

Ohjelmointiharjoitus 10

1. SELVITÄ JDK:N DOKUMENTAATIOSTA, MITÄ LUOKALLA JAVA.LANG.CHARACTER VOI TEHDÄ?

Luokka `java.lang.Character` sisältää useita staattisia metodeja, jotka mahdollistavat merkkijonojen käsittelyn. Esimerkiksi:

- `isLetter(char ch)` - palauttaa totuusarvon, joka kertoo, onko annettu merkki kirjain vai ei.
- `isDigit(char ch)` - palauttaa totuusarvon, joka kertoo, onko annettu merkki numero vai ei.
- `toUpperCase(char ch)` - palauttaa annetun merkin isoilla kirjaimilla kirjoitettuna.
- `toLowerCase(char ch)` - palauttaa annetun merkin pienillä kirjaimilla kirjoitettuna.

2. ENTÄ MITÄ TEKEE JAVA.LANG.MATH-LUOKAN METODI ABS?

`java.lang.Math`-luokan `abs`-metodi palauttaa annetun numeron absoluuttisen arvon (eli sen etäisyyden nolasta). Esimerkiksi:

```
int x = -5;
```

```
int absX = Math.abs(x); // absX:n arvo on 5
```

**3. LAADI OHJELMA, JOSSA KÄYTTÄJÄN SYÖTTÄMÄ
PITUUS (CM) ILMOITETAAN JALKOINA (FT) JA
TUUMINA (IN). TOTEUTA MUUNNOS MYÖS
TOISINPÄIN. TEE AINAKIN MUUNNOKSILLE OMAT
ALIOHJELMAT.**

Pituusmuunnin.java ×

```
1 package kotitehtävä10;
2
3 import java.util.Scanner;
4
5 public class Pituusmuunnin {
6     public static void main(String[] args) {
7         Scanner lukija = new Scanner(System.in);
8
9         System.out.print("Syötä pituus (cm): ");
10        int pituusCm = lukija.nextInt();
11
12        // muunnetaan syötetty pituus jalka- ja tuuma-arvoiksi
13        double pituusFt = muunnaCmFt(pituusCm);
14        double pituusIn = muunnaCmIn(pituusCm);
15
16        // tulostetaan syötetty pituus ja muunnokset
17        System.out.println(pituusCm + " cm = " + pituusFt + " ft " + pituusIn + " in");
18
19        // muunnetaan jalka- ja tuuma-arvot takaisin senttimetreiksi
20        double pituusCm2 = muunnaFtIn(pituusFt, pituusIn);
21
22        // tulostetaan muunnetut arvot
23        System.out.println(pituusCm2 + " cm");
24    }
25
26    // muuntaa senttimetreit jalkoiksi pyöristäen
27    public static double muunnaCmFt(int cm) {
28        double ft = cm / 30.48;
29        return Math.floor(ft);
30    }
31
32    // muuntaa senttimetreit tuumiksi
33    // % 12:lla varmistetaan, että lopputulos on kokonaisluku
34    public static double muunnaCmIn(int cm) {
35        double in = (cm / 2.54) % 12;
36        return in;
37    }
38
39    // muuntaa jalka- ja tuuma-arvot senttimetreiksi
40    public static double muunnaFtIn(double ft, double in) {
41        double cm = (ft * 30.48) + (in * 2.54);
42    }
43 }
```

4. MITÄ JAVAN DOKUMENTAATIO SANOO OMEN OHJELMIEN KOMMENTOINNISTA LUOKKIEN, METODIEN, MUUTTUIJEN OSALTA?

Javan dokumentaatio suosittelee, että ohjelmakoodi kommentoidaan huolellisesti kahdentyypisillä kommenteilla: peruskommenteilla (kaksoisviiva //) ja javadoc-kommenteilla (kahdella tähdellä /**). Javadoc-kommentit ovat erityistyyppisiä kommentteja, jotka sijoitetaan luokan, metodin tai muuttujan yläpuolelle ja niitä käytetään automaattisesti luotavien dokumentaatioiden luomiseen. Selkeiden ja ymmärrettävien kommenttien käyttö auttaa muita kehittäjiä ymmärtämään koodin toimintaa ja tarkoitusta sekä helpottaa koodin ylläpitoa ja muokkaamista.

- Luokan tarkoitus ja käyttötarkoitus
- Metodin parametrit ja paluuarvo
- Muuttujan käyttötarkoitus ja ominaisuudet

Yleisesti ottaen, Javan dokumentaatio suosittelee, että ohjelmakoodi kommentoidaan selkeästi ja ymmärrettävästi, jotta muut kehittäjät voivat helposti ymmärtää koodin toimintaa ja tarkoitusta. Kommenttien avulla on helpompi ylläpitää ja muokata koodia, mikä voi säästää aikaa ja vaivaa pitkällä aikavälillä.

5. TOTEUTA EDELLINEN TEHTÄVÄSSÄ 3.

sama koodi kun 3 tehtävässä mutta kommentit vaihdettu vastaamana javadocia

Pituusmuunnin.java X

```
1 package kotitehtävä10;
2
3 import java.util.Scanner;
4
5 import java.util.Scanner;
6
7 public class Pituusmuunnin {
8
9     public static void main(String[] args) {
10         Scanner lukija = new Scanner(System.in);
11         System.out.print("Anna pituus senttimetreinä: ");
12         double pituusCm = lukija.nextDouble();
13         double[] jalatTuumat = cmJalkatuumat(pituusCm);
14         double jalkoja = jalatTuumat[0];
15         double tuumia = jalatTuumat[1];
16         System.out.println(pituusCm + " cm = " + jalkoja + " jalkaa ja " + tuumia + " tuumaa");
17         double sentteina = jalkatuumatCm(jalkoja, tuumia);
18         System.out.println(jalkoja + " ft " + tuumia + " in");
19     }
20
21     /**
22      * Muuntaa senttimetrejä jalkoiksi ja tuumiksi.
23      * @param cm Pituus senttimetreinä.
24      * @return Taulukko, jossa ensimmäinen alkio on jalkojen määrä ja toinen tuumien määrä.
25      */
26     public static double[] cmJalkatuumat(double cm) {
27         double[] tulos = new double[2];
28         double jalkoja = cm / 30.48;
29         double kokonaisiaJalkoja = Math.floor(jalkoja);
30         double tuumia = (jalkoja - kokonaisiaJalkoja) * 12;
31         tulos[0] = kokonaisiaJalkoja;
32         tulos[1] = tuumia;
33         return tulos;
34     }
35
36     /**
37      * Muuntaa jalkojen ja tuumien määrän senttimetreiksi.
38      * @param ft Jalkojen määrä.
39      * @param in Tuumien määrä.
40      * @return Pituus senttimetreinä.
41      */
42     public static double jalkatuumatCm(double ft, double in) {
43         return ft * 30.48 + in * 2.54;
44     }
45 }
```

6. LUO JAVADOC-DOKUMENTTI

kuva siitä kuntein javadocin ja myös kuva javadocista



P... X

Pituusmuunnin.java X

```

1 package KT10;
2
3 import java.util.Scanner;
4
5 public class Pituusmuunnin {
6
7     public static void main(String[] args)
8     {
9         Scanner lukija = new Scanner(System.in);
10        System.out.print("Anna pituus senttimetreinä: ");
11        double pituusCm = lukija.nextDouble();
12        double[] jalatTuumat = cmJalkatuumat(pituusCm);
13        double jalvoja = jalatTuumat[0];
14        double tuumia = jalatTuumat[1];
15        System.out.println(pituusCm + " cm on " + jalvoja + " jalkaa ja " + tuumia + " tuumaa.");
16        System.out.println(jalvoja + " ft on " + tuumia + " in");
17    }
18
19    /**
20     * Muuntaa senttimetreit jalkoiksi ja tuumiksi
21     * @param cm Pituus senttimetreinä.
22     * @return Taulukko, jossa ensimmäinen on jalat ja toinen tuumia
23     */
24    public static double[] cmJalkatuumat(double cm) {
25        double[] tulos = new double[2];
26        double jalvoja = cm / 30.48;
27        double kokonaisiaJalvoja = Math.floor(jalvoja);
28        double tuumia = (jalvoja - kokonaisiaJalvoja) * 12;
29        tulos[0] = kokonaisiaJalvoja;
30        tulos[1] = tuumia;
31        return tulos;
32    }
33
34    /**
35     * Muuntaa jalvojen ja tuumien määrän senttimetreiksi
36     * @param ft Jalvojen määrä.
37     * @param in Tuumien määrä.
38     * @return Pituus senttimetreinä.
39     */
40    public static double jalkatuumatCm(double ft, double in) {
41        double cm = ft * 30.48 + in * 2.54;
42        return cm;
43    }
44 }
45
46

```

Console

<terminated>

Loading...

Constants...

Build...

Standard...

Build...

Generated...

C:\Us...

public...

C:\Us...

public...

C:\Us...

public...

C:\Us...

P...

Gener...

Gener...

Gener...

C:\Us...

* @a...

^

Gener...

Gener...

Gener...

C:\Us...

Gener...

Gener...

Gener...

Gener...

Build...

Gener...

Gener...

Gener...

Gener...

Gener...

Gener...

Gener...

Gener...

Gener...

Gener...

Gener...

4 war...

7. SELVITÄ, MITEN JAVA-OHJELMAN VOI AJAA KOMENTORIVILTÄ.

Java-ohjelman ajaminen komentoriviltä on yksinkertaista. Voit seurata seuraavia vaihteita ohjelman ajamiseksi komentoriviltä:

1. Avaa komentokehote tai terminäli.
2. Siirry komentokehoteessa tai terminaalissa Java-projektin juurihakemistoon.
3. Käännä Java-ohjelma komennolla `javac` ja anna sen jälkeen argumentiksi `.java`-tiedoston polku. Esimerkiksi, jos ohjelma on tiedostossa `MyProgram.java`, komento olisi `javac MyProgram.java`.
4. Kun kompilaatio on valmis, voit ajaa ohjelman komennolla `java` ja anna sen jälkeen argumentiksi ohjelman luokan nimen, joka sisältää `main`-metodin. Esimerkiksi, jos ohjelman `main`-metodi on luokassa `MyProgram`, komento olisi `java MyProgram`.
5. Ohjelma suoritetaan ja tulostaa mahdollisen syötteen, laskelman tai muun ohjelman toiminnan komentoriville.

On tärkeää huomata, että ohjelman ajamiseen tarvitaan JDK:n (Java Development Kit) asennus. Voit asentaa JDK:n Oracle-sivustolta löytyvän ohjelmiston avulla.

7. SELVITÄ, MITEN JAVA-OHJELMAN VOI AJAA KOMENTORIVILTÄ.

Java-ohjelman ajaminen komentoriviltä on yksinkertaista. Voit seurata seuraavia vaiheita ohjelman ajamiseksi komentoriviltä:

1. Avaa komentokehote tai terminaali.
2. Siirry komentokehoteessa tai terminaalissa Java-projektin juurihakemistoon.
3. Käännä Java-ohjelma komennolla `javac` ja anna sen jälkeen argumentiksi `.java`-tiedoston polku. Esimerkiksi, jos ohjelma on tiedostossa `MyProgram.java`, komento olisi `javac MyProgram.java`.
4. Kun kompilaatio on valmis, voit ajaa ohjelman komennolla `java` ja anna sen jälkeen argumentiksi ohjelman luokan nimen, joka sisältää `main`-metodin. Esimerkiksi, jos ohjelman `main`-metodi on luokassa `MyProgram`, komento olisi `java MyProgram`.
5. Ohjelma suoritetaan ja tulostaa mahdollisen syötteen, laskelman tai muun ohjelman toiminnan komentoriville.

On tärkeää huomata, että ohjelman ajamiseen tarvitaan JDK:n (Java Development Kit) asennus. Voit asentaa JDK:n Oracle-sivustolta löytyvän ohjelmiston avulla.

8. NÄYTÄ EDELLINEN KÄYTÄNNÖSSÄ OHJELMALLA, JOKA LASKEE KAKSI LUKUA YHTEEN. TEE VERSIO, JOSSA LUVUT LUODAAN OHJELMAN SISÄLLÄ SEKÄ VERSIO, JOSSA LUVUT ANNETAAN KOMENTORIVILLÄ.

```
Summa.java ×
1 package kotitehtävät10;
2
3 public class Summa {
4     public static void main(String[] args) {
5         int luku1 = Integer.parseInt(args[0]);
6         int luku2 = Integer.parseInt(args[1]);
7         int summa = luku1 + luku2;
8         System.out.println("Lukujen " + luku1 + " ja
9     }
10 }
11
12
```

9. ALLA ON KUVAKAAPPAUS LUOKASTA TEST. MITÄ OHJELMA TEKEE? POISTA KOODISTA BREAKOHJELMAN TOIMINTAA MUUTTAMATTA.

```
Test.java X
1 package kotitehtävät10;
2
3 public class Test {
4     public static void main(String[] args) {
5         int[] numbers = {10, 20, 30, 40, 50};
6
7         for(int x: numbers) {
8             if (x== 30) {
9                 break;
10            }
11            System.out.println(x);
12            System.out.println("\n");
13        }
14    }
15
16 }
```

Tulostaa taulukon numbers alkioita kunnes saavuttaa ensimmäisen luvun, joka on yhtä suuri kuin 30. Tämän jälkeen se lopettaa for-silmukan suorittamisen break-käskyllä.

Jos break poistetaan ohjelmasta muuttamatta sen toimintaa se voidaan poistaa ja jättää muuten koodi ennalleen tuolloin for-silmukka suoritetaan loppuun asti ja tulostetaan kaikki taulukon numbers alkiot

Alla on koodi missä koodi ilman break-toimintoa

```
Test.java ×
1 package kotitehtävät10;
2
3 public class Test {
4     public static void main(String[] args) {
5         int[] numbers = {10, 20, 30, 40, 50};
6
7         for(int x: numbers) {
8             System.out.println(x);
9             System.out.println("\n");
10        }
11    }
12 }
13
```

**9. ALLA ON KUVAKAAPPAUS LUOKASTA TEST.
MITÄ OHJELMA TEKEE? POISTA KOODISTA
BREAKOHJELMAN TOIMINTAA MUUTTAMATTA.**

```

package kotitehtävät10;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Double number, sum = 0.0;
        // Create an object Scanner
        Scanner input = new Scanner(System.in);

        for (int i = 1; i < 6; ++i) {
            System.out.println("Enter number " + i +
                //takes input from the user
                number = input.nextDouble();
                // if number negative
                // continue statement is executed
                if (number <= 0.0) {
                    continue;
                }
                sum += number;
            }
            System.out.println("Sum =" + sum);
            input.close();
        }
    }
}

```

Pyytää käyttäjältä syöttämään viisi lukua jokainen syötetään erikseen, jos syötetty luku on negatiivinen se ohitetaan continue komennon avulla ja siirtyy seuraavaa annettavaan lukuun. Muussa tapauksessa ohjelma lisää syötetyn luvun summaan ja lopuksi ohjelma tulostaa summattujen lukujen tuloksen

Mikäli poistaa continue-komennon, voisi sen tilalle käyttää if-lauseen sisälle toista if-lausetta, joka tarkistaa

onko syötetty luku negatiivinen. Tällöin ohjelma laskee myös negatiiviset luvut mukaan summaan. Seuraava koodi esittelee muutetun ohjelman:

```
Main.java ×
1 package kotitehtävät10;
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Double number, sum = 0.0;
8         // Create an object Scanner
9         Scanner input = new Scanner(System.in);
10
11         for (int i = 1; i < 6; ++i) {
12             System.out.println("Enter number " + i);
13             //takes input from the user
14             number = input.nextDouble();
15             // if number negative
16             // continue statement is executed
17             if (number < 0.0) {
18                 System.out.println("Negative number");
19             }
20             sum += number;
21         }
22         System.out.println("Sum =" + sum);
23         input.close();
24     }
25 }
26
```

Ohjelma ottaa muutoksien negatiiviset luvut mukaan summaan mutta tulostaa samalla ilmoituksen että negatiivinen luku lisätty summaan

