

R0027 – Ohjelmoinnin perusteet

Kehittämisharjoitukset

E: Metodit

Viopen luku 7

METODEITA

1. AakkosetJaNumerot.java

Tee komentoriviohjelma, jossa on kolme metodia:

- `main(String[] args)`
- `tulostaAakkoset()`
- `tulostaNumerot()`

Kutsu pääohjelmasta (eli `main()` -metodista) aakkosmetodia, numerometodia ja lopuksi taas aakkosmetodia.

2. LoremIpsum.java

Kutsu pääohjelmasta (eli `main()` -metodista) metodia `tulostaLoremIpsumLauseKymmenesti()`. Rakenna jälkimmäinen metodi sellaiseksi, että siellä suoritetaan silmukkarakenne, joka hoitaa tekstin tulostuksen ruudulle 10 kertaa.

Esimerkkitulostus:

"Lorem Ipsum dolor sit amet, consectetur adipiscing elit. "

Kun saat ohjelman toimimaan, kehitä sitä edelleen siten, että silmukan sisältä ei suoraan tulosteta mitään, vaan kutsutaan kolmatta metodia nimeltä `tulostaLoremIpsumLause()`, joka hoitaa yhden rivin tulostuksen.

Tarkastele ohjelmasi etenemistä käyttäen Eclipsen Debug-ominaisuutta.

- Lisää breakpoint `main`-metodin ensimmäiselle riville.
- Debug as Java application
- Step into, step over, step over, step into...

3. LukujenSumma.java

Tee metodi `summa(int ekaLuku, int tokaLuku)`, joka palauttaa kahden parametrina sille välitetyn arvon summan. Kutsu kyseistä metodia päämetodista, sijoita metodin palauttama arvo johonkin muuttujaan ja tulosta muuttujan paluuarvo konsoliin päämetodissa. Tee kutsu ja tulostus kolme kertaa peräkkäin päämetodissa eri arvoja käyttäen.

Vinkki: Paluuarvon voit ottaa talteen erilliseen muuttujaan, jos tarvitset sitä myöhemmin ohjelmassa ao. tapaan:

```
int summa = summa(5,2);
System.out.println( summa );
```

4. YmpyränKaari.java

Tee metodi laskeYmpyränKaari(double sade), joka palauttaa säteen avulla lasketun ympyrän kaaren double-arvona. Kirjoita metodin sisään algoritmi, joka laskee parametrina saadun säteen avulla ympyrän kaaren pituuden (kaari = $2 \cdot \pi \cdot \text{sade}$). Kutsu uutta metodiasi päämetodista kolme kertaa, kolmella eri arvolla ja tulosta palautuneet arvot konsoliin päämetodissa.

Huomaa, että paluuarvoa ei tarvitse ottaa talteen erilliseen muuttujaan, jos et tarvitse sitä myöhemmin ohjelmassa. Eli voit suoraa tulostaa metodin palauttaman arvon seuraavalla tavalla:

```
System.out.println( YmpyränKaari(5,27) );
```

5. EmailKyselija.java

Tee päämetodi, joka kutsuu kysyEmail()-metodia ja tulostaa palautetun arvon ruudulle. Toteuta metodi kysyEmail() siten, että se kysyy käyttäjältä sähköpostiosoitteen ja palauttaa sen kutsujalle. kysyEmail() kysyy sähköpostiosoitetta uudestaan ja uudestaan, kunnes annettu merkkijono sisältää merkin '@'.

Vihje: Apua @-merkin löytämiseen: String- luokan [indexOf](#)-metodilla voit etsiä esiintyykö jokin merkki/merkkijono annetussa Stringissä. Metodi palauttaa -1 jos etsittävää ei löytynyt ja sen alkamispaikan merkkijonossa (esim. 4) jos merkki löytyy.

```
char merkki = '@';  
int merkinSijainti = merkkijono.indexOf(merkki);  
boolean loytyi = (merkinSijainti != -1);
```

6. Tee sovellus, jossa käyttäjän antamasta tekstistä hymiöt korvataan toisella tekstillä tai poistetaan kokonaan.

Tee pääohjelmaan (eli main() -metodiin) ensin metodi lueTeksti(), joka kysyy käyttäjältä annettavaa merkkijonoa ja lukee sen talteen johonkin muuttujaan.

Tee tämän jälkeen kaksi metodia korvaaHymiöt(String teksti) sekä korvaaHymiöt(String teksti, String korvaaja). Näistä ensimmäinen käy käyttäjän syöttämän tekstin läpi ja korvaa sieltä löytyvät hymiöt välilyönneillä, eli (" ") -merkillä. Jälkimmäinen metodi korvaa hymiöt parametrina annetulla merkillä. esim. "*" -merkillä.

Lopuksi molemmat metodit tulostavat siistityn version tekstistä ruudulle.

Vinkki: Käytä metodin sisällä apuna String-olion [replaceAll\(\)](#)-metodia, joka ottaa ensimmäisenä parametrina vastaan korvattavan merkkijonon ja toisena parametrina korvaavan merkkijonon.

Note however that the first parameter in the replaceAll() method is not a normal string, but a string that complies to the structure of a "regular expression". In this link you can find more info about regular expressions: <http://www.vogella.com/tutorials/JavaRegularExpressions/article.html>

*So, in this exercise, if you want to replace the ":" characters from a string using the replaceAll() method, you should call the method in the following way:
newString = originalString.replaceAll(":", "");*

7. **Tee metodi, joka osaa heittää 6-sivuista noppaa.**

Käytä hyväksesi Math-luokan random()-metodia, jonka avulla voit arpoa jonkin liukuluvun väliltä $0.0 \leq \text{luku} < 1.0$. Kertomalla ja pyöristämällä tulosta, saat arvon 1-6.

Esim.

```
int tulos = (int)(Math.random()*6+1);
```

Tee pääluokkaan for – silmukka, joka heittää noppaa ja laskee montako kertaa noppaa on heitetty. Toista nopanheittoja niin kauan, kunnes noppa on antanut 10 kertaa tuloksen 6.

Tulosta ruudulle montako nopanheittoa tämä vaati.