

Folge 2.1

Die Klasse Waage

Version vom 12.10.2025

Web-Version

Wiederholung

Einführung in die Objektorientierte Programmierung (OOP)

2.1.1 Die erste Version der Klasse Waage

Deklaration der
Instanzvariablen

Der **Konstruktor**

Setter-Methoden
für Instanzvariablen

Initialisierung der
Instanzvariablen

```
public class Waage
{
    private double gewicht, groesse;

    public Waage()
    {
    }

    public void setGewicht(double gewicht)
    {
        this.gewicht = gewicht;
    }

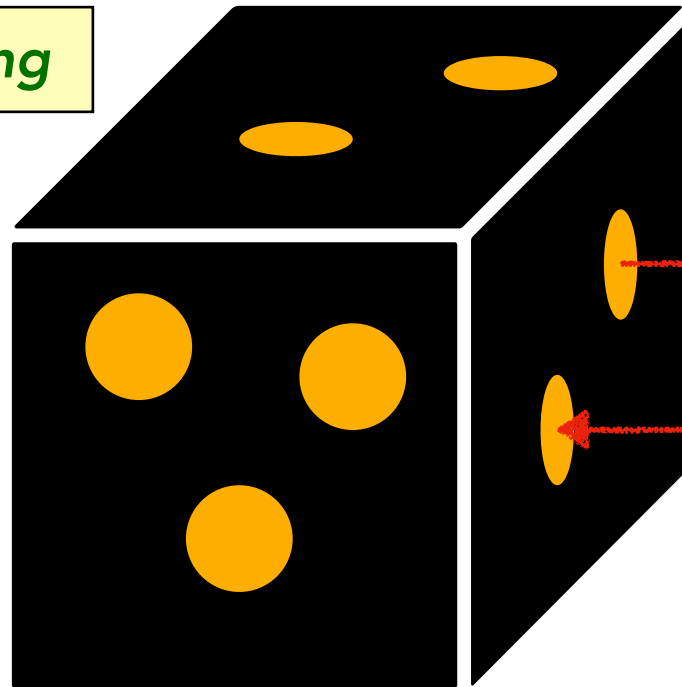
    public void setGroesse(double groesse)
    {
        this.groesse = groesse;
    }
}
```

Einführung in die Objektorientierte Programmierung (OOP)

2.1.2 Datenkapseln und Information hiding

`private`-Instanzvariablen
sind verborgen:

Information hiding



Zugriff auf `private`-Variablen nur über
Getter- und Setter-Methoden

Getter-Methode

Setter-Methode

Eine **Datenkapsel** als Black Box

2.1.3 Exkurs: Ausgabe-Befehl println()

```
1 public class PrintlnDemo
2 {
3     public PrintlnDemo()
4     {
5         ausgeben();
6     }
7
8     public void ausgeben()
9     {
10         int x = 42;
11         double volumen = 123.45;
12
13         System.out.println("Hallo Welt!");
14         System.out.println(x);
15         System.out.println("Sinn des Universums: " + x);
16         System.out.println("Die Kugel hat ein Volumen von " + volumen + " m3");
17     }
18
19 }
```

Ausgabe eines Strings

Ausgabe einer Zahl

Ausgabe von Ausdrücken aus Strings und Zahlen

2.1.3 Exkurs: Ausgabe-Befehl printf()

```
public void formatiertAnzeigen()  
{  
    System.out.printf("Gewicht = %6.2f kg%n", gewicht);  
    System.out.printf("Groesse = %6.2f cm%n", groesse);  
}
```

Platzhalter für eine
Gleitkommazahl

Insgesamt 6
Zeichen lang

Davon 2 Nach-
kommastellen

Wert, der in den Platzhal-
ter eingesetzt wird

2.1.4 Die zweite Version der Klasse (a)



```
1 public class Waage
2 {
3     private double gewicht, groesse;
4
5     public Waage()
6     {
7         setGewicht(67.8);
8         setGroesse(178.9);
9         ausgeben();
10    }
```

Methodenaufrufe im
Konstruktor

Einführung in die Objektorientierte Programmierung (OOP)

2.1.4 Die zweite Version der Klasse (b)

```
1 public class Waage
2 {
3     private double gewicht, groesse;
4
5     public Waage()
6     {
7         run();
8     }
9
10    public void run()
11    {
12        setGewicht(67.8);
13        setGroesse(178.9);
14        ausgeben();
15    }
```

Aufruf einer run()-Me-
thode



Einführung in die Objektorientierte Programmierung (OOP)

2.1.4 Die zweite Version der Klasse (c)

```
1 public class TesteWaage
2 {
3     private Waage waage;
4
5     public TesteWaage()
6     {
7         waage = new Waage();
8         waage.setGewicht(67.8);
9         waage.setGroesse(175.8);
10        waage.ausgeben();
11    }
12 }
```

Objekt **waage** soll getestet werden.

Objekt **waage** wird erzeugt.

Methoden von **waage** werden getestet.

Einführung in die Objektorientierte Programmierung (OOP)

2.1.4 Die zweite Version der Klasse (d)

```
1 public class TesteWaage
2 {
3     public static void main(String[] args)
4     {
5         Waage waage = new Waage();
6         waage.setGewicht(67.8);
7         waage.setGroesse(175.8);
8         waage.ausgeben();
9     }
10 }
```

Richtige Java-Apps
starten mit einer
`main()`-Methode.

Diese erzeugt dann
ein neues Objekt der
zu testenden Klasse.

Einführung in die Objektorientierte Programmierung (OOP)

2.1.6 Die dritte Version der Klasse Waage

```
public double getGroesse()  
{  
    return groesse;  
}  
  
public double getGewicht()  
{  
    return gewicht;  
}  
  
public double getIdealgewicht()  
{  
    return (groesse-100) * 0.9;  
}
```

Direkte Getter-Methoden

Liefert Wert einer Instanzvariable unverändert zurück

Indirekte Getter-Methode

Liefert Wert zurück, der aus Instanzvariablen berechnet wurde.

2.1.7 Instanzvariablen und Attribute

```
public double getGroesse()  
{  
    return groesse;  
}  
  
public double getGewicht()  
{  
    return gewicht;  
}  
  
public double getIdealgewicht()  
{  
    return (groesse-100) * 0.9;  
}
```

Die **Instanzvariablen** des Objektes sind:

```
double gewicht  
double groesse
```

Die **Attribute** des Objektes sind:

```
Gewicht  
Größe  
Idealgewicht
```

2.1.8 Der printf()-Befehl, Fortsetzung

```
System.out.printf("%-13s %6.2f kg%n", "Gewicht", getGewicht());  
System.out.printf("%-13s %6.2f cm%n", "Groesse", getGroesse());  
System.out.printf("%-13s %6.2f kg%n", "Idealgewicht", getIdealgewicht());
```

Platzhalter für
einen String

Platzhalter für eine
Gleitkommazahl

Der einzusetzende
String

Die einzusetzende
Gleitkommazahl