

Ohjelmoinnin perusteet, kurssikoe 3.6.2013

Kirjoita jokaiseen konseptiin oma nimi, opiskelijanumero, kokeen päivämäärä sekä TMC tunnukset.

1. (6p) Vastaa jokaiseen kohtaan lyhyesti ja ytimekkäästi, anna jokaisesta kohdasta konkreettinen esimerkki. Koko tehtävän vastauksen pituudeksi riittää noin kaksi sivua.
 - (a) Mikä on muuttuja? Mitä tarkoitetaan muuttujan tyyppillä? Miten alkeis- ja viittaustyyppiset muuttujat eroavat toisistaan? Anna muutama esimerkki alkeis- ja viittaustyyppisistä muuttujista. (2p)
 - (b) Mikä on metodi? Miten ja mihin metodeja käytetään? (2p)
 - (c) Mitä ovat olio ja luokka? Mikä on olion ja luokan välinen suhde? (2p)
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) (2p) Tee ohjelma, joka tulostaa 3:lla jaolliset kokonaisluvut välillä 3-1000. Ohjelman tulostuksen tulee näyttää seuraavalta:

```
3
9
12
15
18
21
(paljon rivejä välissä)
999
```

- (b) (3p) Tee ohjelma, joka kysyy käyttäjältä positiivisen kokonaisluvun ja pyöristää sen ylöspäin seuraavaan kuudella jaolliseen lukuun.

Esimerkki 1:

```
Anna luku: 7
Pyöristys: 10
```

Esimerkki 2:

```
Anna luku: 41
Pyöristys: 45
```

Esimerkki 3:

```
Anna luku: 20
Pyöristys: 20
```

Esimerkki 4:

Anna luku: 2

Pyöristys: 5

3. Lisää perusvääntöä (7 p)

- (a) (4p) Kirjoita ohjelma, joka kysyy käyttäjältä merkkijonoja, kunnes käyttäjä antaa tyhjän merkkijonon. Sitten ohjelma tulostaa lyhyimmän käyttäjän antaman epätyhjän merkkijonon. Jos sopivia merkkijonoja on useita, ohjelma voi tulostaa minkä tahansa niistä.

Esimerkki 1:

Anna merkkijonoja:

tammi

helmi

maalis

huhti

Lyhin: tammi

Esimerkki 2:

Anna merkkijonoja:

apina

banaani

cembalo

Lyhin: apina

Esimerkki 2:

Anna merkkijonoja:

Jarmo

Kalle

Matti

Tony

Lyhin: Tony

- (b) (3p) Tee metodi `public static int montakoKertaaEsiintyy(int[] luvut, int luku)` joka palauttaa tiedon, montako kertaa paremetrinä annettu luku esiintyy parametrinä annetussa taulukossa. Mikäli lukua ei taulukosta löydy, palauttaa se 0.
Esim. Metodi taulukon `[1,2,3,4,5,5]` ja luvun 5 parametrina saadessaan palauttaa 2.
Tee myös pääohjelma, joka demonstroii metodin käyttöä.

Huom: jos et osaa käyttää taulukkoja, voit tehdä metodista version `public static int jarestyksessa(ArrayList<Integer> luvut, int luku)` jossa se siis saa parametrikseen luvut sisältävän `ArrayList`:in.

4. (12 p) Tehtävässä tehdään yksinkertainen sähköposteista huolehtiva ohjelma.

(a) (6 p) Ohjelmoi ensin luokka `Email`. Luokan pitää toimia seuraavasti:

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

Tulostuu:

```
lahettaja: pekka@helsinki.fi
aihe: oluelle?
...
```

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`
- `asetaNluetuksi()`, joka asettaa viestin luetuksi
- `asetaNlukemattomaksi()`, joka asettaa viestin takaisin lukemattomaksi

(b) (6 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("oluelle?", "pekka@helsinki.fi", "...\\n");
    Email maili2 = new Email("syysloma alkaa", "mikko@helsinki.fi", "...\\n");

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

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

    Email maili3 = new Email("syksyn 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: oluelle?
...

```

```

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

```

```

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

```

```

-----
lahettaja: pekka@helsinki.fi
aihe: oluelle?
...

```

```

-----
lahettaja: pekka@helsinki.fi
aihe: oluelle?
...

```

```

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

```

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`