

Ohjelmoinnin perusteet, kurssikoe 28.2.2012

Vastaa tehtävät 1, 2 ja 3 **erillisille** konsepteille. Kirjoita jokaiseen konseptiin kurssin nimi, kokeen päivämäärä, nimi, nimikirjoitus ja opiskelijanumero. Vastaukset palautetaan tehtäväkohtaisiin pinoihin.

Vaikka jättäisit johonkin tehtävään vastaamatta, tulee vastauspaperi siinäkin tapauksessa palauttaa.

1. Käsitteistöä (6p)

Selitä mitä seuraavat termit tarkoittavat. Vastaa jokaiseen kohtaan lyhyesti ja ytimekkäästi. Koko tehtävän vastauksen pituudeksi riittää noin sivu.

- (a) muuttuja
- (b) metodi
- (c) parametri
- (d) paluuarvo
- (e) luokka
- (f) olio

2. Ohjelmoinnin perusasiat (12 p)

Tämän 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);
        // koodi tulee tähän

        // muistutus Scannerin käytöstä:
        // int luku = Integer.parseInt( lukija.nextLine() );
        // String rivi = lukija.nextLine();
    }
}
```

- (a) Tee ohjelma, joka tulostaa 3:lla jaolliset kokonaisluvut välillä 3-999. Ohjelman tulostuksen tulee näyttää seuraavalta:

```
3
6
9
12
(paljon rivejä välissä)
999
```

- (b) Tee ohjelma, joka toistaa käyttäjän antaman nimen yhtä monta kertaa kuin nimessä on kirjaimia.

Esimerkki 1:

```
Anna nimi: Sari
Sari
Sari
Sari
Sari
```

Esimerkki 1:

```
Anna nimi: Markus
Markus
Markus
Markus
Markus
Markus
Markus
```

Esimerkiksi nimessä Sari on neljä kirjainta, joten ohjelma toistaa nimen neljä kertaa.

- (c) Tee ohjelma, jolle annetaan kellonaika tasatuntina ja siihen lisättävä tuntimäärä. Ohjelma ilmoittaa, mitä kello on lisäyksen jälkeen. Molemmat kellonajat ovat kokonaislukuja välillä 0-23.

Esimerkki 1:

```
Anna kellonaika: 14
Anna lisäys: 5
Uusi kellonaika: 19
```

Esimerkki 2:

```
Anna kellonaika: 21
Anna lisäys: 8
Uusi kellonaika: 5
```

Esimerkki 3:

```
Anna kellonaika: 10
Anna lisäys: 1000
Uusi kellonaika: 2
```

3. Olio-ohjelmointi (12 p)

Tehtävässä tehdään yksinkertainen puhelinmuistio.

- (a) (6 p) Ohjelmoi ensin luokka `Henkilo`. Luokan pitää toimia seuraavan esimerkin osoittamalla tavalla:

```
public static void main(String[] args) {
    Henkilo pekka = new Henkilo("Pekka Mikkola", "040-123123");
    System.out.println( pekka );

    pekka.vaihdaNumeroa("050-333444");
    System.out.println( pekka );
}
```

Tulostuu:

```
Pekka Mikkola puh: 040-123123
Pekka Mikkola puh: 050-333444
```

Tee siis luokalle

- konstruktori, jolla asetetaan henkilölle nimi ja puhelinnumero
- metodi `vaihdaNumeroa(String uusiNumero)`, joka muuttaa henkilön puhelinnumeroa
- metodi `toString()`, joka palauttaa henkilön merkkijonoesityksen (yo. esimerkin tapaan muotoiltuna)

Ja näiden lisäksi vielä metodit:

- `getNimi()`, joka palauttaa nimen
- `getNumero()`, joka palauttaa puhelinnumeron

(b) (6 p) Ohjelmoi luokka Puhelinmuistio.

Puhelinmuistio:ssa pitäisi olla Henkilo-olioita sisältävä ArrayList. Huolehdi ArrayList:n alustuksesta ja ohjelmoi luokkaan seuraavat metodit:

- void lisää(String nimi, String numero), luo Henkilo-olion ja lisää sen puhelinmuistion ArrayList:iin.
- String haeNumero(String nimi), palauttaa parametrina annetun henkilön numeron. Jos henkilö ei ole muistiossa, palautetaan tieto tästä (ks. esimerkki alla)
- void tulostaKaikki(), tulostaa puhelinmuistion sisällön

Seuraava pääohjelma havainnollistaa ohjelmoitavien luokkien käyttöä:

```
public static void main(String[] args) {
    Puhelinmuistio muistio = new Puhelinmuistio();
    muistio.lisaa("Pekka Mikkola", "040-123123");
    muistio.lisaa("Antti Laaksonen", "045-456123");
    muistio.lisaa("Juhana Laurinharju", "050-222333");

    String numero = muistio.haeNumero("Pekka Mikkola");
    System.out.println( numero );

    numero = muistio.haeNumero("Martti Tienari");
    System.out.println( numero );

    System.out.println("");
    muistio.tulostaKaikki();
}
```

Ohjelman tulostus oikein toteutetuilla luokilla on:

```
040-123123
numero ei tiedossa
```

```
Pekka Mikkola puh: 040-123123
Antti Laaksonen puh: 045-456123
Juhana Laurinharju puh: 050-222333
```

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