

Ohjelmoinnin perusteet, kurssikoe 21.10.2014

Vastaa jokainen tehtävä erilliselle konseptille. Kirjoita jokaiseen konseptiin kurssin nimi, kokeen päivämäärä, nimi, TMC-tunnus ja opiskelijanumero tai henkilötunnus. Vastaukset palautetaan tehtäväkohtaisiin pinoihin.

Vastaa kokeen lisäksi tässä koepaperissa olevaan kyselyyn, ja palauta se koevastausten mukana. Käytä tähän kyselyyn korkeintaan 15 minuuttia. Kyselyyn rehellisesti vastaamisesta saat yhden ylimääräisen koepisteen.

Nimi: _____

Opiskelijanumero: _____

Käytin tähän kurssiin yhteensä noin _____ tuntia.

Kävin _____ tämän kurssin luennolla (esim. kolmella).

Vastasin noin _____ prosenttiin materiaalissa olleista selitystehtävistä (esim 80).

Tein noin _____ prosenttia kaikista materiaalissa olleista ohjelmointitehtävistä (esim 80).

Seuraavassa on selitystehtäviin liittyviä väittämiä. **Jos väite on mielestäsi täysin totta, ympyröi numero yksi. Jos väite on mielestäsi täysin epätotta, ympyröi numero kuusi.** Jos väite pätee kohdallasi enemmän tai vähemmän, ympyröi sopiva numero lukujen yksi ja kuusi väliltä. Jos et vastannut selitystehtäviin, tai et osaa sanoa, ympyröi vaihtoehto 0.

Materiaalin selitystehtävät							
.. auttoivat kokeeseen kertaamisessa	1	2	3	4	5	6	0
.. eivät auttaneet ohjelmointitehtävien tekemisessä	1	2	3	4	5	6	0
.. eivät tuoneet lisäarvoa ohjelmointitehtäviin	1	2	3	4	5	6	0
.. toivat lisäarvoa materiaaliin	1	2	3	4	5	6	0
.. olivat työläitä	1	2	3	4	5	6	0
.. olivat vaikeita	1	2	3	4	5	6	0
.. tukivat opiskeluani	1	2	3	4	5	6	0
Materiaalin selitystehtävistä							
.. oli hyötyä	1	2	3	4	5	6	0
.. ei ollut hyötyä materiaalin ymmärtämisessä	1	2	3	4	5	6	0
.. oli hyötyä omien väärinkäsitysteni huomaamisessa ja korjaamisessa	1	2	3	4	5	6	0
.. oli hyötyä ohjelmointitehtävissä käsiteltyjen asioiden ymmärtämisessä	1	2	3	4	5	6	0
.. oli hyötyä oman osaamiseni tarkastelussa	1	2	3	4	5	6	0
Pidin selitystehtäviin vastaamisesta	1	2	3	4	5	6	0
Kertasin kurssimateriaalia ennen koetta	1	2	3	4	5	6	0

En vastannut selitystehtäviin, koska

Vapaata palautetta materiaalissa olleista selitystehtävistä

Kiitos jo nyt ja tsemppiä kokeeseen!

Kurssikoe

Vastaa jokainen tehtävä erilliselle konseptille. Kirjoita jokaiseen konseptiin kurssin nimi, kokeen päivämäärä, nimi, TMC-tunnus ja opiskelijanumero tai henkilötunnus. Vastaukset palautetaan tehtäväkohtaisiin pinoihin.

Voit lyhentää komennon `System.out.println()` kirjoittamalla `sout()` ja käyttää muitakin lyhenteitä, mikäli selität lyhenteen jokaisessa paperissa, jossa sitä käytät. Voit olettaa kaikissa tehtävissä, että ohjelman runko on seuraava:

```
import java.util.Scanner;

public class Ohjelma {
    public static void main(String[] args) {
        Scanner lukija = new Scanner(System.in);
        // int luku = Integer.parseInt( lukija.nextLine() );
        // String rivi = lukija.nextLine();
    }
}
```

1. Käsitteistö ja koodin luku (5p)

(a) Vastaa jokaiseen kohtaan lyhyesti ja ytimekkäästi. Anna myös jokaisessa kohdassa konkreettinen esimerkki. Koko tehtävän vastauksen pituudeksi riittää noin kaksi sivua.

- i. Mikä on muuttuja? Mitä tarkoitetaan muuttujan tyypillä? Miten alkeis- ja viittaustyyppiset muuttujat eroavat toisistaan? Anna muutama esimerkki alkeis- ja viittaustyyppisistä muuttujista. (1p)
- ii. Mikä on metodi? Miten ja mihin metodeja käytetään? (1p)
- iii. Mitä ovat olio ja luokka? Mikä on olion ja luokan välinen suhde? (1p)

(b) (2p) Käytössäsi on seuraava lähdekoodi.

```
public class Auto {
    private int bensaa;
    private String merkki;

    public Auto(String merkki) {
        this.bensaa = 0;
        this.merkki = merkki;
    }

    public Auto(String merkki, int bensaa){
        this.bensaa = bensaa;
        this.merkki = merkki;
    }

    public void tankkaa(int litraa){
        this.bensaa += litraa;
    }

    public void aja() {
        if (this.bensaa > 0) {
            this.bensaa--;
            System.out.println( this.merkki + " ...");
        } else {
            System.out.println( this.merkki + " ei bensaa...");
        }
    }
}
```

Selitä mitä allaolevassa autosimulaattorikoodissa tapahtuu kun se suoritetaan. Kerro rivi riviltä mikä muuttuu, miten, missä, milloin, ja miksi. Voit myös piirtää kuvia havainnollistamaan ohjelman suoritusta. **Huom:** käytä ohjelmointikäsitteistä niiden "oikeita nimiä": luokka, olio, oliomuuttuja, muuttuja, metodi, parametri, paluuarvo, staattinen metodi.

Sopiva pituus tehtävän vastaukselle on noin 1 sivu.

```
public class Autosimulaattori {

    public static void main(String[] argh) {

        Auto auto1 = new Auto("bmw");
        Auto auto2 = new Auto("lada");

        vieraileHuoltsikalla(auto1);
        auto1.aja();
        auto1.aja();

        auto2.aja();

        Auto auto3 = new Auto("mersu", 1000);
        auto3.aja();
    }

    public static void vieraileHuoltsikalla(Auto auto) {
        auto.tankkaa(10);
    }
}
```

2. Ohjelmoinnin perusasiat (4p)

- (a) (2p) Tee ohjelma, joka tulostaa toistolausetta (esim. while tai for) käyttäen kaikki 8:lla jaolliset kokonaisluvut aloittaen luvusta 999 ja päättyen lukuun 1. Tulostuksen tulee tapahtua siten, että jokaiselle riville tulostetaan 4 lukua, tämän jälkeen tulostus alkaa seuraavalta riviltä. Luvut tulee tulostaa sarakkeittain siten, että kunkin sarakkeen lukujen viimeisen numeron tulee sijaita luvun koosta siippumatta samalla kohtaa.

Ohjelman tulostuksen tulee näyttää **täsmälleen** seuraavalta:

```
992 984 976 968
960 952 944 936
928 920 912 904
896 888 880 872
(paljon rivejä välissä)
64  56  48  40
32  24  16   8
```

- (b) (2p) Tee ohjelma, joka lukee käyttäjän syöttämiä kokonaislukuja kunnes käyttäjä syöttää luvun 0. Luettuaan luvut, ohjelma ilmoittaa suurimman syötetyn luvun ja suurimman peräkkäisten lukujen erotuksen.

Esimerkki:

Syötä lukuja, 0 lopettaa:

```
5
3
7
-4
4
0
```

```
suurin luku: 7
suurin erotus: 11
```

Esimerkissä luku 7 on suurin luku, ja suurin peräkkäisten lukujen erotus on lukujen 7 ja -4 välillä ($7 - -4 = 11$). Huomioi, että ohjelman tulee jättää huomioimatta luku 0.

Esimerkki:

```
Syötä lukuja, 0 lopettaa:
-5
-7
-2
-9
0
suurin luku: -2
suurin erotus: 7
```

Esimerkissä luku -2 on suurin luku, suurin peräkkäisten lukujen erotus 7 on lukujen -2 ja -9 välillä ($-2 - -9 = 7$).

3. Metodeja (4 p)

- (a) (2p) Tee metodi `public static void tulostaVali(int raja1, int raja2)` joka tulostaa luvuista lukuväliltä 1-99 nousevassa järjestyksessä ne, jotka eivät ole parametrien määrittelemällä välillä.

Voit olettaa, että ensimmäinen parametri on aina pienempi tai yhtäsuuri kuin toinen parametri.

Eli esim. jos kutsutaan `tulostaVali(3, 98)` tulostuu

```
1 2 99
```

ja jos kutsutaan `tulostaVali(5, 90)` tulostuu

```
1 2 3 4 91 92 93 94 95 96 97 98 99
```

- (b) (2p) Tee metodi `public static int parillistenSumma(int[] luvut)` joka palauttaa parametrina annetun taulukon parillisten lukujen summan.

Esim. Kun metodi saa parametrina taulukon `[1,2,3,4,5]`, se palauttaa arvon 6.

Tee myös pääohjelma, joka demonstroi metodin käyttöä.

Huom: jos et osaa käyttää taulukkoja, voit tehdä metodista version `public static int parillistenSumma(ArrayList<Integer> luvut)` jossa metodi saa parametrina luvut sisältävän `ArrayList`:in.

4. (7 p) Tehtävässä tehdään sähköposteista huolehtiva ohjelma.

(a) (4 p) Ohjelmoi ensin luokka `Email`. Luokan pitää toimia seuraavasti:

```
public static void main(String[] args) {
    Email maili1 = new Email("koeasiaa", "pekka@helsinki.fi", "...\\n");
    System.out.println(maili1);
}
```

Tulostuu:

```
lahettaja: pekka@helsinki.fi
aihe: koeasiaa
...
```

Tee siis luokalle

- konstruktori, jolla asetetaan viestin aihe, lähettäjä sekä sisältö
- metodi `toString()`, joka palauttaa viestin tekstiesityksen (yo. esimerkin tapaan muotoiltuna)

Jokainen viesti on aluksi lukematon ja muuttuu jossain vaiheessa luetuksi. Tee tämän tiedon ylläpitoa varten luokalle vielä seuraavat metodit:

- `onkoLukematon()`, joka palauttaa `true` jos viesti on lukematon ja muuten `false`
- `asettaLuetuksi()`, joka asettaa viestin luetuksi
- `asettaLukemattomaksi()`, joka asettaa viestin takaisin lukemattomaksi

(b) (3 p) Ohjelmoi luokka `Postilaatikko`.

`Postilaatikko`:ssa pitäisi olla `Email`-olioita sisältävä `ArrayList`. Huolehdi `ArrayList:n` alustuksesta ja ohjelmoi luokkaan seuraavat metodit:

- `public void vastaanota(Email viesti)`, lisää parametrina saadun viestin postilaatikon `ArrayList:iin`. Viestin tila vastaanottamisen jälkeen on lukematon.
- `public void naytaLukemattomat()`, tulostaa kaikki viestit joiden tila on lukematon. Viestien tila muuttuu tämän jälkeen luetuksi.
- `public void naytaViestitJoidenLahettaja(String lahettaja)`, tulostaa kaikki parametrina olevalta lähettäjältä tulleet viestit, jos jokin näytetty viesti oli lukematon, muutetaan se luetuksi.

Seuraava pääohjelma havainnollistaa ohjelmoitavien luokkien käyttöä:

```
public static void main(String[] args) {
    Postilaatikko inbox = new Postilaatikko();

    Email maili1 = new Email("koeasiaa", "pekka@helsinki.fi", "...\\n");
    Email maili2 = new Email("syysloma?", "mikko@helsinki.fi", "...\\n");

    inbox.vastaanota(maili1);
    inbox.vastaanota(maili2);

    inbox.naytaLukemattomat();
    System.out.println("-----");

    Email maili3 = new Email("opinto-opas", "pekka@helsinki.fi", "...\\n");
    inbox.vastaanota(maili3);

    inbox.naytaLukemattomat();
    System.out.println("-----");
}
```

```

maili1.asetaNukemattomaksi();
inbox.naytaLukemattomat();
System.out.println("-----");

inbox.naytaViestitJoidenLahettaja("pekka@helsinki.fi");
}

```

Ohjelman tulostus oikein toteutetuilla luokilla on:

```

lahettaja: pekka@helsinki.fi
aihe: koeasiaa
...

```

```

lahettaja: mikko@helsinki.fi
aihe: syysloma?
...

```

```

-----
lahettaja: pekka@helsinki.fi
aihe: opinto-opas
...

```

```

-----
lahettaja: pekka@helsinki.fi
aihe: koeasiaa
...

```

```

-----
lahettaja: pekka@helsinki.fi
aihe: koeasiaa
...

```

```

lahettaja: pekka@helsinki.fi
aihe: opinto-opas
...

```

ArrayListin käyttöohje

- Konstruktori `public ArrayList<T>()` luo uuden `ArrayList`-olion, jossa listan alkiot ovat tyyppiä `T`.
- `public boolean add(T x)` lisää listan loppuun olion `x`.
- `public int get(int i)` palauttaa listan alkion kohdasta `i`.