

Luento 2 Kotitehtävä

Maksimipistemäärä 10 pistettä

Tee tehtäviä niin, että saat kerättyä yhteensä 10 pistettä.

Tehtävä 1 (3 p)

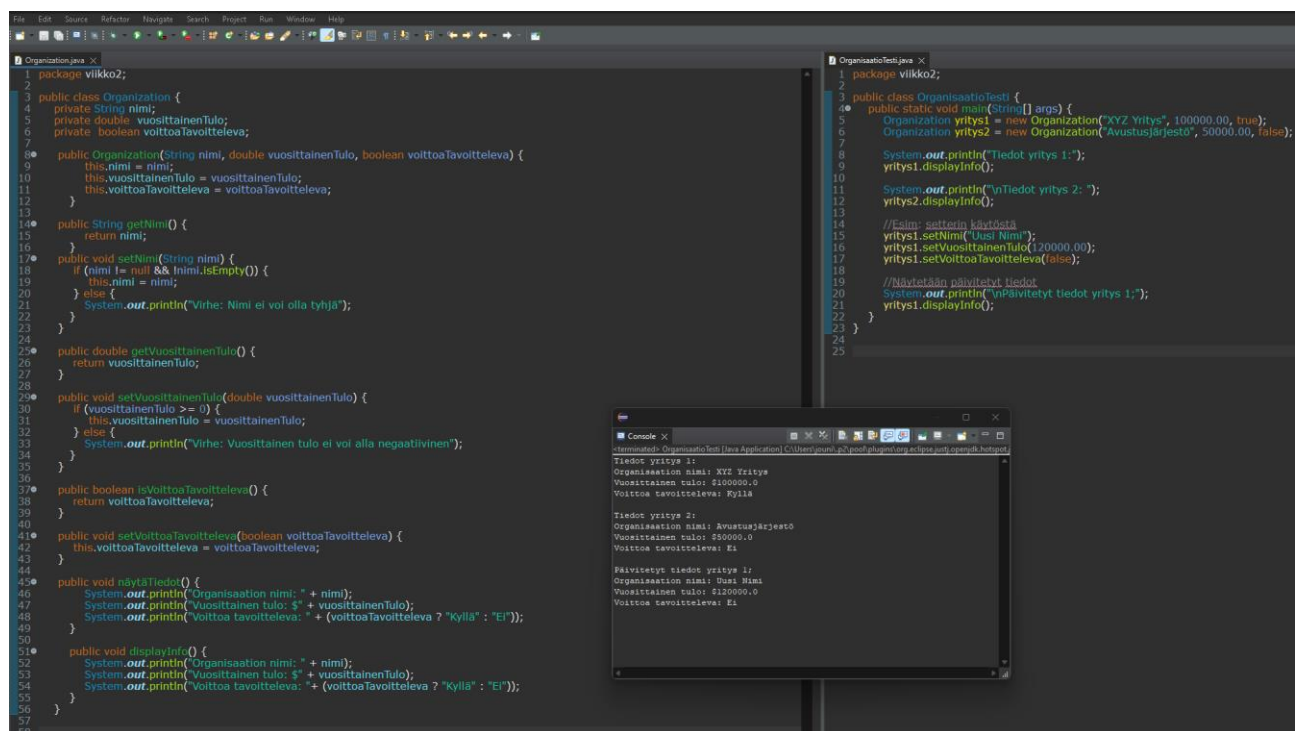
Jatketaan kirjan Java For Dummies, luvun 7 tehtävien kanssa.

Tämä on kirjan Try It Out tehtävä sivulla 190.

Käytä edellisen kotitehtävän Organization and FoodProduct luokkia.

- Merkitse luokkien attribuutit (ominaisuudet) privateiksi (1 p)
- Ota getterit ja setterit käyttöön. (1 p)
- Setter metodeissa lisää tarvittavat tarkastukset, joilla varmistat, että String arvot eivät ole tyhjiä ja numeeriset arvot eivät ole negatiivisia. (1 p)

Organization set/get 1



```
1 package viikko2;
2
3 public class Organization {
4     private String nimi;
5     private double vuosittainenTulo;
6     private boolean voittoaTavoitteleva;
7
8     public Organization(String nimi, double vuosittainenTulo, boolean voittoaTavoitteleva) {
9         this.nimi = nimi;
10        this.vuosittainenTulo = vuosittainenTulo;
11        this.voittoaTavoitteleva = voittoaTavoitteleva;
12    }
13
14    public String getNimi() {
15        return nimi;
16    }
17    public void setNimi(String nimi) {
18        if (nimi != null && !nimi.isEmpty()) {
19            this.nimi = nimi;
20        } else {
21            System.out.println("Virhe: Nimi ei voi olla tyhjä");
22        }
23    }
24
25    public double getVuosittainenTulo() {
26        return vuosittainenTulo;
27    }
28
29    public void setVuosittainenTulo(double vuosittainenTulo) {
30        if (vuosittainenTulo >= 0) {
31            this.vuosittainenTulo = vuosittainenTulo;
32        } else {
33            System.out.println("Virhe: Vuosittainen tulo ei voi olla negatiivinen");
34        }
35    }
36
37    public boolean isVoittoaTavoitteleva() {
38        return voittoaTavoitteleva;
39    }
40
41    public void setVoittoaTavoitteleva(boolean voittoaTavoitteleva) {
42        this.voittoaTavoitteleva = voittoaTavoitteleva;
43    }
44
45    public void näytäTiedot() {
46        System.out.println("Organisaation nimi: " + nimi);
47        System.out.println("Vuosittainen tulo: $" + vuosittainenTulo);
48        System.out.println("Voittoa tavoitteleva: " + (voittoaTavoitteleva ? "Kyllä" : "Ei"));
49    }
50
51    public void displayInfo() {
52        System.out.println("Organisaation nimi: " + nimi);
53        System.out.println("Vuosittainen tulo: $" + vuosittainenTulo);
54        System.out.println("Voittoa tavoitteleva: " + (voittoaTavoitteleva ? "Kyllä" : "Ei"));
55    }
56 }
57
58
```

```
1 package viikko2;
2
3 public class OrganisaatioTesti {
4     public static void main(String[] args) {
5         Organization yritys1 = new Organization("XYZ Yritys", 100000.00, true);
6         Organization yritys2 = new Organization("Avustusjärjestö", 50000.00, false);
7
8         System.out.println("Tiedot yritys 1:");
9         yritys1.displayInfo();
10
11        System.out.println("\nTiedot yritys 2: ");
12        yritys2.displayInfo();
13
14        //Esim: setterin käyttöä
15        yritys1.setNimi("Uusi Nimi");
16        yritys1.setVuosittainenTulo(120000.00);
17        yritys1.setVoittoaTavoitteleva(false);
18
19        //Näytetään päivitetty tiedot
20        System.out.println("\nPäivitetyt tiedot yritys 1:");
21        yritys1.displayInfo();
22    }
23 }
24
25
```

```
Tiedot yritys 1:
Organisaation nimi: XYZ Yritys
Vuosittainen tulo: $100000.0
Voittoa tavoitteleva: Kyllä

Tiedot yritys 2:
Organisaation nimi: Avustusjärjestö
Vuosittainen tulo: $50000.0
Voittoa tavoitteleva: Ei

Päivitetyt tiedot yritys 1:
Organisaation nimi: Uusi Nimi
Vuosittainen tulo: $120000.0
Voittoa tavoitteleva: Ei
```

FoodProduct set/get

```
1 package vikko2;
2
3 public class FoodProduct {
4     public String tyyppi;
5     public double paino;
6     public double hinta;
7     public int annostenMaara;
8     public double kaloreitaAnnostaKohti;
9
10    public FoodProduct(String tyyppi, double paino, double hinta, int annostenMaara, double kaloreitaAnnostaKohti) {
11        this.tyyppi = tyyppi;
12        this.paino = paino;
13        this.hinta = hinta;
14        this.annostenMaara = annostenMaara;
15        this.kaloreitaAnnostaKohti = kaloreitaAnnostaKohti;
16    }
17
18    public String getTyyppi() {
19        return tyyppi;
20    }
21
22    public void setTyyppi(String tyyppi) {
23        if (tyyppi != null && tyyppi.isEmpty()) {
24            this.tyyppi = tyyppi;
25        } else {
26            System.out.println("Virhe: Tyyppi ei voi olla tyhjä");
27        }
28    }
29
30    public double getPaino() {
31        return paino;
32    }
33
34    public void setPaino(double paino) {
35        if (paino >= 0) {
36            this.paino = paino;
37        } else {
38            System.out.println("Virhe: Paino ei voi olla tyhjä");
39        }
40    }
41
42    public double getHinta() {
43        return hinta;
44    }
45
46    public void setHinta(double hinta) {
47        if (hinta >= 0) {
48            this.hinta = hinta;
49        } else {
50            System.out.println("Virhe: Hinta ei voi olla negatiivinen");
51        }
52    }
53 }
```

```
1 package vikko2;
2
3 public class FoodProductTest {
4     public static void main(String[] args) {
5         FoodProduct hedelmia = new FoodProduct("Hedelmät", 100, 2.50, 5, 100);
6         FoodProduct leivonnaiset = new FoodProduct("Leivonnaiset", 200, 5.00, 10, 250);
7
8         System.out.println("Tiedot hedelmistä");
9         hedelmia.displayInfo();
10
11         System.out.println("\nTiedot leivonnaisista");
12         leivonnaiset.displayInfo();
13
14         // Kutsutaan set-metodeja päivittämään tietoa
15         hedelmia.setTyyppi("Uudet Hedelmät");
16         hedelmia.setPaino(120);
17         hedelmia.setHinta(3.00);
18         hedelmia.setAnnostenMaara(6);
19         hedelmia.setKaloreitaAnnostaKohti(120);
20
21         System.out.println("\nPäivitetty tiedot hedelmistä");
22         hedelmia.displayInfo();
23     }
24 }
```

```
Tiedot tyyppi 1:
Organisation nimi: KTD Terveystieteiden tutkimuskeskus
Voittoa tavoittelematon: Kyllä
Tiedot tyyppi 2:
Organisation nimi: Arvostusjärjestö
Voittoa tavoittelematon: Ei
Päivitetty tiedot tyyppi 1:
Organisation nimi: Uudet Hedelmät
Voittoa tavoittelematon: Ei
Päivitetty tiedot tyyppi 2:
Organisation nimi: Uudet Hedelmät
Voittoa tavoittelematon: Ei
```

```
59 public void setAnnostenMaara(int annostenMaara) {
60     if (annostenMaara >= 0) {
61         this.annostenMaara = annostenMaara;
62     } else {
63         System.out.println("Virhe: Annosten määrä ei voi olla negatiivinen");
64     }
65 }
66
67 public double getKaloreitaAnnostaKohti() {
68     return kaloreitaAnnostaKohti;
69 }
70
71 public void setKaloreitaAnnostaKohti(double kaloreitaAnnostaKohti) {
72     if (kaloreitaAnnostaKohti >= 0) {
73         this.kaloreitaAnnostaKohti = kaloreitaAnnostaKohti;
74     } else {
75         System.out.println("Virhe: Kalorimäärä ei voi olla negatiivinen");
76     }
77 }
78
79 public void näytäTiedot() {
80     System.out.println("Tuotetiedot");
81     System.out.println("Tyyppi: " + tyyppi);
82     System.out.println("Paino: " + paino + " grammaa");
83     System.out.println("Hinta: $" + hinta);
84     System.out.println("Annosten määrä: " + annostenMaara);
85     System.out.println("Kaloreita annosta kohti: " + kaloreitaAnnostaKohti);
86 }
87
88 public void displayInfo() {
89     System.out.println("Tuotetiedot");
90     System.out.println("Tyyppi: " + tyyppi);
91     System.out.println("Paino: " + paino + " grammaa");
92     System.out.println("Hinta: $" + hinta);
93     System.out.println("Annosten määrä: " + annostenMaara);
94     System.out.println("Kaloreita annosta kohti: " + kaloreitaAnnostaKohti);
95 }
96 }
97 }
```

Tehtävä 2 (1 p)

Selitä jommankumman esimerkin koodi yksityiskohtaisesti, kerro mitä siinä tapahtuu ja mihin muuttujien arvot milloinkin ”liikkuvat”.

Tässä on selitys FoodProduct-luokan koodista, mukaan lukien, mitä siinä tapahtuu ja milloin muuttujien arvot päivitetään:

1. Luodaan FoodProduct-luokka, joka kuvaa elintarviketuotetta ja sisältää viisi yksityistä attribuuttia: tyyppi, paino, hinta, annostenMaara ja kaloreitaAnnostaKohti.
2. Luodaan FoodProduct-luokan konstruktori, joka ottaa vastaan nämä attribuutit ja asettaa ne luokan sisäisiin muuttujiin. Konstruktori varmistaa, että attribuuttien arvot ovat kelvollisia, koska se käyttää settereitä niiden asettamiseen.
3. setTyyppi, setPaino, setHinta, setAnnostenMaara ja setKaloreitaAnnostaKohti ovat setter-metodeja, jotka tarkistavat, että niiden saamat syötteet ovat kelvollisia. Jos ne ovat kelvollisia, ne päivittävät vastaavan attribuutin arvon. Muussa tapauksessa ne näyttävät virheviestin.
4. getTyyppi, getPaino, getHinta, getAnnostenMaara ja getKaloreitaAnnostaKohti ovat getter-metodeja, jotka palauttavat attribuutin arvon.
5. näytTiedot-metodi tulostaa elintarviketuotteen tiedot konsoliin, mukaan lukien tyyppi, paino, hinta, annosten määrä ja kalorit annosta kohti.
6. displayInfo-metodi on yksinkertaisesti kutsu näytTiedot-metodiin. Se on lisätty vain luettavuuden parantamiseksi.

Kokonaisuudessaan tämä koodi mahdollistaa elintarviketuotteiden luomisen, niiden tietojen päivittämisen turvallisesti settereiden avulla ja tietojen näyttämisen käyttämällä näytTiedot- tai displayInfo-metodeja. Setterit tarkistavat, että syötteet ovat kelvollisia ennen attribuuttien päivittämistä, mikä tekee luokasta turvallisen ja ylläpitävän.

Tehtävä 3 (4 p)

Luo luokka Ajastin. Ajastimella on oliomuuttuja private int aika, parametrin konstruktori (asettaa muuttujan aika alkuarvoksi 0), sekä seuraavat neljä metodia:

1. Metodi public void lisää() kasvattaa oliomuuttujan aika arvoa viidellä. Arvoa ei kasvateta yli 30:n.
2. Metodi public void vähenna() vähentää oliomuuttujan aika arvoa viidellä. Arvoa ei vähennetä negatiiviseksi.
3. Metodi public int aika() palauttaa oliomuuttujan aika arvon.
4. Metodi public boolean loppu() palauttaa arvon true, mikäli oliomuuttujan arvo on 30, muutoin palautetaan false.

Luo Main luokka erilliseen tiedostoon, jossa on seuraava koodi:

```
import java.util.Scanner;
```

```
public class Main {
```

```

public static void main(String[] args) {

    Aika time = new Aika();

    while(!time.loppe()) {

        System.out.println("Aika ei ole vielä loppu! Aika: " +
            time.aika());

        time.lisaa();

    }

    System.out.println("Aika loppui! Aika: " + time.aika());

    aika.vahenna();

    System.out.println("Aika ei ole loppu! Aika: " + time.aika());

}

}

```

Tämän koodin avulla pystyt testaamaan, toimivatko luomasi metodit.

Kopioi vastaukseesi myös tulosteet.

Tehtävä 4 (Maks. 6 p)

Lue artikkeli sivulta <https://dzone.com/articles/java-getter-and-setter-basics-common-mistakes-and>

Vastaa seuraaviin kysymyksiin käyttäen artikkelia lähteenä. Lue kohtaan 6 asti. Kohta 7 saattaa tuntua jo vähän hankalalta tässä vaiheessa.

1. Mitä ovat getterit ja setterit? (1 p)
ovat kaksi tavanomaista menetelmää, joita käytetään muuttujan arvon hakemiseen ja päivittämiseen.
2. Miksi settereitä ja gettereitä tarvitaan? (1 p)
Näitä käyttämällä voidaan hallita, miten tärkeitä muuttujia käytetään ja päivitetään tarkasti, mikä auttaa varmistamaan turvallisen ja hallitun pääsyn muuttujiin ohjelman rakenteessa
3. Mitä ovat tyypillisimmät sudenkuopat? (1 p)
voivat aiheuttaa ongelmia, kuten liiallista monimutkaisuutta, jos niihin lisätään monimutkaista logiikkaa. Niiden väärinkäyttö tai puutteellinen suojaus voi myös johtaa koodin turvallisuusongelmiin ja odottamattomaan käyttäytymiseen. Siksi niitä tulisi käyttää huolellisesti ja yksinkertaisesti, ja muuttujien tulisi olla asianmukaisesti yksityisiä varmistaaksesi koodin eheyden ja ylläpidettävyyden.
4. Miten niiltä vältetään? (1 p)
Sudenkuopilta gettereiden ja settereiden käytössä Javassa voi välttää noudattamalla muutamia periaatteita. Käytä yksinkertaisia gettereitä ja settereitä ilman monimutkaista liiketoimintalogiikkaa, merkitse muuttujat yksityisiksi (private), anna selkeät ja kuvaavat nimet metodeille, tarjoa vain tarvittavat getterit ja setterit, testaa ne huolellisesti, ja dokumentoi niiden käyttö ja rajoitukset. Näillä toimenpiteillä varmistat koodin eheyden ja ylläpidettävyyden.
5. Testaa kohtien 1-6 esimerkkejä omilla muuttujilla (1 p)

Tässä esimerkki

```
public class Auto {
    private String merkki;
    private int vuosimalli;

    // Getter autojen merkille
    public String getMerkki() {
        return this.merkki;
    }

    // Setter autojen merkille
    public void setMerkki(String merkki) {
        this.merkki = merkki;
    }

    // Getter autojen vuosimallille
    public int getVuosimalli() {
        return this.vuosimalli;
    }

    // Setter autojen vuosimallille
    public void setVuosimalli(int vuosimalli) {
        this.vuosimalli = vuosimalli;
    }

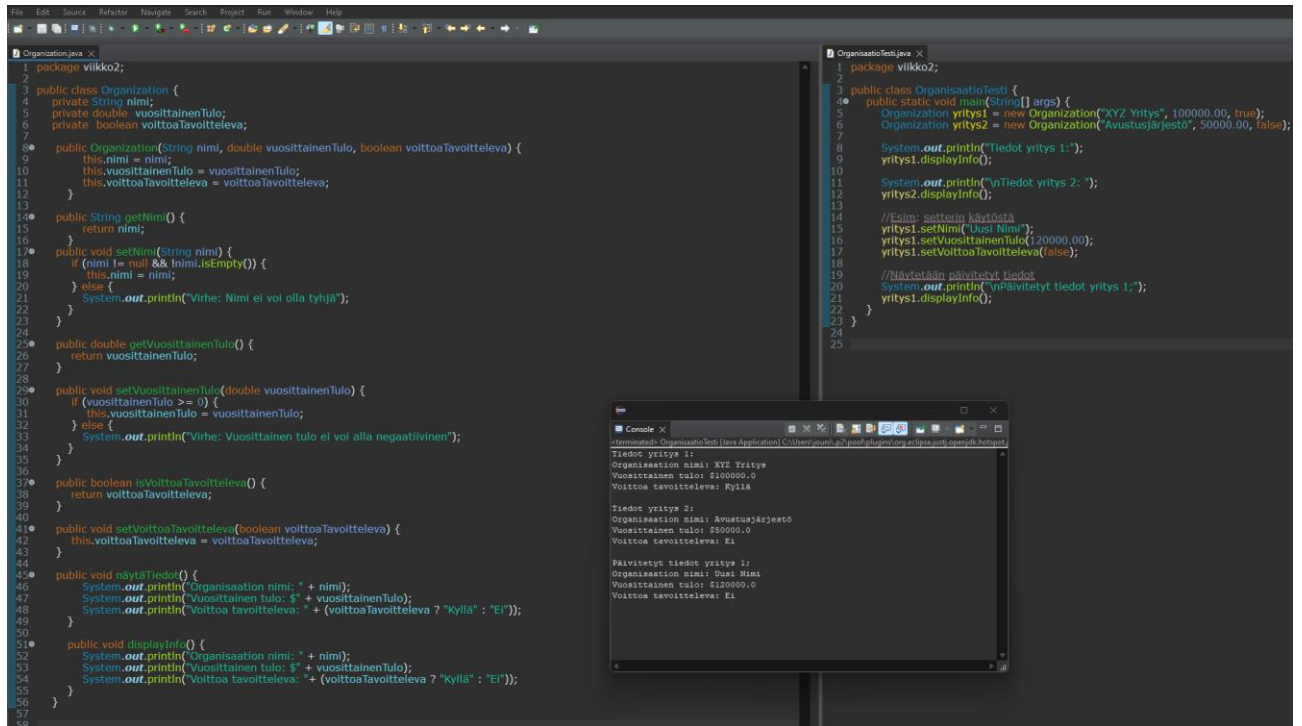
    public static void main(String[] args) {
        Auto auto = new Auto();

        // Asetetaan auton merkki
        auto.setMerkki("Toyota");

        // Asetetaan auton vuosimalli
        auto.setVuosimalli(2020);

        // Tulostetaan auton tiedot
        System.out.println("Auton merkki: " + auto.getMerkki());
        System.out.println("Auton vuosimalli: " + auto.getVuosimalli());
    }
}
```

6. Sovella sivuston koodiesimerkkejä Organization tai FoodProduct -esimerkkiin (1 op)



```
1 package viikko2;
2
3 public class Organization {
4     private String nimi;
5     private double vuosittainenTulo;
6     private boolean voittoaTavoitteleva;
7
8     public Organization(String nimi, double vuosittainenTulo, boolean voittoaTavoitteleva) {
9         this.nimi = nimi;
10        this.vuosittainenTulo = vuosittainenTulo;
11        this.voittoaTavoitteleva = voittoaTavoitteleva;
12    }
13
14    public String getNimi() {
15        return nimi;
16    }
17    public void setNimi(String nimi) {
18        if (nimi != null && !nimi.isEmpty()) {
19            this.nimi = nimi;
20        } else {
21            System.out.println("Virhe: Nimi ei voi olla tyhjä");
22        }
23    }
24
25    public double getVuosittainenTulo() {
26        return vuosittainenTulo;
27    }
28
29    public void setVuosittainenTulo(double vuosittainenTulo) {
30        if (vuosittainenTulo >= 0) {
31            this.vuosittainenTulo = vuosittainenTulo;
32        } else {
33            System.out.println("Virhe: Vuosittainen tulo ei voi olla negatiivinen");
34        }
35    }
36
37    public boolean isVoittoaTavoitteleva() {
38        return voittoaTavoitteleva;
39    }
40
41    public void setVoittoaTavoitteleva(boolean voittoaTavoitteleva) {
42        this.voittoaTavoitteleva = voittoaTavoitteleva;
43    }
44
45    public void näytäTiedot() {
46        System.out.println("Organisaation nimi: " + nimi);
47        System.out.println("Vuosittainen tulo: $" + vuosittainenTulo);
48        System.out.println("Voittoa tavoitteleva: " + (voittoaTavoitteleva ? "Kyllä" : "Ei"));
49    }
50
51    public void displayInfo() {
52        System.out.println("Organisaation nimi: " + nimi);
53        System.out.println("Vuosittainen tulo: $" + vuosittainenTulo);
54        System.out.println("Voittoa tavoitteleva: " + (voittoaTavoitteleva ? "Kyllä" : "Ei"));
55    }
56 }
57
58
```

```
1 package viikko2;
2
3 public class OrganisaatioTesti {
4     public static void main(String[] args) {
5         Organization yritys1 = new Organization("XYZ Yritys", 100000.00, true);
6         Organization yritys2 = new Organization("Avustusjärjestö", 50000.00, false);
7
8         System.out.println("Tiedot yritys 1:");
9         yritys1.displayInfo();
10
11         System.out.println("\nTiedot yritys 2:");
12         yritys2.displayInfo();
13
14         //Nimi: saatoin käyttää
15         yritys1.setNimi("Uusi Nimi");
16         yritys1.setVuosittainenTulo(120000.00);
17         yritys1.setVoittoaTavoitteleva(false);
18
19         //Näytetään päivitetty tiedot
20         System.out.println("\nPäivitetty tiedot yritys 1:");
21         yritys1.displayInfo();
22     }
23 }
24
25
```

```

Tiedot yritys 1:
Organisaation nimi: XYZ Yritys
Vuosittainen tulo: $100000.0
Voittoa tavoitteleva: Kyllä

Tiedot yritys 2:
Organisaation nimi: Avustusjärjestö
Vuosittainen tulo: $50000.0
Voittoa tavoitteleva: Ei

Päivitetty tiedot yritys 1:
Organisaation nimi: Uusi Nimi
Vuosittainen tulo: $120000.0
Voittoa tavoitteleva: Ei

```