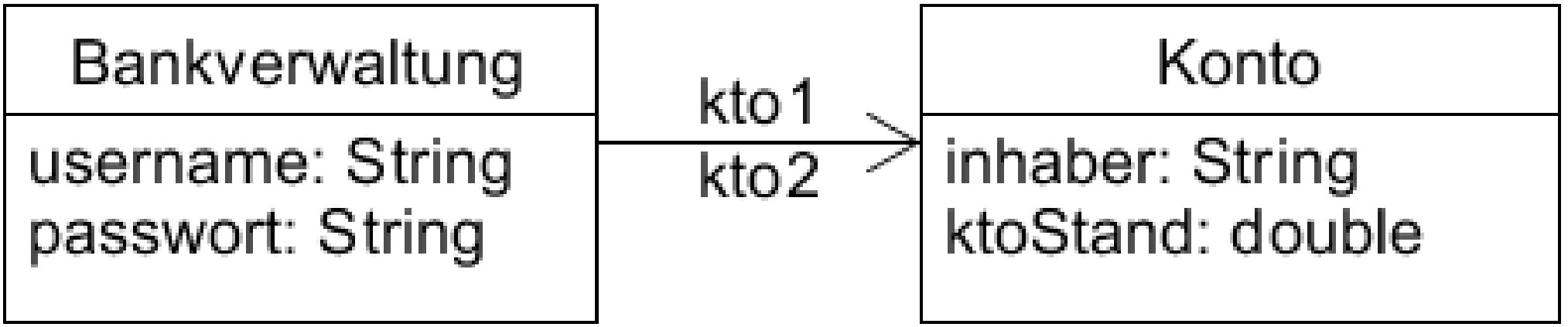
Objekte erzeugen mit dem

Konstruktor

Beispiel: Bankverwaltung



Deklaration von Referenzen

```
class Bankverwaltung
{
    Konto kto1, kto2;
}
```

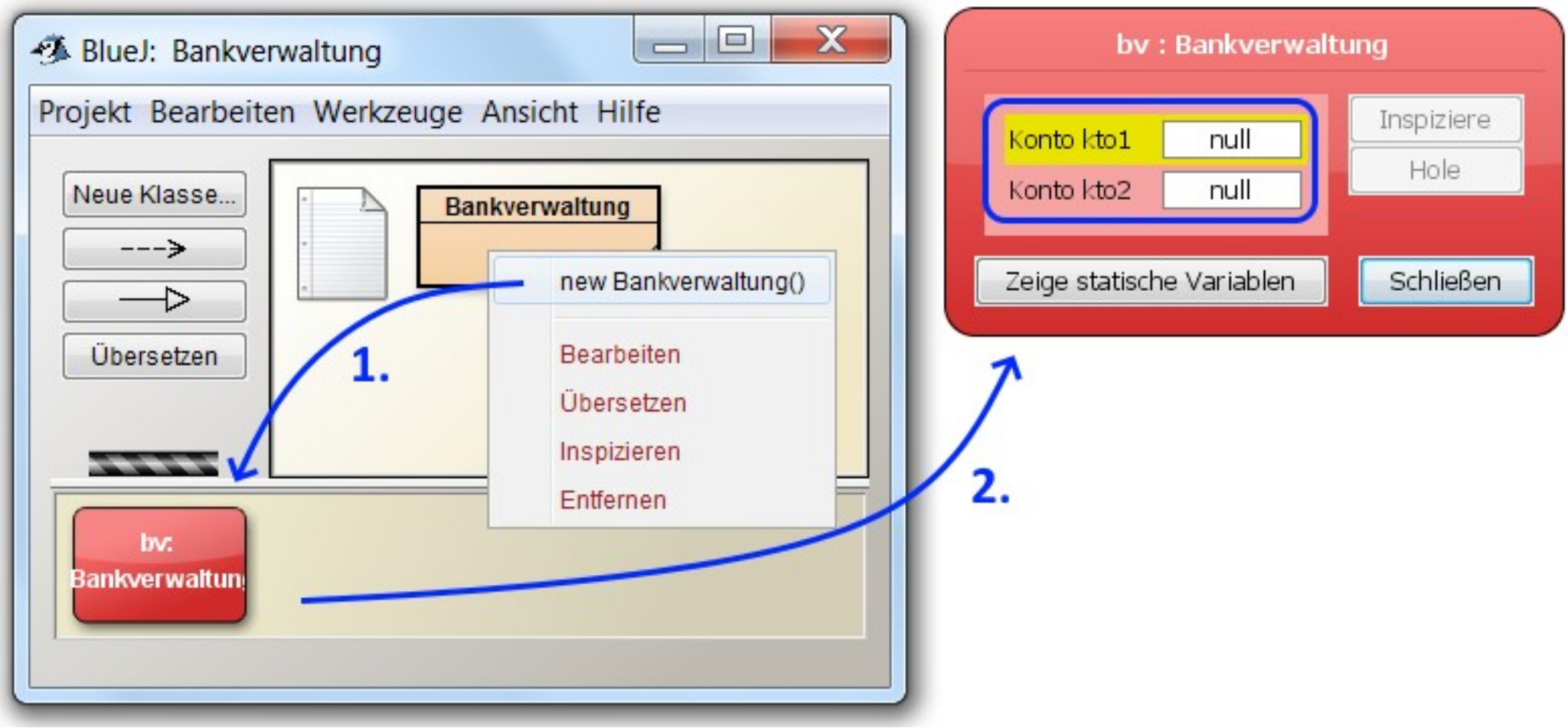
„**Deklaration**“

Deklaration: Ein Objekt der Klasse Bankverwaltung **kann** zwei Objekte der Klasse Konto haben.

Deklaration **erzeugt** diese Objekte aber **nicht**!

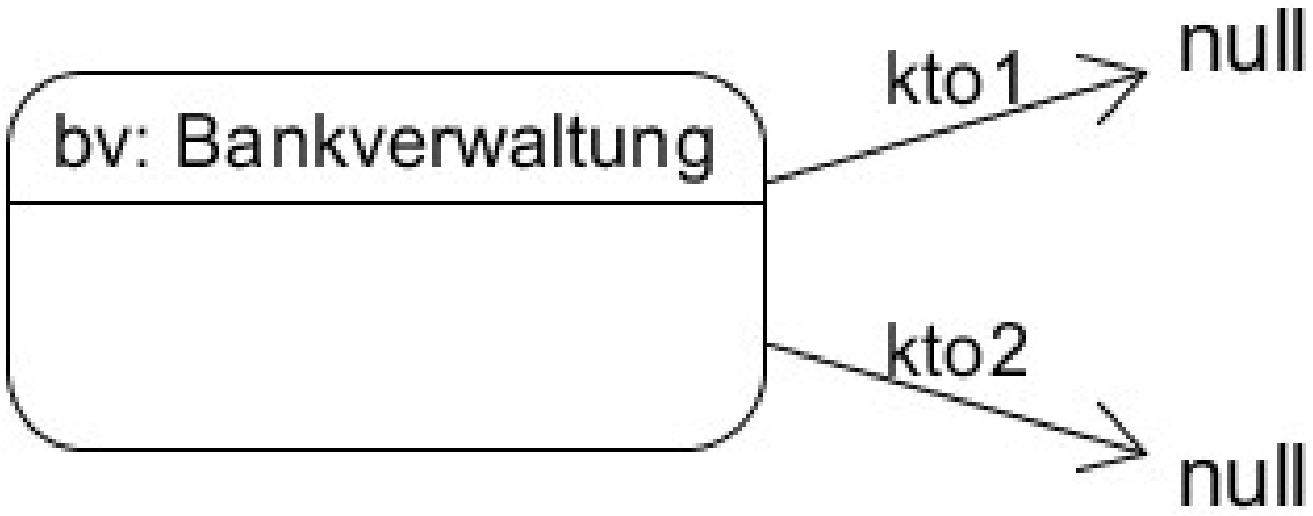
kto1 und kto2 sind „**Referenzen**“, d.h. **Pfeile**, die mit Objekten verbunden werden können.

Erzeugen von Objekten mit BlueJ



Erzeuge Objekt der Klasse Bankverwaltung
→ kto1 und kto2 sind „**null**".

Objekt „bv“ erzeugt



Erzeuge Objekt der Klasse Bankverwaltung
→ kto1 und kto2 sind „**null**“.

Objekte erzeugen im Quellcode

```
class Bankverwaltung
{
    Konto kto1, kto2;
}
```

Objekte erzeugen im Quellcode

```
class Bankverwaltung
{
    Konto kto1, kto2;

    Bankverwaltung ( )
    {
        kto1 = new Konto();
        kto2 = new Konto();
    }
}
```

← „Konstruktor“

Objekte erzeugen im Quellcode

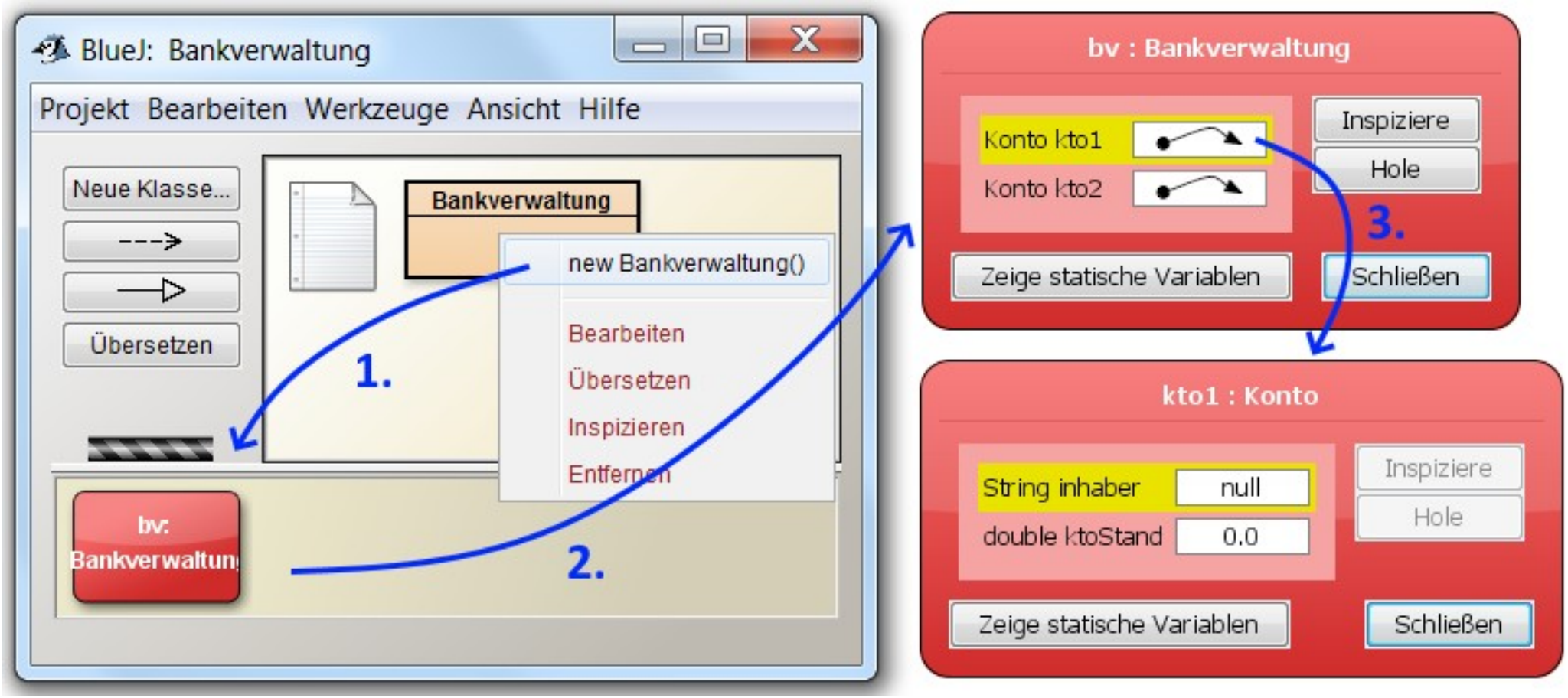
```
class Bankverwaltung
{
    Konto kto1, kto2;

    Bankverwaltung ( )
    {
        kto1 = new Konto();
        kto2 = new Konto();
    }
}
```

„**Konstruktor**“

- Gleicher **Name** wie die Klasse, **Klammern** ( )
- **Erzeugen** eines Objekts: kto1 = new Konto();

Ausführen des Konstruktors

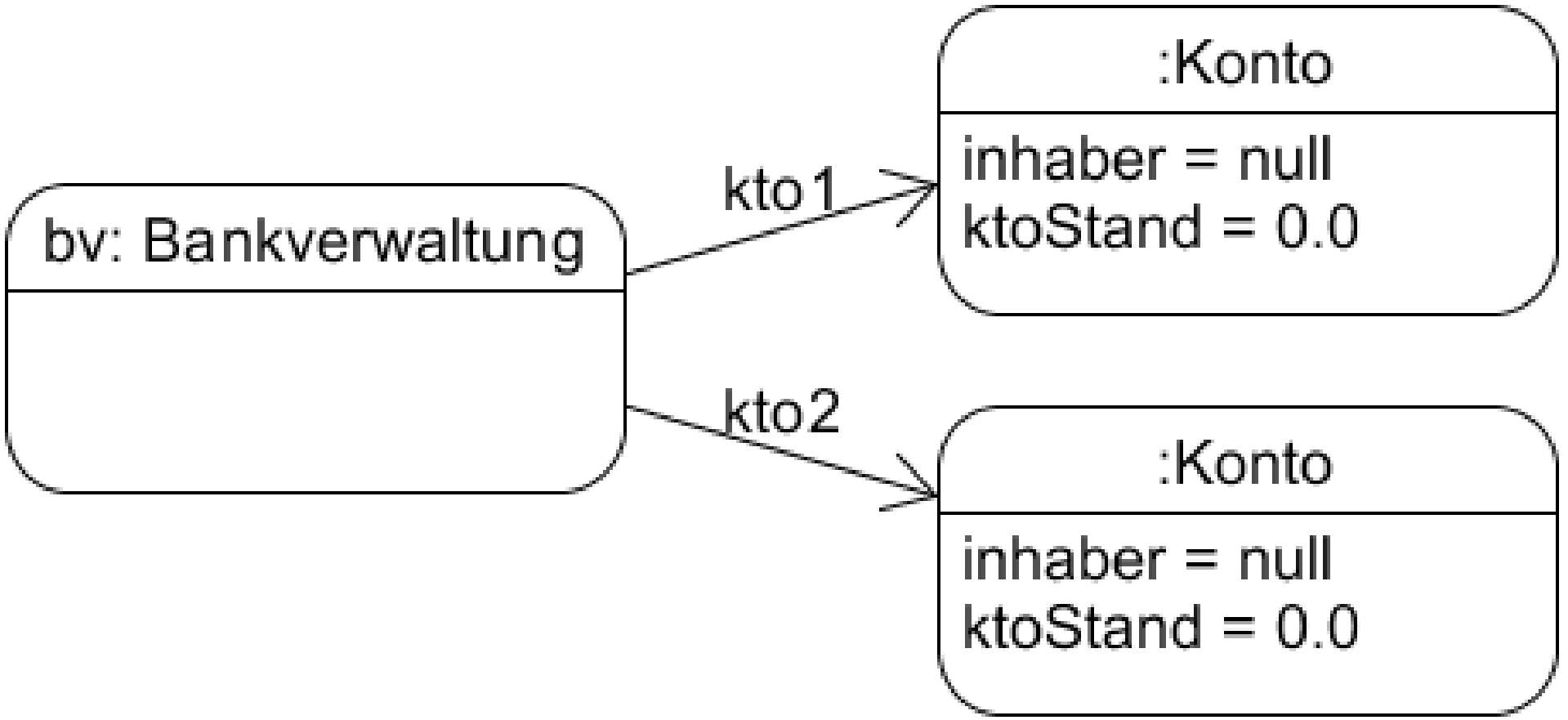


Erzeuge Objekt der Klasse Bankverwaltung

→ Konstruktor wird **automatisch** ausgeführt

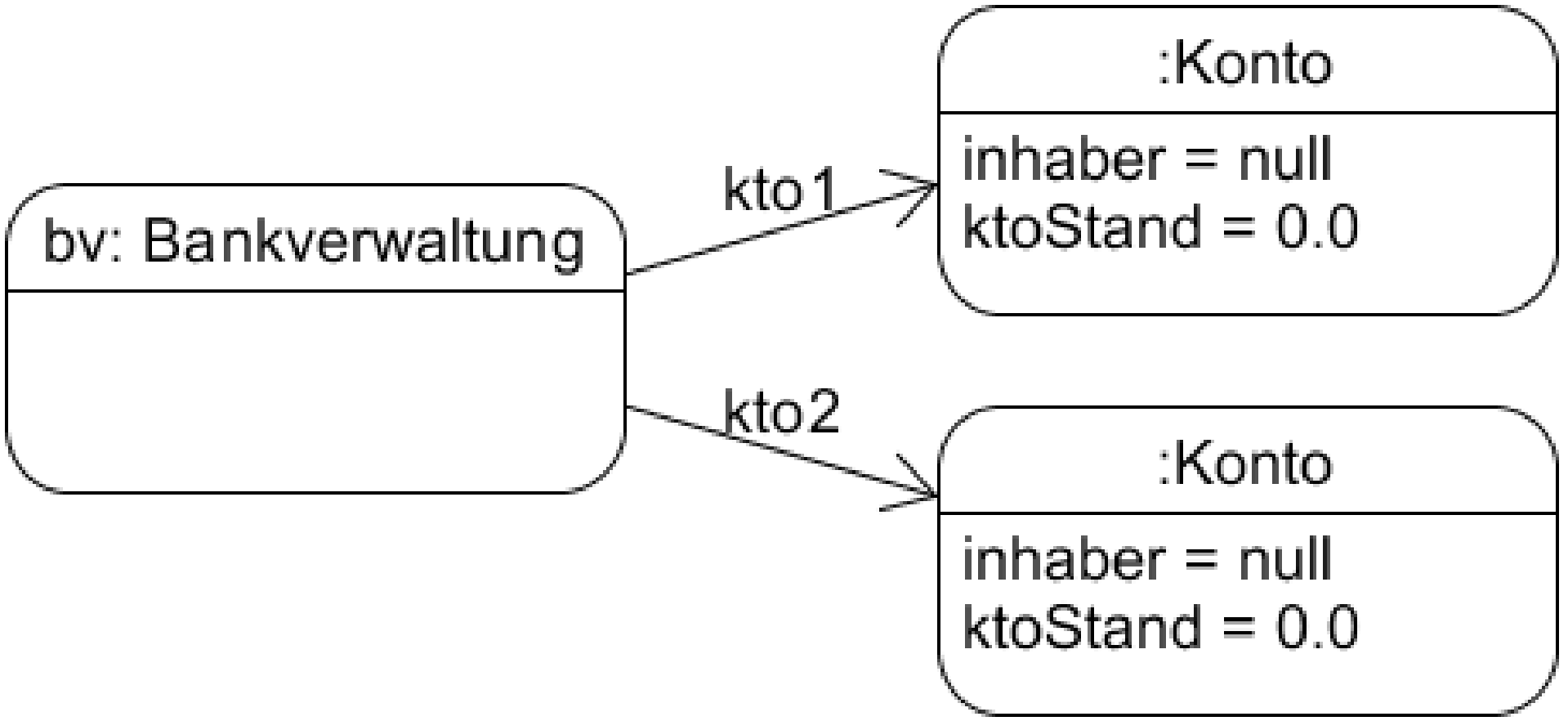
→ Konstruktor **erzeugt** Konto-Objekte

Objekt „bv“ mit Konstruktor erzeugt



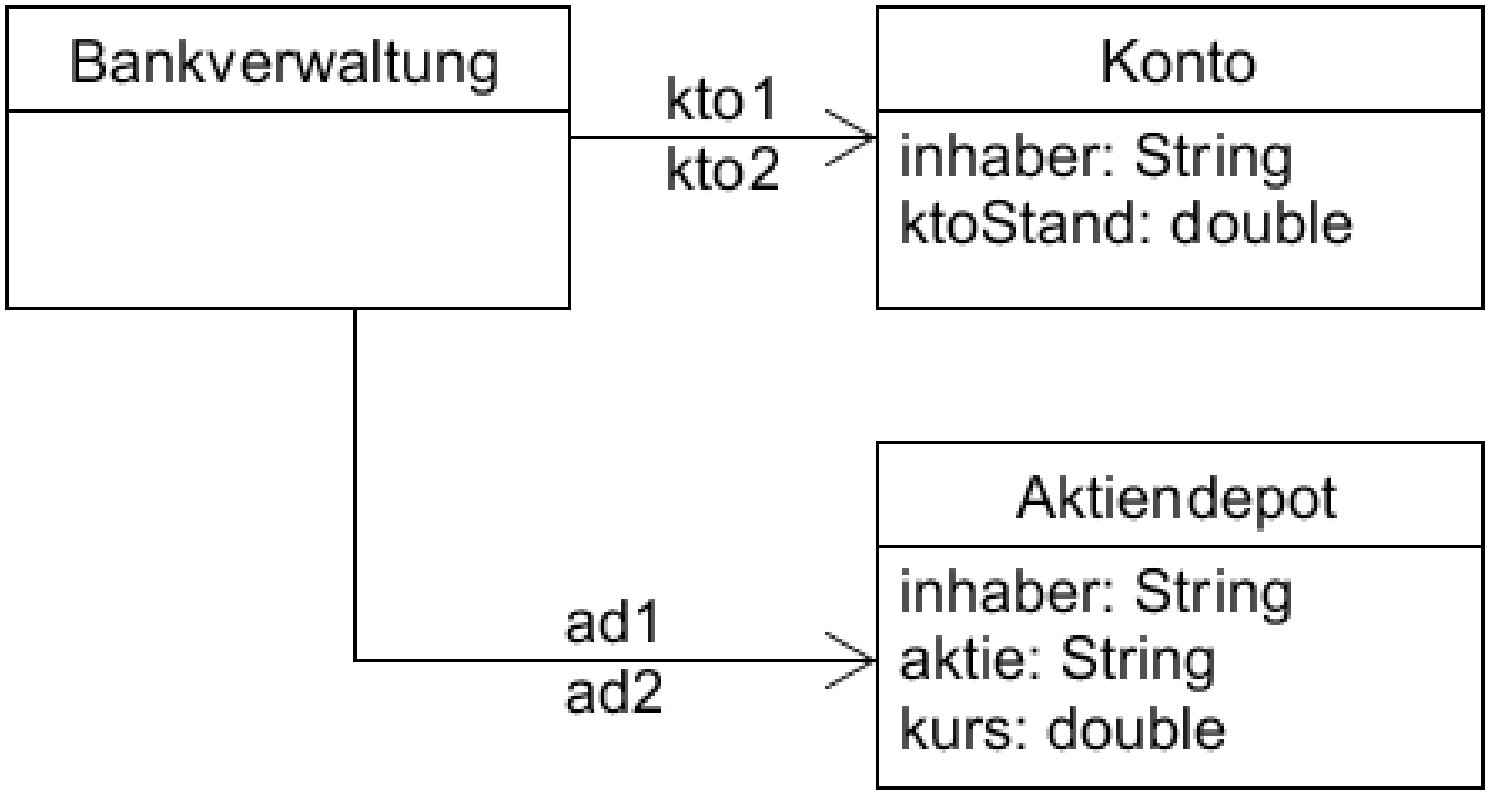
Erzeuge Objekt der Klasse Bankverwaltung
→ kto1 und kto2 zeigen auf **neue** Objekte

Objekt „bv“ mit Konstruktor erzeugt



Attribute neu erzeugter Objekte werden auf 0 (int, double) bzw. null (String) bzw. false (boolean) gesetzt.

Nur ein Konstruktor!



Auch wenn das Objekt der Klasse Bankverwaltung Objekte mehrerer Klasse „hat“, hat die Klasse Bankverwaltung nur einen Konstruktor.

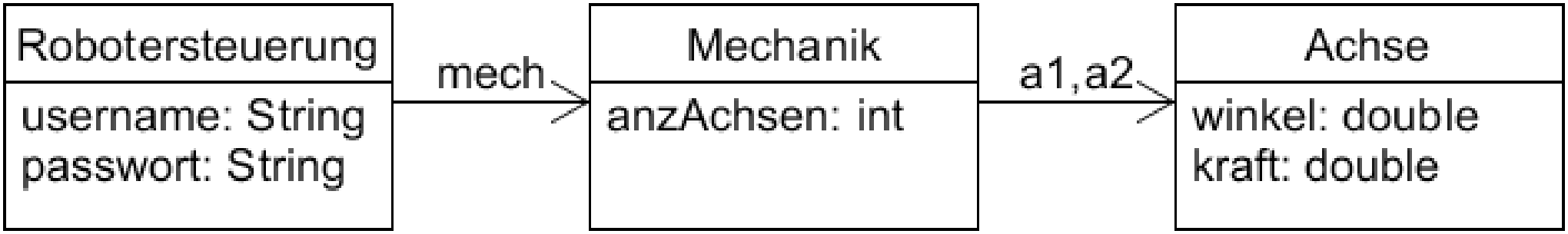
Nur ein Konstruktor!

```
class Bankverwaltung
{
    Konto kto1, kto2;
    Aktiendepot ad1, ad2;

    Bankverwaltung ( )
    {
        kto1 = new Konto();
        kto2 = new Konto();
        ad1  = new Aktiendepot();
        ad2  = new Aktiendepot();
    }
}
```

Weiter nach den Aufgaben

Beispiel: Robotersteuerung



Aufruf des Konstruktors

```
class Robotersteuerung
{
    String username, passwort;
    Mechanik mech;

    Robotersteuerung()
    {
        mech = new Mechanik();
    }
}
```

- **erzeugt** neues Objekt **und**
- **führt** den Konstuktor der Klasse Mechnik **aus**

Aufruf des Konstruktors

```
class Mechanik
{
    int anzAchsen;
    Achse a1, a2;

    Mechanik()
    {
        a1 = new Achse();
        a2 = new Achse();
    }
}
```

- **erzeugt** neue Objekte **und**
- **führt 2x** den Konstuktor der Klasse Achse **aus**

Aufruf des Konstruktors

```
class Achse
{
    double winkel, kraft;
}
```

- Für die Klasse Achse wurde kein Konstruktor angegeben (überflüssig, da keine Unterobjekte).

- Java erzeugt automatisch einen leeren Konstruktor.

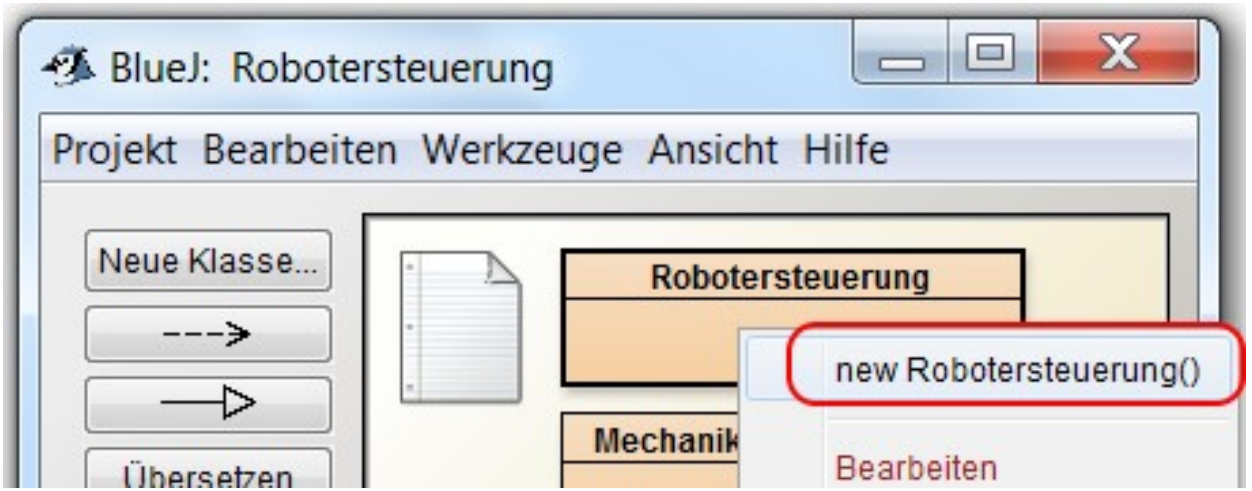
Aufruf des Konstruktors

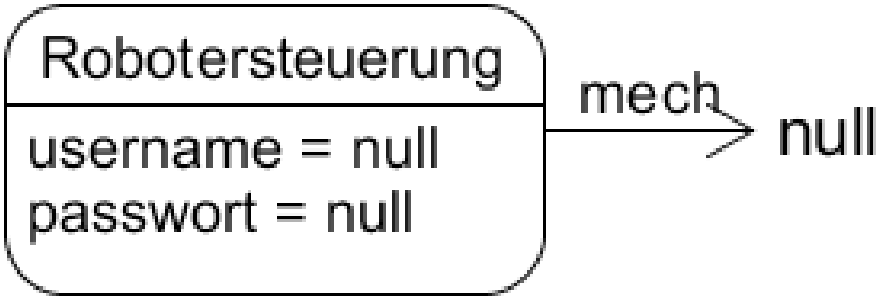
```
class Achse
{
    double winkel, kraft;

    Achse()
    {
    }
}
```

- Für die Klasse Achse wurde kein Konstruktor angegeben (überflüssig, da keine Unterobjekte).

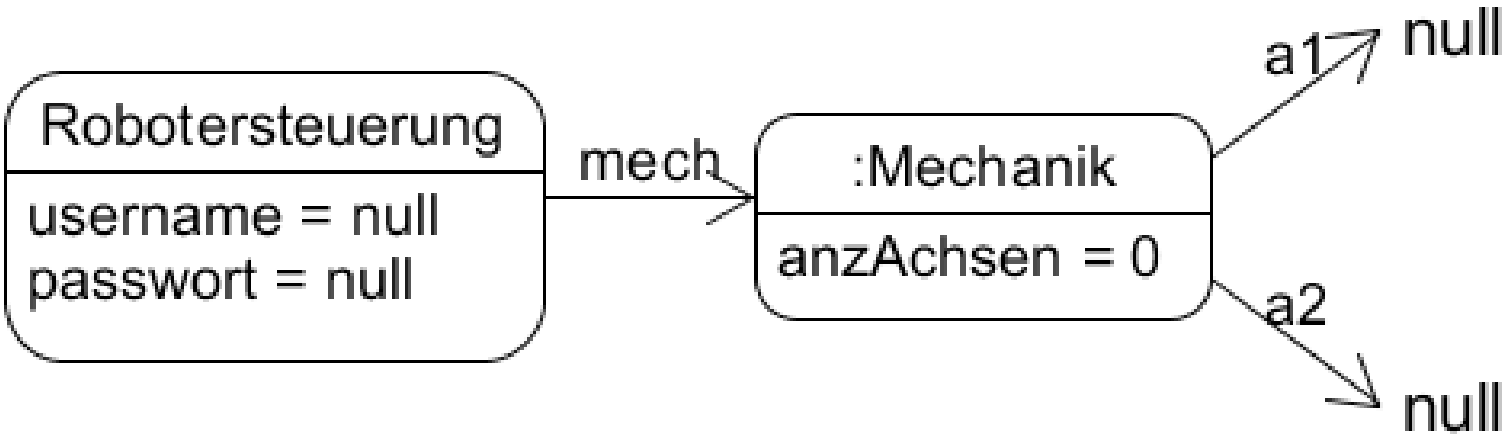
- Java erzeugt automatisch einen leeren Konstruktor.





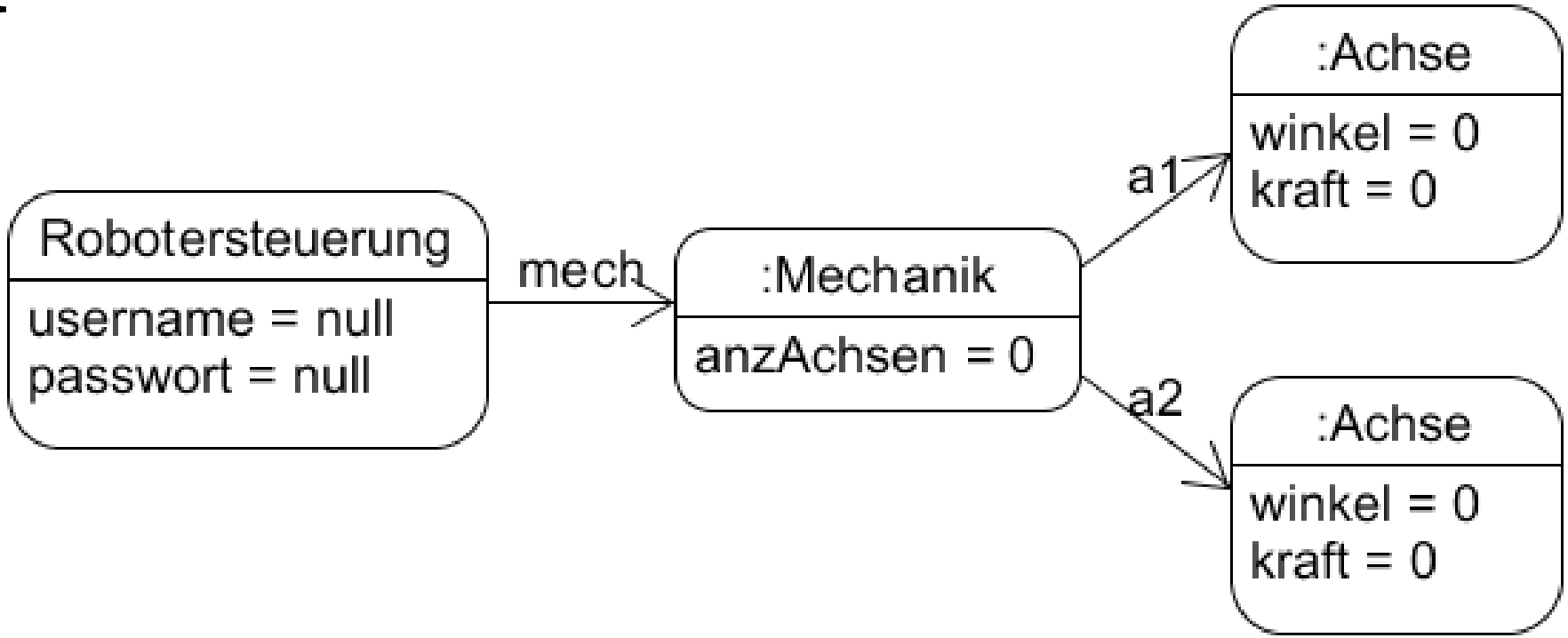
Erstes Objekt wird mit **BlueJ** erzeugt

```
Robotersteuerung()
{
    mech = new Mechanik();
}
```



Unterobjekt wird vom **Konstruktor** erzeugt

```
Mechanik()
{
    a1 = new Achse();
    a2 = new Achse();
}
```



Unterobjekte vom **Konstruktor** erzeugt

Merksätze

```
class Bankverwaltung
{
    Konto kto;

    Bankverwaltung ()
    {
        kto = new Konto();
    }
}
```

Falls ein Objekt Unterobjekte haben soll:

- Gib einen Konstruktor an
- Gleicher Name wie die Klasse, Klammer auf/zu
- Erzeuge Unterobjekte mit new

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
  Freigegeben unter CC BY-NC-SA 4.0, März 2021