

Ohjelmoinnin perusteet

R0027



E

Toistolauseet (for)



Viopen luku 6 loppuun

for - syntaksi

```
for(alkutoimet; toistoehto; lopputoimet) {  
    lauseita;  
}
```

- ❑ **Alkutoimet:** Laskurin määrittely ja alkuarvon asettaminen
- ❑ **Toistoehto:** Laskurin arvon vertaaminen loppuarvoon
- ❑ **Lopputoimet:** Laskurin arvon askeltaminen (yleensä kasvattaminen yhdellä)

for - toiminta

```
for(int i=1; i<=10; i++) {  
    System.out.println(i + " potenssiin 2 on " +i*i);  
}
```

```
System.out.println("Lasketaan nollasta ysiin");
```

```
for(int i=0; i<10; i++) {  
    System.out.println("Ki erros" +i);  
}
```

Sisäkkäiset silmukat

- Silmukoita voidaan tehdä myös sisäkkäin.
- Tällöin ulomman silmukan kullakin yksittäisellä kierroksella suoritetaan sisemmän silmukan kaikki kierrokset läpi

Sisäkkäiset silmukat -esimerkki

- Kysy käyttäjältä kuinka monta riviä halutaan tulostaa
- Kysy käyttäjältä kuinka monta saraketta halutaan tulostaa
- Tulosta ruudulle kertotaulu ohessa esitetyllä tyyllillä

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

Solution: (check file IndentedFor.java in Optima)

```
1 import java.util.Scanner;
2
3 public class IndentedFor {
4     public static void main(String[] args) {
5         int i,j;
6         int rows, columns;
7
8         Scanner inputReader = new Scanner(System.in);
9
10        System.out.println("How many rows?: ");
11        rows = inputReader.nextInt();
12        System.out.println("How many columns?: ");
13        columns = inputReader.nextInt();
14
15        for (i=1; i<=rows; i++) //each iteration of 'i' prints one row
16        {
17            //each iteration of 'j' prints one column in a specific row
18            for (j=1; j<=columns; j++)
19            {
20                //using printf so we can nicely format the output
21                System.out.printf("%4d", (i*j));
22            }
23
24            System.out.println(); //Print a newline to start a new row
25        }
26
27        inputReader.close();
28    }
29 }
30
```

Tulostuksen muotoilu: printf

- ❑ `System.out.printf(muotoilu, muuttuja);`
- ❑ Muotoilun kaava:
`%[flags][width][.precision]`

Esim.

- ❑ `%5d` - kokonaisluku 5 merkin kentässä
- ❑ `%.2f` - kahden desimaalin tarkkuus
- ❑ `(%-16.2f)\n`

Rahan arvo = 66.6666666666666667

- `System.out.printf("%.2f)\n", raha);`
(66,67)

`System.out.printf("%-16.2f)\n", raha);`
(66,67)

`System.out.printf("%16.3f)\n", raha);`
(66,667)

```
System.out.printf("(%016.5f)\n", raha);  
(00000000066,66667)
```

```
System.out.printf("( %+16.3f)\n", raha);  
System.out.printf("( %+16.3f)\n", raha*-1);  
(          +66,667)  
(          -66,667)
```

Debugging code with Eclipse – getting started

- Perspectives
 - Java perspective, the one you have been using
 - Debug perspective, with useful views to debug code, inspect variable values, and step through the code
- Breakpoint
 - suspends the execution of a program at the location where the breakpoint is set
- Step over
 - step over a method call (without entering it) at the currently executing line of code
- Step into
 - step into the next method call at the currently executing line of code
- Resume
 - Continue program execution until the next Breakpoint

Debugging code with Eclipse – some resources

- ❑ Java Debugging with Eclipse – Tutorial
 - <http://www.vogella.com/tutorials/EclipseDebugging/article.html>
- ❑ Eclipse – Debugging a program (tutorial)
 - https://www.tutorialspoint.com/eclipse/eclipse_debugging_program.htm

Exercise

Get the file `Interest.java` from Optima and debug it according to the instructions written in its comments