

Overload von Konstruktoren und Instanzmethoden

Overload

- Der Standardkonstruktor lässt es nur zu, Objekte mit identischen Attributwerten zu erzeugen.

```
21 private int tiefe;
22
23
24 /**
25  * Erzeuge einen neuen Stuhl mit einer Standardfarbe und Standardgrosse
26  * an einer Standardposition. (Standardkonstruktor)
27  */
28 public Stuhl() {
29     xPosition = 160;
30     yPosition = 80;
31     farbe = "blau";
32     orientierung = 0;
33     istSichtbar = false;
34     breite = 40;
35     tiefe = 40;
36 }
37
38 /**
39  * Berechnet das zu zeichnende Shape anhand der gegebenen Daten
40  */
41 private Shape gibAktuelleFigur() {
```

Overload

- Overload ermöglicht es, Konstruktoren mit unterschiedlicher Zahl von Parametern nutzen zu können.

```
28 public Stuhl() {
29     xPosition = 160;
30     yPosition = 80;
31     farbe = "blau";
32     orientierung = 0;
33     istSichtbar = false;
34     breite = 40;
35     tiefe = 40;
36 }
```

```
37
38 /**
39  * Erzeuge einen neuen Stuhl mit einer Standardfarbe und Standardgrosesse
40  */
41 public Stuhl(int xPos, int yPos, int winkel) {
42     xPosition = xPos;
43     yPosition = yPos;
44     farbe = "blau";
45     orientierung = winkel;
46     istSichtbar = false;
47     breite = 40;
48     tiefe = 40;
49 }
```

Overload

- Test:

```
Stuhl stuhl1 = new Stuhl(10, 10, 0);
```

```
stuhl1.zeige();
```

```
Stuhl stuhl2 = new Stuhl(60, 20, 90);
```

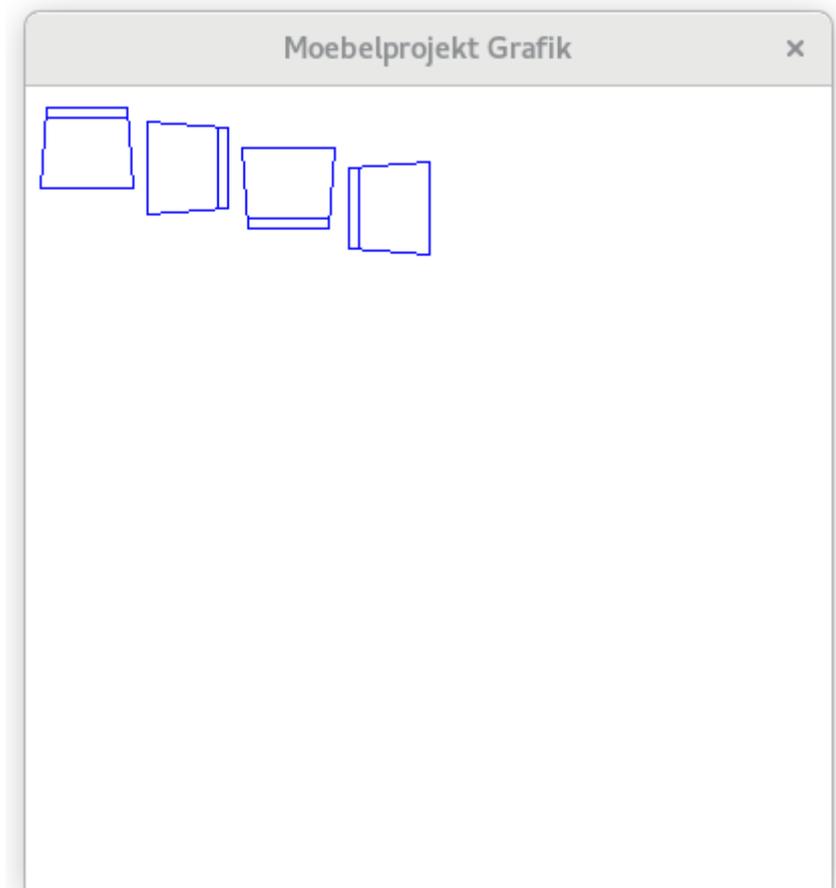
```
stuhl2.zeige();
```

```
Stuhl stuhl3 = new Stuhl(110, 30, 180);
```

```
stuhl3.zeige();
```

```
Stuhl stuhl4 = new Stuhl(160, 40, 270);
```

```
stuhl4.zeige();
```



Overload

- Java lässt auch Methoden mit gleichem Namen bei unterschiedlicher **Signatur** zu.

```
105  /**
106   * Bewege dieses Objekt
107   */
108  public void bewege(int xWeite, int yWeite) {
109      bewegeHorizontal(xWeite);
110      bewegeVertikal(yWeite);
111  }
112
113  /**
114   * Bewege dieses Objekt allein horizontal
115   */
116  public void bewege(int entfernung) {
117      bewegeHorizontal(entfernung);
118  }
119
```

Overload

- Unterschiedliche **Signatur** ist hier aber **nicht** erfüllt!

```
185  /**
186   * Bewege dieses Objekt
187   */
188  public void bewege(int xWeite, int yWeite) {
189      bewegeHorizontal(xWeite);
118      bewegeVertikal(yWeite);
111  }
```

```
113  /**
114   * Bewege dieses Objekt allein horizontal
115   */
116  public void bewege(int entfernung) {
117      bewegeHorizontal(entfernung);
118  }
```

```
120  /**
121   * Bewege dieses Objekt allein vertikal
122   */
123  public void bewege(int entfernung) {
124      bewegeVertikal(entfernung);
125  }
```

Overload

- Der Grund ist, dass bei einem Aufruf nicht klar wäre, welche Methode verwendet werden soll.

```
105  /**
106   * Bewege dieses Objekt
107   */
108  public void bewege(int xWeite, int yWeite) {
109      bewegeHorizontal(xWeite);
110      bewegeVertikal(yWeite);
111  }
```

```
113  /**
114   * Bewege dieses Objekt allein horizontal
115   */
116  public void bewege(int entfernung) {
117      bewegeHorizontal(entfernung);
118  }
```

```
120  /**
121   * Bewege dieses Objekt allein vertikal
122   */
123  public void bewege(int entfernung) {
124      bewegeVertikal(entfernung);
125  }
```