

Harjoitus 2 (viikko 45)

- Mikäli tehtävissä on jotain epäselvää, laita sähköpostia vastuuopettajalle (jorma.laurikkala@tuni.fi).
- Muista nimetä muuttujat hyvin sekä kommentoida ja sisentää koodisi. Tee ohjelman alkuun kommentti, jossa on harjoituksen ja tehtävän numero sekä nimesi ja sähköpostiosoite.
- **Lue syötteet Scanner-luokalla. Käytä ohjelmassa vain yhtä Scanner-muuttujaa. Tietovirrassa tulee olla samanaikaisesti vain yksi lukija, jotta ohjelma toimii oikein kaikissa tilanteissa. Ohjelmasi pysähtyy ajonaikaiseen virheeseen WETOssa, jos käytät siinä useampaa kuin yhtä lukijaa.**
- Ensi viikon maanantaina klo 12–14 pidettävissä Zoom-harjoituksissa saa apua ongelmakohtiin.
- Palauta vastauksesi WETO-järjestelmään (<https://wetodev.sis.uta.fi/>) viimeistään ensi viikon torstaina 5.11. klo 23.55.
- **WETO tarkistaa kaikki tämän harjoituskerran tehtävät automaattisesti.** Varmista, että ohjelmasi toimii täsmälleen esimerkkien mukaisesti, koska tarkistus tehdään opiskelijan ja malliohjelman tulosteita vertailemalla. Huomaa, että rivien alkuun tai loppuun ei tulosteta välilyöntejä ja että kaikki tulostettavat rivit – viimeinen rivi mukaan lukien – päätetään rivinvaihtoon. Kurssisivujen *Harjoitukset | Ratkaisujen tarkistus* -kohdassa on annettu lisäohjeita. Ota yhteyttä harjoitusryhmän vetäjään, jos et keksi järjellisessä ajassa miksi WETO hylkää palautuksesi.

1. Tee Javalla *Seasons*-niminen ohjelma, joka tiedustele käyttäjältä tämän suosikkivuoden ajan ja tulostaa alla annettujen esimerkkien mukaisesti. Käytä `if-else`-lauseita.

Esimerkki ohjelman toiminnasta, kun syöte on 1:

```
Hello! I read and tell.
Which do you like most: 1) winter, 2) spring, 3) summer, 4) autumn?
1
Your favourite season of the year is winter.
```

Esimerkki ohjelman toiminnasta, kun syöte on 2:

```
Hello! I read and tell.
Which do you like most: 1) winter, 2) spring, 3) summer, 4) autumn?
2
Your favourite season of the year is spring.
```

Esimerkki ohjelman toiminnasta, kun syöte on 3:

```
Hello! I read and tell.
Which do you like most: 1) winter, 2) spring, 3) summer, 4) autumn?
3
Your favourite season of the year is summer.
```

Esimerkki ohjelman toiminnasta, kun syöte on 4:

```
Hello! I read and tell.
Which do you like most: 1) winter, 2) spring, 3) summer, 4) autumn?
4
Your favourite season of the year is autumn.
```

Esimerkki ohjelman toiminnasta, kun syöte on virheellinen eli välille 1–4 kuulumaton kokonaisluku:

```
Hello! I read and tell.
Which do you like most: 1) winter, 2) spring, 3) summer, 4) autumn?
0
Your favourite season of the year is unknown.
```

2. Toteuta ensimmäinen tehtävä `switch-case`-lauseella. Ohjelman nimi on *Seasons2*.

Harjoitus 2 (viikko 45)

3. Tee Java-ohjelma, joka lukee käyttäjältään kolme lukua ja tulostaa luvuista arvoltaan keskimmäisen. Ohjelman nimi on *MiddleNumber*.

Esimerkki ohjelman toiminnasta, kun syötteet ovat 1, 2 ja 3:

```
Hello! I determine the number in the middle.  
Please, enter the first number:  
1  
Please, enter the second number:  
2  
Please, enter the third number:  
3  
The number in middle is 2.
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat 3, 2 ja 1:

```
Hello! I determine the number in the middle.  
Please, enter the first number:  
3  
Please, enter the second number:  
2  
Please, enter the third number:  
1  
The number in middle is 2.
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat 1, 1 ja 1:

```
Hello! I determine the number in the middle.  
Please, enter the first number:  
1  
Please, enter the second number:  
1  
Please, enter the third number:  
1  
The number in middle is 1.
```

4. Kirjoita Javalla *MultiplicationTable*-niminen ohjelma käyttäjältä luetun kokonaisluvun kertotaulun tulostamisen.

Esimerkki ohjelman toiminnasta, kun syöte on 5:

```
Hello! I print a multiplication table.  
Please, enter an integer:  
5  
1 x 5 = 5  
2 x 5 = 10  
3 x 5 = 15  
4 x 5 = 20  
5 x 5 = 25  
6 x 5 = 30  
7 x 5 = 35  
8 x 5 = 40  
9 x 5 = 45  
10 x 5 = 50
```

Harjoitus 2 (viikko 45)

5. Tee Javalla *LinePrinter*-niminen ohjelma, joka tulostaa käyttäjän antaman määritelmän mukaisia rivejä kunnes käyttäjä päättää pysäyttää ohjelman. Käyttäjän syötteet määrittelevät tulostuksessa käytettävän merkin ja rivin pituuden. Tietojen lukeminen ja niitä vastaavien rivien tulostus tapahtuu **do-while-silmukan** sisällä. Silmukointia jatketaan kunnes käyttäjä antaa rivin pituudeksi luvun, joka on pienempi tai yhtä suuri kuin nolla (≤ 0). Silmukkaa ohjataan **lippumuuttujalla**. Älä käytä `break`-lausetta. Muista, että merkkijonoa ei voi kertoa kokonaisluvulla Javassa. Tarvitset siksi `do-while-silmukan` sisään toisen silmukan.

Opettaja antaa ratkaisulle pisteen vain, jos ohjelmassa on lipulla ohjattu `do-while-silmukka`.

Esimerkki ohjelman toiminnasta, kun syötteet ovat 'x', 3, 'o', 10, 'z', 1, 'e' ja 0:

```
Hello! I print lines.
Please, enter character:
x
Please, enter length:
3
xxx
Please, enter character:
o
Please, enter length:
10
oooooooooooo
Please, enter character:
z
Please, enter length:
1
z
Please, enter character:
e
Please, enter length:
0
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat "+" ja -1:

```
Hello! I print lines.
Please, enter character:
+
Please, enter length:
-1
```

6. Kirjoita Javalla *SteppingPrinter*-niminen ohjelma, joka tulostaa käyttäjän antaman merkkijonon merkit käyttäen käyttäjän määrittämää väliä. Ohjelma tulostaa merkkijonon sellaisenaan, jos väli on nolla. Näytölle tulostetaan joka toinen merkki, joka kolmas merkki ja niin edelleen, kun väli on vastaavasti yksi merkki, kaksi merkkiä ja niin edelleen. Merkkien tulostaminen aloitetaan aina merkkijonon ensimmäisestä merkistä. Ohjelma tulostaa aina ensimmäisen merkin. Muita merkkejä tulostetaan niin kauan kuin merkin indeksiarvo on pienempi kuin merkkijonon pituus. Voit olettaa, että väli on suurempi tai yhtä suuri kuin nolla (≥ 0). Javassa ei voi viipaloida merkkijonoja `[]`-notaatiolla.

Merkin voi tulostaa riviä vaihtamatta `System.out.print`-metodilla. Pelkän rivinvaihto tulostetaan lauseella: `System.out.println()` ;.

Harjoitus 2 (viikko 45)

Esimerkki ohjelman toiminnasta, kun syötteet ovat "abc" ja 0:

```
Hello! I print every n:th character of a string.  
Please, enter a string:  
abc  
Please, enter a step:  
0  
abc
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat "abc" ja 1:

```
Hello! I print every n:th character of a string.  
Please, enter a string:  
abc  
Please, enter a step:  
1  
ac
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat "abc" ja 2:

```
Hello! I print every n:th character of a string.  
Please, enter a string:  
abc  
Please, enter a step:  
2  
a
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat "abcde" ja 1:

```
Hello! I print every n:th character of a string.  
Please, enter a string:  
abcde  
Please, enter a step:  
1  
ace
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat "abcde" ja 2:

```
Hello! I print every n:th character of a string.  
Please, enter a string:  
abcde  
Please, enter a step:  
2  
ad
```

Esimerkki ohjelman toiminnasta, kun syötteet ovat "a" ja 0:

```
Hello! I print every n:th character of a string.  
Please, enter a string:  
a  
Please, enter a step:  
0  
a
```

Harjoitus 2 (viikko 45)

7. Kirjoita Javalla *IncreasingString*-niminen ohjelma, joka tutkii ovatko merkkijonon merkit aakkosjärjestyksessä. Tehtävässä voi käyttää vertailuoperaattoria, koska merkeillä on Unicode-lukukoodit ja kirjaimet a–z ovat Unicode-merkistössä aakkosjärjestyksessä. Yhden merkin mittaisen merkkijonon katsotaan olevan aakkosjärjestyksessä. Voit olettaa, että syötteessä on vain pieniä kirjaimia a–z.

Esimerkki ohjelman toiminnasta, kun syöte on "abc":

```
Hello! I check whether string characters are in alphabetical order.  
Please, enter a string:  
abc  
Characters are in alphabetical order.
```

Esimerkki ohjelman toiminnasta, kun syöte on "ba":

```
Hello! I check whether string characters are in alphabetical order.  
Please, enter a string:  
ba  
Characters are not in alphabetical order.
```

Esimerkki ohjelman toiminnasta, kun syöte on "cat":

```
Hello! I check whether string characters are in alphabetical order.  
Please, enter a string:  
cat  
Characters are not in alphabetical order.
```

Esimerkki ohjelman toiminnasta, kun syöte on "almost":

```
Hello! I check whether string characters are in alphabetical order.  
Please, enter a string:  
almost  
Characters are in alphabetical order.
```

Esimerkki ohjelman toiminnasta, kun syöte on "aaa":

```
Hello! I check whether string characters are in alphabetical order.  
Please, enter a string:  
aaa  
Characters are in alphabetical order.
```

Esimerkki ohjelman toiminnasta, kun syöte on "x":

```
Hello! I check whether string characters are in alphabetical order.  
Please, enter a string:  
x  
Characters are in alphabetical order.
```

8. BizzBuzz-peliä pelataan siten, että pelaajat istuvat piirissä ja joku aloittaa pelin sanomalla "1". Seuraava pelaaja sanoo joko seuraavan luvun, "bizz", "buzz" tai "bizz buzz". Luku korvautuu sanalla "bizz", jos luku on kolmella jaollinen. Sana "buzz" korvaa luvun, jos luku on viidellä jaollinen. Luvun sijasta sanotaan "bizz buzz", kun luku sekä kolmella että viidellä jaollinen. Peli jatkuu kunnes joku takeltelee tai sanoo väärän vastauksen.

Esimerkki: 1, 2, bizz, 4, buzz, bizz, 7, 8, bizz, buzz, 11, bizz, 13, 14, bizz buzz, 16, ...

Harjoitus 2 (viikko 45)

Kirjoita Java-kielellä *BizzBuzz*-niminen ohjelma, joka kysyy käyttäjältä pelivuorojen lukumäärän ja tulostaa näytölle oikeiden vastauksien sarjan. Voit olettaa, että syöte on aina oikeellinen (> 0).

Esimerkki ohjelman toiminnasta, kun käyttäjä antaa vuorojen määräksi 5:

```
Hello! This is a BizzBuzz simulation.  
Please, enter the number of turns:  
5  
1, 2, bizz, 4, buzz.
```

Esimerkki ohjelman toiminnasta, kun käyttäjä antaa vuorojen määräksi 15:

```
Hello! This is a BizzBuzz simulation.  
Please, enter the number of turns:  
15  
1, 2, bizz, 4, buzz, bizz, 7, 8, bizz, buzz, 11, bizz, 13, 14, bizz buzz.
```