) {
        a2[i] = yksiNumeroinenSumma(TÄYDENNÄ);
    }
T4   return TÄYDENNÄ;
}

/**

```

```

* Laskee kuinka monta mitäkin sanan kirjainta löytyy jonosta.
* @param sana jonka kirjaimia tutkitaan
* @param jono josta kirjaimia etsitään
* @return kirjainten lukumäärät kokonaislukutaulukossa
* @example
* <pre name="test">
* #import java.util.Arrays;
* Arrays.toString(laskeKirjaimet("apu","saapas")) === "[3, 1, 0]";
* Arrays.toString(laskeKirjaimet("apu","")) === "[0, 0, 0]";
* Arrays.toString(laskeKirjaimet("Apua","saapas")) === "[0, 1, 0, 3]";
* </pre>
*/
public static int[] laskeKirjaimet(String sana,String jono) {
    int[] a = new int[sana.length()];
    for (int si = 0; si < a.length; si++) {
        char c = sana.charAt(si);

        int lkm = 0;
        for (int ji=0; ji<jono.length(); ji++)
            T5 TÄYDENNÄ // jos sama kirjain paikoissa si ja ji kasvata lkm

            a[si] = lkm;
        }
    return a;
}

/**
* Lasketaan sanojen rakkausprosentti.
* @param nimil
* @param nimi2
* @return nimien rakkausprosentti
*
* @example
* <pre name="test">
* laskeRakkausprosentti("Lotta","Esko") === 86
* laskeRakkausprosentti("Viivi-Loviisa","Lasse-Sylvester") === 39
* </pre>
*/
T6 TÄYDENNÄ

/**
* Lasketaan rakkausprosentti käyttäjän syötteistä
* @param args ei käytössä
*/
public static void main(String[] args) {
    System.out.println("Ohjelma laskee kahdesta nimestä rakkausprosentin.");
    Scanner scan = new Scanner(System.in);
    System.out.print("Anna ensimmäinen nimi: ");
    String nimil = scan.nextLine();
    System.out.print("Anna toinen nimi: ");
    String nimi2 = scan.nextLine();
    System.out.printf("Rakkausprosenttinne on %d!%n",
        laskeRakkausprosentti(nimil, nimi2));
}
}

```