

Harjoitus 3 (viikko 46)

- Mikäli tehtävissä on jotain epäselvää, laita sähköpostia vastuopettajalle (jorma.laurikkala@tuni.fi).
- Muista nimetä muuttujat hyvin sekä kommentoida ja sisentää koodisi. **Kirjoita metodin otsikkoon liittyvä yleisluonteinen kommentti** metodin tarkoituksesta sekä metodin mahdollisesti saamista ja palauttamista tiedoista. Tee ohjelman alkuun kommentti, jossa on harjoituksen ja tehtävän numero sekä nimesi ja sähköpostiosoitteesi.
- **Lue syötteet Scanner-luokalla. Käytä vain yhtä Scanner-oliota. Esittele lukija vakiomuotoisena attribuuttina, jos syötteitä on luettava kahdessa tai useammassa metodissa (katso luentorungon luku 2.2).**
- **Harjoitusryhmä on siirretty** tämän viikon perjantaille 13.11. klo 12–14. Tuolloin pidettävissä Zoom-harjoituksissa saa apua ongelmakohtiin.
- **Palautusaikaa on jätetty.** Palauta vastauksesi WETO-järjestelmään (<https://wetodev.sis.uta.fi/>) viimeistään ensi viikon sunnuntaina 22.11. klo 23.55.
- WETO tarkistaa kaikki tämän harjoituskerran tehtävät automaattisesti. **WETO voi hakea ohjelmastasi tehtävänannossa määritellyä metodia ja löydetyn metodin automaattinen kutsuminen on mahdollista.** Ole siksi tarkkana erityisesti metodien otsikoiden kanssa. Määrittele parametrit täsmälleen tehtävässä pyydettyllä tavalla. Älä esimerkiksi tee metodille muita kuin tehtävässä annettuja parametreja. Kurssisivujen *Harjoitukset | Ratkaisujen tarkistus* -kohdassa on annettu lisäohjeita. Ota yhteyttä harjoitusryhmän vetä-

1. Kirjoita Javalla *HelloYou*-ohjelma. Ohjelmalla on `void`-tyyppinen *moikkaa*-metodi, joka tervehtii ohjelman käyttäjää nimellä. Käyttäjän nimi välitetään metodille `String`-tyyppisen parametrin avulla. Voit olettaa, että parametrin arvossa on aina vähintään yksi merkki.

Pääosa ohjelman ja käyttäjän välisestä vuorovaikutuksesta tapahtuu pääohjelmassa (*main*-metodissa), jossa tulostetaan alkutervehdys ja kehote antaa syöte sekä luetaan käyttäjän nimi. Metodin tuloste on esimerkkien viimeinen rivi. *Moikkaa*-metodia kutsutaan päämetodista siten, että käyttäjän syöte annetaan metodille parametriarvona.

Esimerkki ohjelman toiminnasta, kun syöte on "Conan the Librarian":

```
Hello! I shall say hello to you.  
Please, enter your name:  
Conan the Librarian  
Hello, Conan the Librarian!
```

Esimerkki ohjelman toiminnasta, kun syöte on "Brian Cohen":

```
Hello! I shall say hello to you.  
Please, enter your name:  
Brian Cohen  
Hello, Brian Cohen!
```

2. Kirjoita Javalla `void`-tyyppinen *lausuKuukaudenNimi*-metodi, joka saa parametrinaan (`int`) kuukauden järjestysluvun ja tulostaa näytölle lukua vastaavan kuukauden nimen esimerkkitulosteiden tapaan. Metodin tuloste on esimerkkien viimeinen rivi. Luku 1 vastaa tammikuuta ("January"), luku 2 vastaa helmikuuta ("February") ja niin edelleen. Voit olettaa, että metodin parametrin arvo on aina välillä 1–12.

Lue järjestysnumero käyttäjältä pääohjelmassa ja kutsu metodiasi antaen syöte metodin parametrin arvoksi. Anna ohjelman nimeksi *Month*.

Harjoitus 3 (viikko 46)

Esimerkki ohjelman toiminnasta:

```
Hello! I verbalise your input.  
Please, enter a number:  
1  
It is January.
```

Toinen esimerkki ohjelman toiminnasta:

```
Hello! I verbalise your input.  
Please, enter a number:  
3  
It is March.
```

3. Toteuta tehtävä 2.4 siten, että kertotaulu tulostetaan *tulostaKertotaulu*-nimisessä metodissa. Metodi on `void`-tyyppinen. Luku välitetään metodille `int`-tyyppisenä parametrina.

Lue luku käyttäjältä pääohjelmassa ja kutsu metodiasi antaen syöte metodin parametrin arvoksi. Ohjelman nimi on *MultiplicationTable*.

4. Toteuta tehtävä 2.3 siten, että arvoltaan keskimmäisin luku päätellään *annaKeskimmäisen*-nimisessä metodissa. Luvut välitetään metodille kolmen `int`-tyyppisen parametrin avulla. Päätelyn tulos palautetaan metodista. Metodi on `int`-tyyppinen. Metodi ei tulosta mitään.

Lue luvut käyttäjältä pääohjelmassa ja kutsu metodiasi antaen syötteet metodin parametrien arvoiksi. Sijoita metodin palauttama arvo pääohjelmassa muuttujaan ja tulosta tulos näytölle pääohjelmassa tämän muuttujan avulla. Ohjelman nimi on *MiddleNumber*.

Opettaja antaa ratkaisulle pisteen vain, kun pääohjelmassa kutsutaan omaa metodia kerran ja metodikutsun paluuarvo sijoitetaan muuttujaan.

5. Toteuta tehtävä 2.7 siten, että merkkijonon merkkien aakkosjärjestys päätellään *onkoAakkosjärjestyksessä*-nimisessä metodissa. Merkkijono välitetään metodille `String`-tyyppisen parametrin avulla. Voit olettaa, että parametriarvo on kunnossa: merkkijono on aina vähintään yhden merkin pituinen.

Päätelyn tulos palautetaan metodista `boolean`-tyyppisenä arvona. Paluuarvo on `true`, jos merkit ovat aakkosjärjestyksessä. Metodi ei tulosta mitään.

Lue merkkijono käyttäjältä pääohjelmassa ja kutsu metodiasi antaen syöte metodin parametrin arvoksi. Sijoita metodin palauttama arvo pääohjelmassa muuttujaan ja tulosta tulos näytölle pääohjelmassa tämän muuttujan avulla. Ohjelman nimi on *IncreasingString*.

Opettaja antaa ratkaisulle pisteen vain, kun pääohjelmassa kutsutaan omaa metodia kerran ja metodikutsun paluuarvo sijoitetaan muuttujaan.

Harjoitus 3 (viikko 46)

6. Tee Javalla `int`-tyyppinen *laskeEsiintymät*-metodi, joka tutkii moniko ensimmäisen merkkijonon (metodin ensimmäinen parametri) merkeistä esiintyy jälkimmäisessä merkkijonossa (metodin toinen parametri). Molemmat parametrit ovat `String`-tyyppiä. Parametrien arvoja ei tarvitse tarkistaa. Metodi ei tulosta mitään, vaan palauttaa esiintymien lukumäärän paluuarvonaan.

Oletetaan, että kukin ensimmäisen merkkijonon merkeistä esiintyy jonossaan vain kerran. Näin ollen ensimmäinen merkkijono ei voi olla esimerkiksi "aa". Suuret ja pienet kirjaimet katsotaan eri merkeiksi.

Tulos on yksi, jos ensimmäinen merkkijono on esimerkiksi "ab" ja toinen merkkijono "Big data", koska merkki 'a' esiintyy jälkimmäisessä merkkijonossa, mutta merkillä 'b' ei ole esiintymiä. Esiintymät lasketaan kyllä/ei-asteikolla. Tästä syystä edellisessä esimerkissä tulos on yksi, vaikka merkillä 'a' on kaksi esiintymää jälkimmäisessä merkkijonossa.

Molemmat merkkijonot luetaan käyttäjältä pääohjelmassa. Kutsu kirjoittamaasi metodia pääohjelmasta siten, että annat käyttäjältä lukemasi syötteet metodillesi parametriarvona arvojen järjestys huomioiden. Sijoita metodin palauttama arvo pääohjelmassa muuttujaan ja tulosta tulos näytölle pääohjelmassa tämän muuttujan avulla.

Anna ohjelmasi nimeksi *Occurrences*.

Opettaja antaa ratkaisulle pisteen vain, kun pääohjelmassa kutsutaan omaa metodia kerran ja metodikutsun paluuarvo sijoitetaan muuttujaan.

Esimerkki ohjelman toiminnasta, kun syötteet ovat "em" ja "Tampere":

```
Hello! I count character occurrences between strings.
Enter the first string:
em
Enter the second string:
Tampere
The number of occurrences is 2.
```

Toinen esimerkki ohjelman toiminnasta, kun syötteet ovat "k" ja "Kuopio":

```
Hello! I count character occurrences between strings.
Enter the first string:
k
Enter the second string:
Kuopio
The number of occurrences is 0.
```