

Ohjelmoinnin perusteet, kurssikoe 26.2.2013

Vastaa tehtävät 1, 2, 3 ja 4 erillisille konsepteille. Kirjoita jokaiseen konseptiin kurssin nimi, kokeen päivämäärä, nimi ja opiskelijanumero. Vastaukset palautetaan tehtäväkohtaisiin pinoihin. Vaikka jättäisit johonkin tehtävään vastaamatta, tulee vastauspaperi siinäkin tapauksessa palauttaa.

1. (3 p) Vastaa seuraavaan lyhyesti ja ytimekkäästi kokonaisiin lausein. Kannattaa myös käyttää koodiesimerkkejä.

Mitä tarkoitetaan alkeis- ja viittaustyyppisillä muuttujilla? Miten alkeistyyppiset ja viittaustyyppiset muuttujat poikkeavat toisistaan jos niitä käytetään metodien parametrina?

2. Ohjelmoinnin perusasiat (5 p)

Tämän ja seuraavan tehtävän kohdissa voit olettaa, 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();
    }
}
```

- (a) (2 p) Tee ohjelma, joka kysyy käyttäjältä kokonaislukua joka vastaa kuukauden numeroa eli on väliltä 1-12. Tämän jälkeen ohjelma tulostaa käyttäjän syöttämää kuukausinumeroa seuraavien kolmen kuukauden numerot. Ohjelma voi olettaa, että käyttäjän syöte on kunnollinen, eli oikealla välillä oleva luku.

Seuraavassa kaksi esimerkkiä ohjelman toiminnasta:

```
Syötä kuukauden numero:
3
Seuraavat kolme kuukautta ovat:
4 5 6
```

```
Syötä kuukauden numero:
11
Seuraavat kolme kuukautta ovat
12 1 2
```

- (b) (3 p) Tee ohjelma, joka kysyy käyttäjältä kokonaislukuja, kunnes käyttäjä antaa luvun joka ei ole väliltä 1-100. Ohjelma tulostaa syötettyjen lukujen neliöiden summan (ainoastaan välillä 1-100 oleet luvut huomioidaan summaan).

Seuraavassa esimerkki ohjelman toiminnasta:

```
syötä lukuja väliltä 1-100. välin ulkopuoleinen lopettaa
2
3
1
3
-6
syöttämiesi lukujen neliöiden summa: 23
```

Tulostettu luku 23 siis saatiin laskemalla $2 * 2 + 3 * 3 + 1 * 1 + 3 * 3$.

3. Metodeja, listoja ja taulukoita (8 p)

- (a) (4 p) Tee metodi `public static boolean sisältavatSaman(int[] luvut1, int[] luvut2)` joka palauttaa `true` jos sen parametrinaan saamat taulukot sisältävät vähintään yhden saman luvun ja muuten `false`. Parametrina annettuja taulukoita ei saa muuttaa millään tavalla.

Esim. parametreilla `luvut1 = [1,2,3]` ja `luvut2 = [5, 6]` metodi palauttaisi `false` ja parametreilla `luvut1 = [1,2,3]` ja `luvut2 = [2, 9]` `true`.

Tee myös pääohjelma, joka demonstroi metodin käyttöä.

Huom: jos et osaa käyttää taulukkoja, voit tehdä metodista version `public static boolean sisältavatSaman(ArrayList<Integer> luvut1, ArrayList<Integer> luvut2)` jossa siis parametrit sisältävän lukuja sisältäviä `ArrayList`:jä

- (b) (4 p) Tee metodi `public static ArrayList<Integer> duplikaatioton(ArrayList<Integer> luvut)` joka palauttaa uuden listan, joka sisältää samat luvut kuin sen parametrina saama lista, mutta jolla sama luku ei esiinny kuin kertaalleen. Parametrina annettua listaa ei saa muuttaa millään tavalla.

Esim. jos parametrina olevalla listalla on luvut `3, 2, 3, 5, 3, 1, 5`, palauttaa metodi uuden listan, joka sisältää luvut `3, 5, 2, 1`,

4. (14 p) Tehtävässä toteutetaan yksinkertainen kurssikirjanpitosovellus

- (a) (4 p) Toteuta ensin luokka `Opiskelija`. Luokan pitää toimia seuraavasti:

```
public static void main(String[] args) {
    Opiskelija op = new Opiskelija("Pekka", 45);
    System.out.println( op );
    System.out.println("opiskelijan "+op.getNimi() + " pistemäärä "+op.getPisteet());
}
```

Tulostuu:

```
Pekka 45 pistettä
opiskelijan Pekka pistemäärä 45
```

Tee siis luokalle

- konstruktori, jolla asetetaan opiskelijan nimi sekä pistemäärä
- metodi `toString()`, joka palauttaa opiskelijan tekstiesityksen (yo. esimerkin tapaan muotoiltuna)
- metodit `getNimi()` ja `getPisteet()` jotka palauttavat opiskelijan nimen ja pisteet

- (b) (10 p) Toteuta luokka `Kurssi`.

`Kurssi`:ssa tulee olla `Opiskelija`-olioita sisältävä `ArrayList`, sekä seuraava toiminnallisuus:

- konstruktori, jolla asetetaan kurssin nimi sekä läpipääsyyn vaadittava pistemäärä
- metodi `lisaaOpiskelija(Opiskelija lisattava)`, joka lisää parametrina annettun opiskelijan kurssille
- metodi `tulokset()`, joka tulostaa kurssin tulokset alla olevan esimerkin tapaan
- metodi `korkeinPistemaara()`, joka palauttaa korkeimman kurssilla saadun pistemäärän

Seuraava pääohjelma havainnollistaa ohjelmoitavien luokkien käyttöä:

```

public static void main(String[] args) {
    Kurssi ohja = new Kurssi("Ohjelmoinnin jatkokurssi", 30);

    Opiskelija o1 = new Opiskelija("Pekka", 45);
    Opiskelija o2 = new Opiskelija("Liisa", 51);
    Opiskelija o3 = new Opiskelija("Matti", 25);

    ohja.lisaaOpiskelija(o1);
    ohja.lisaaOpiskelija(o2);
    ohja.lisaaOpiskelija(o3);

    ohja.tulokset();

    System.out.println("Korkein pistemäärä kurssilla "+ohja.korkeinPistemaara());
}

```

Ohjelman tulostus oikein toteutetuilla luokilla on:

```

Kurssin Ohjelmoinnin jatkokurssi tulokset:
Pekka 45 pistettä HYVÄKSYTTY
Liisa 51 pistettä HYVÄKSYTTY
Matti 25 pistettä HYLÄTTY
Korkein pistemäärä kurssilla 51

```

ArrayList:n 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`.
- `import`-rivi on `import java.util.ArrayList`