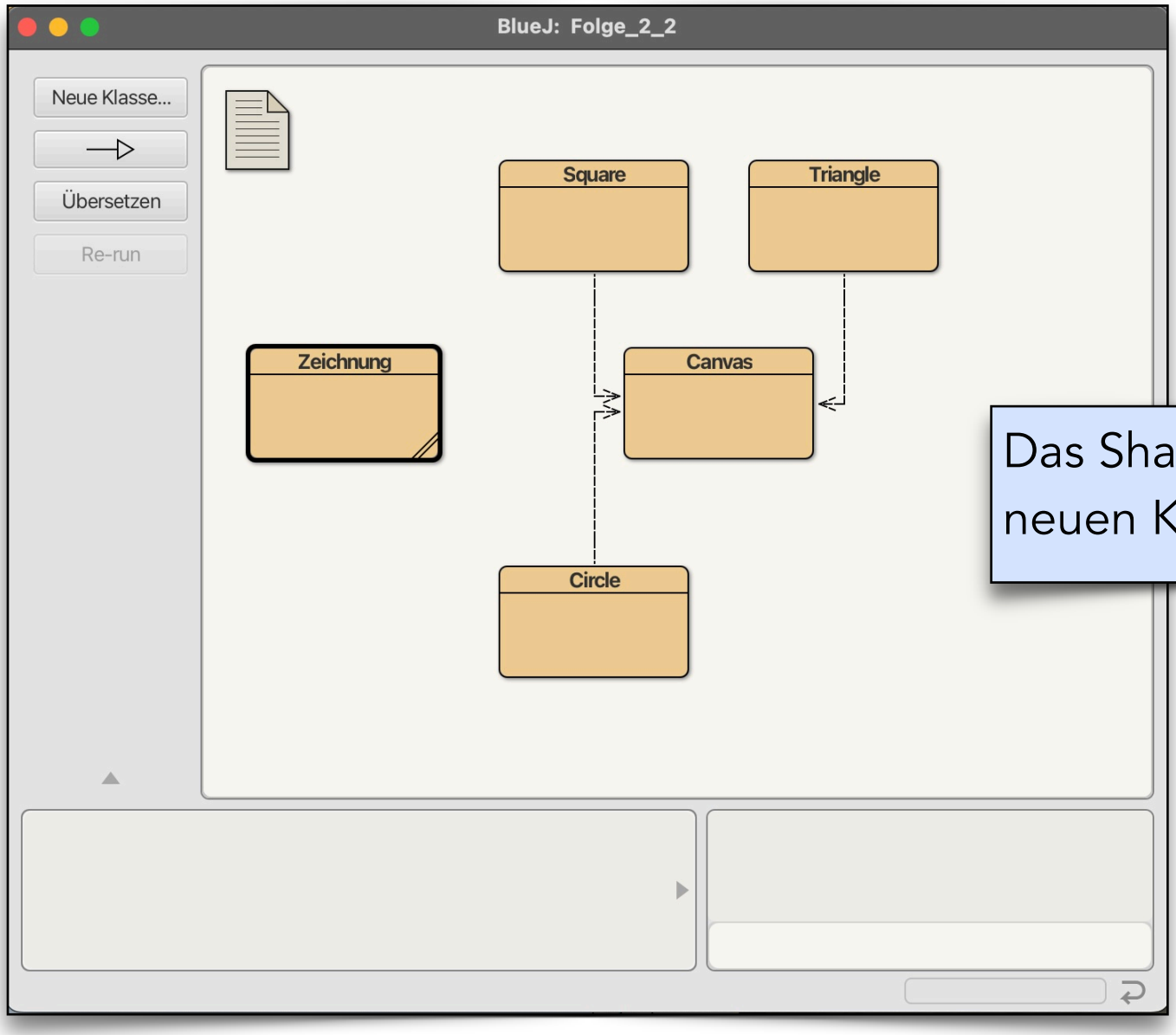
Folge 1.3
Wir programmieren ein kleines Bild

Folge 1.3
Wir programmieren ein kleines Bild

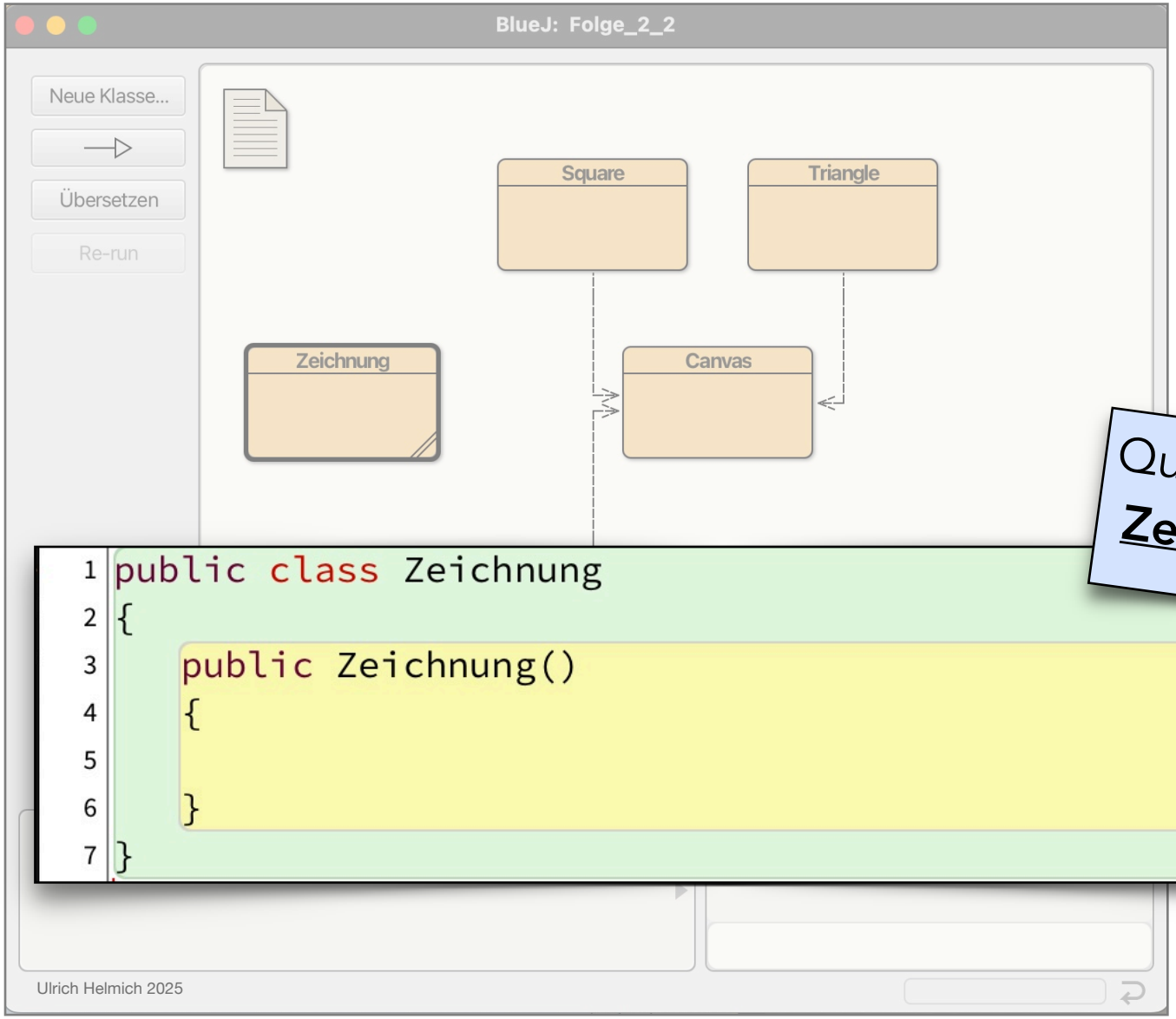
1.3.1 Das Shapes-Projekt, Fortsetzung
1.3.2 Eigene Methoden erstellen

1.3.1 Das Shapes-Projekt, Fortsetzung



Das Shapes-Projekt mit der neuen Klasse **Zeichnung**.

1.3.1 Das Shapes-Projekt, Fortsetzung



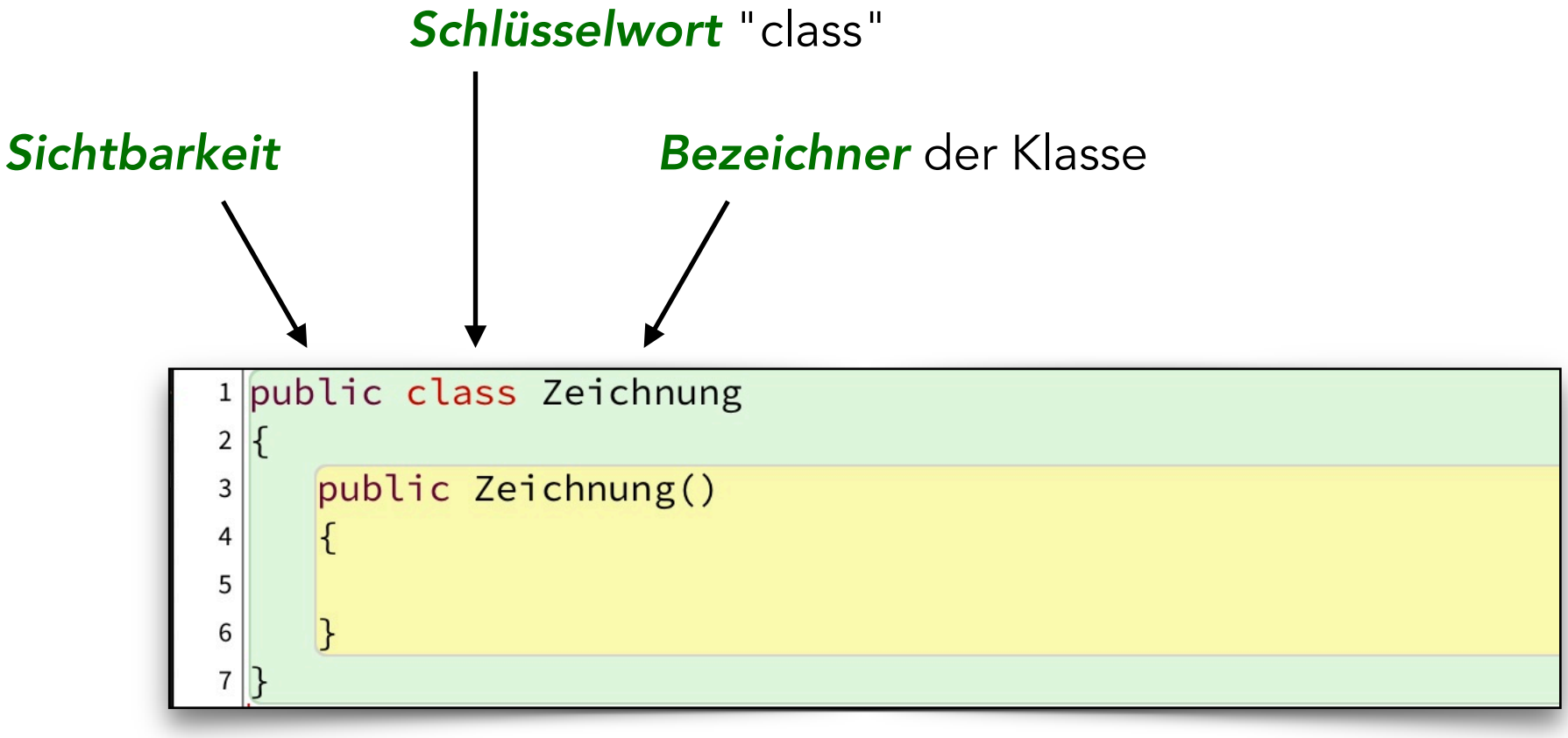

```
1 public class Zeichnung
2 {
3     public Zeichnung()
4     {
5
6     }
7 }
```

Quelltext der Klasse **Zeichnung**.

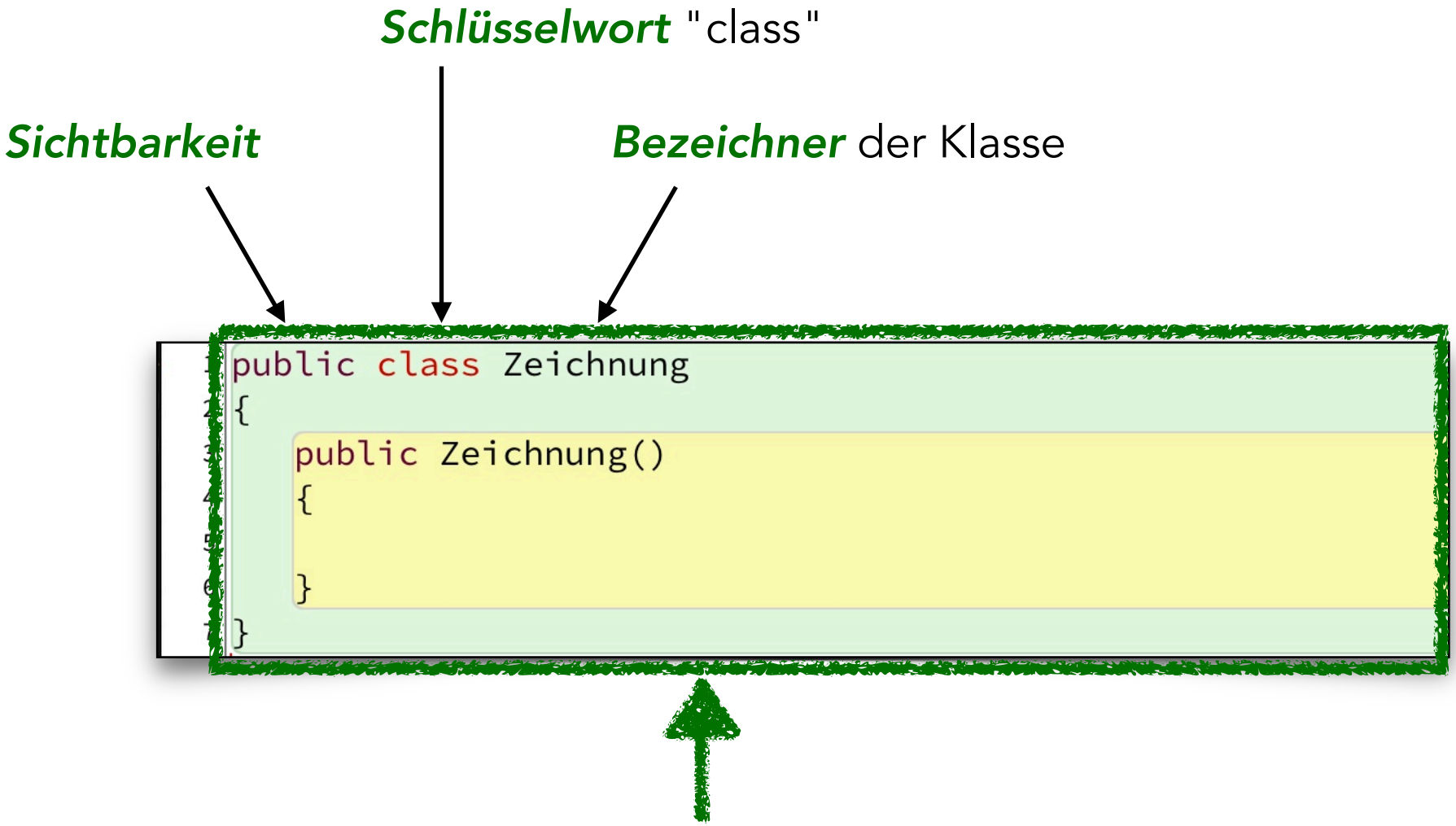
1.3.1 Das Shapes-Projekt, Fortsetzung

```
1  public class Zeichnung
2  {
3      public Zeichnung()
4      {
5
6      }
7  }
```

1.3.1 Das Shapes-Projekt, Fortsetzung

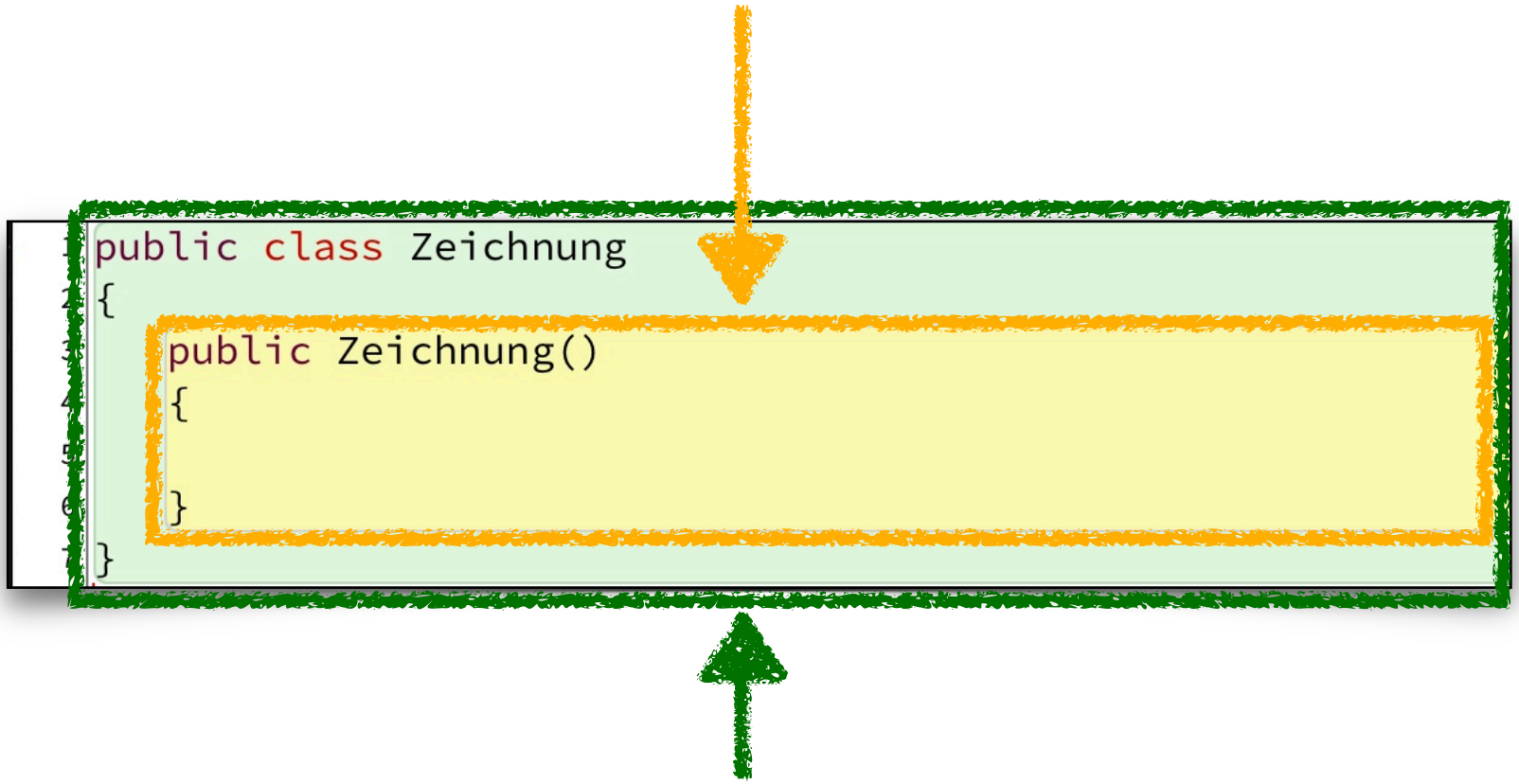


1.3.1 Das Shapes-Projekt, Fortsetzung



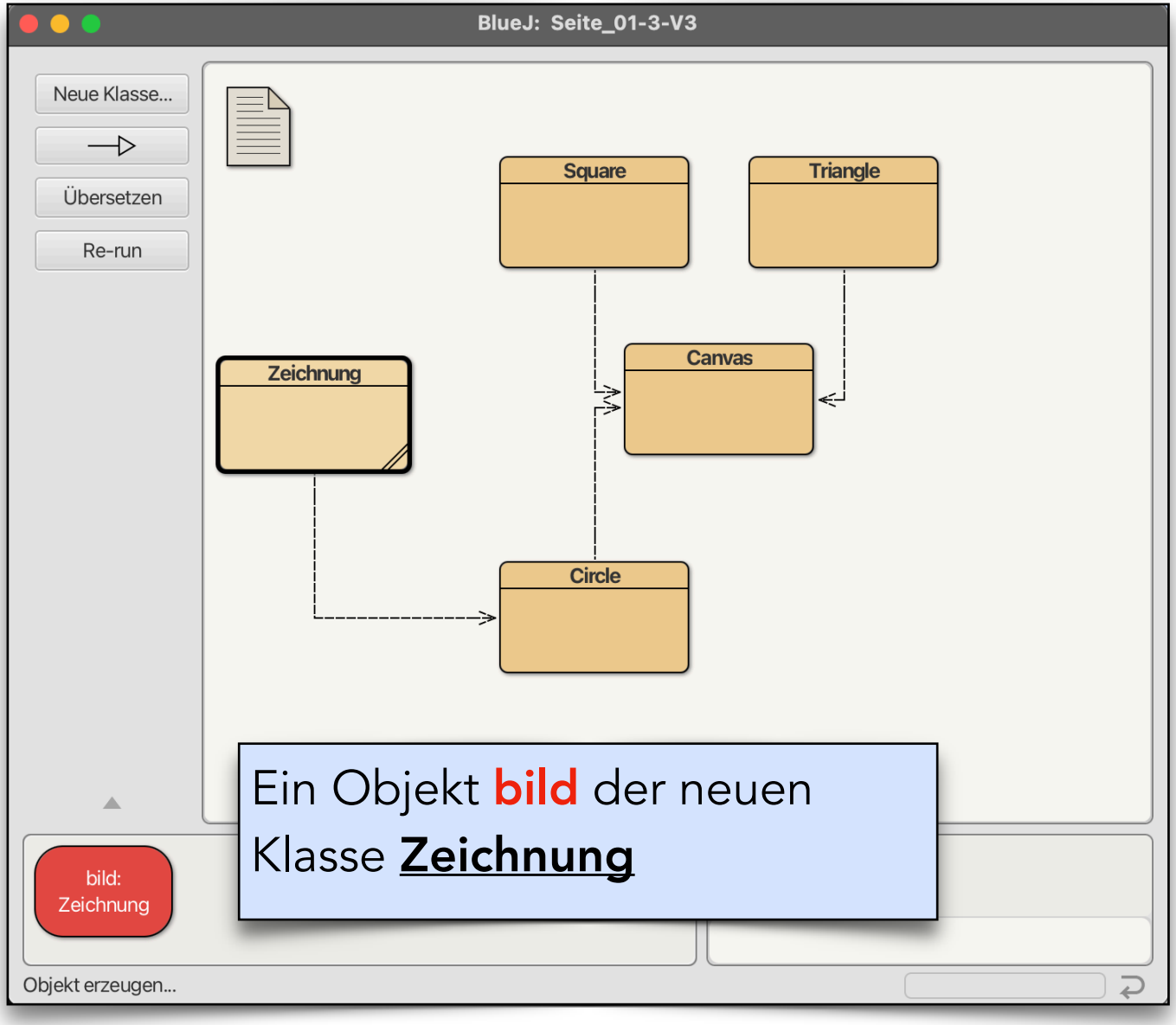
1.3.1 Das Shapes-Projekt, Fortsetzung

Konstruktor der Klasse

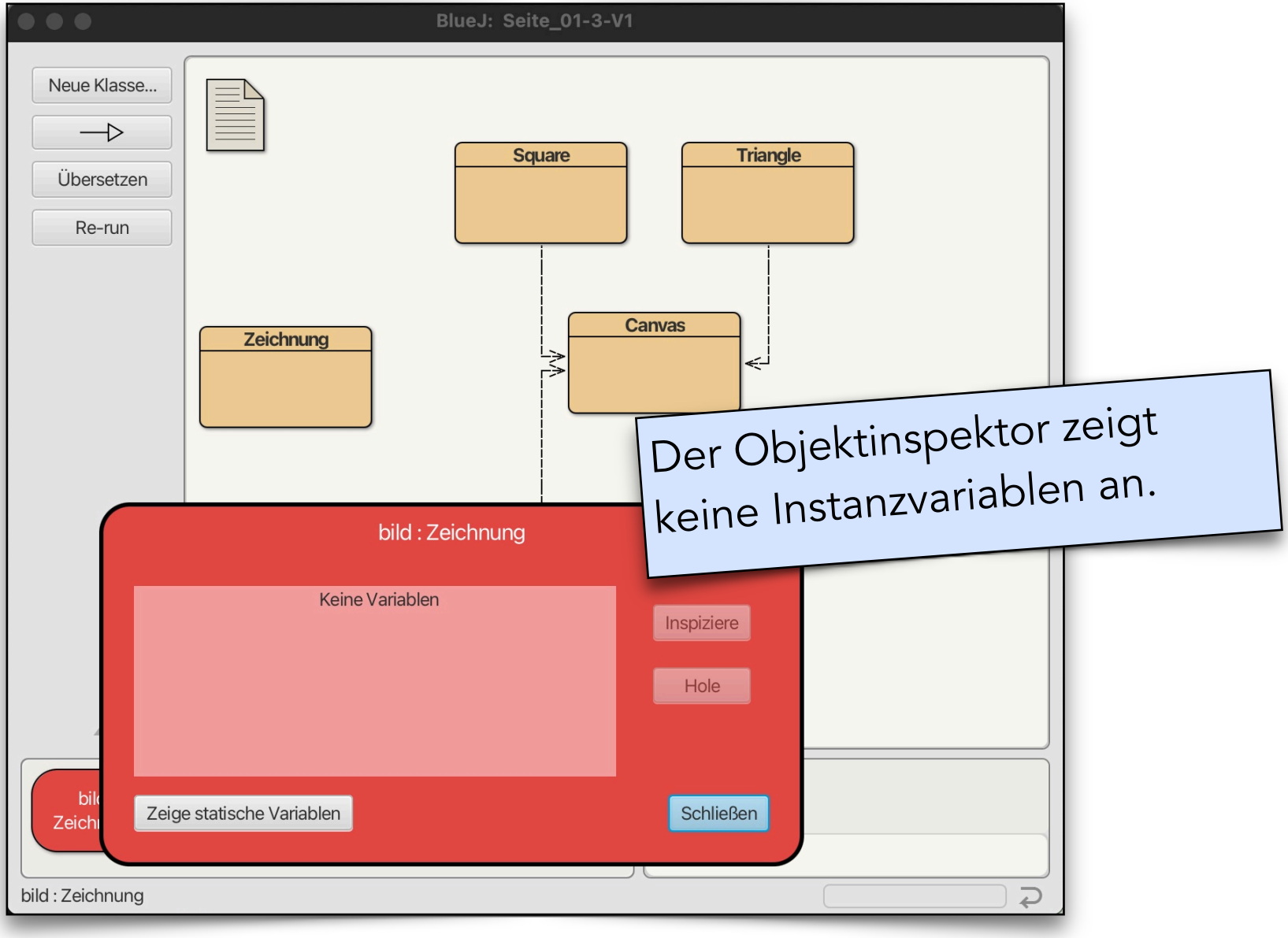


Klasse **Zeichnung**

1.3.1 Das Shapes-Projekt, Fortsetzung

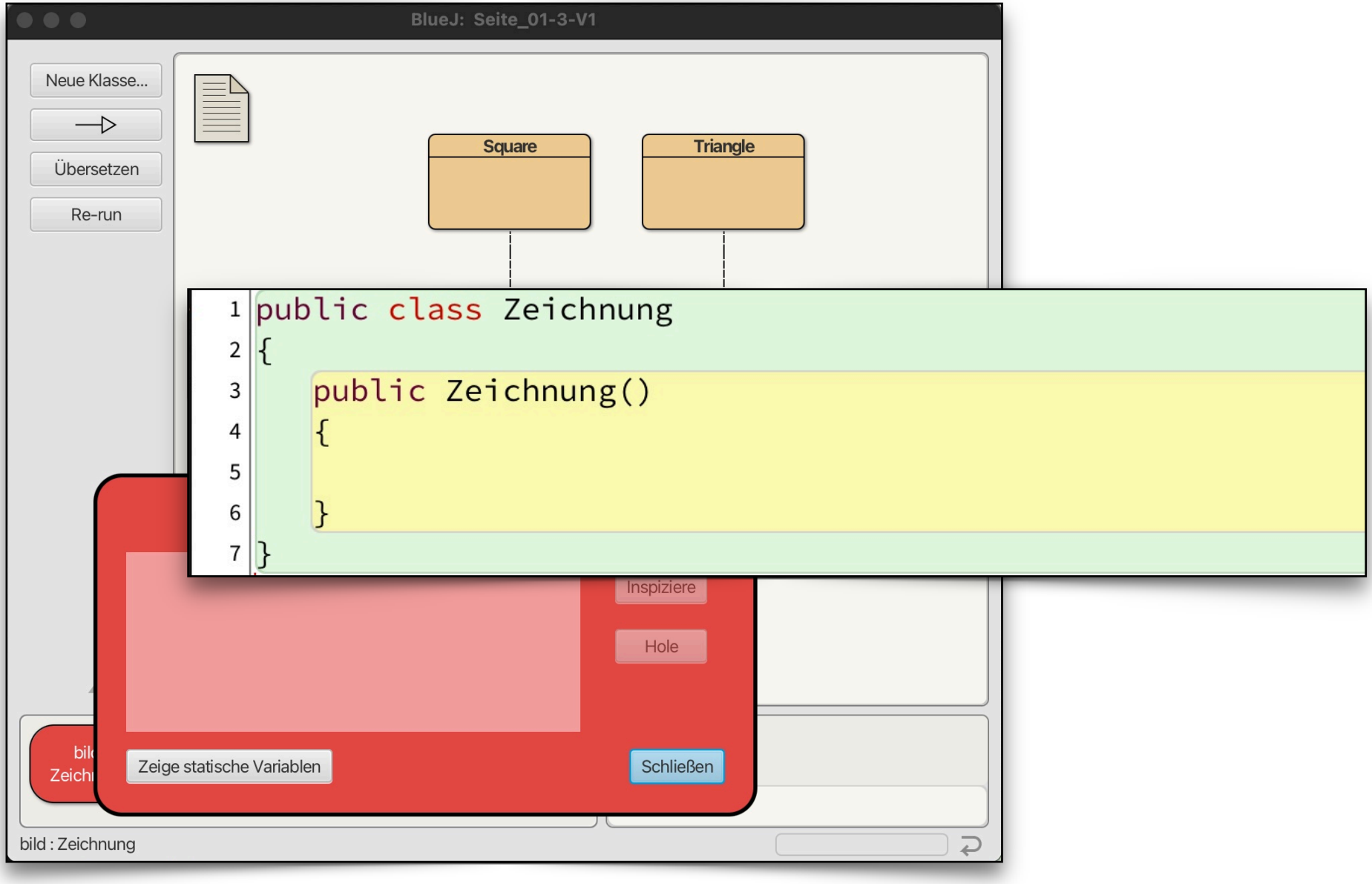


1.3.1 Das Shapes-Projekt, Fortsetzung

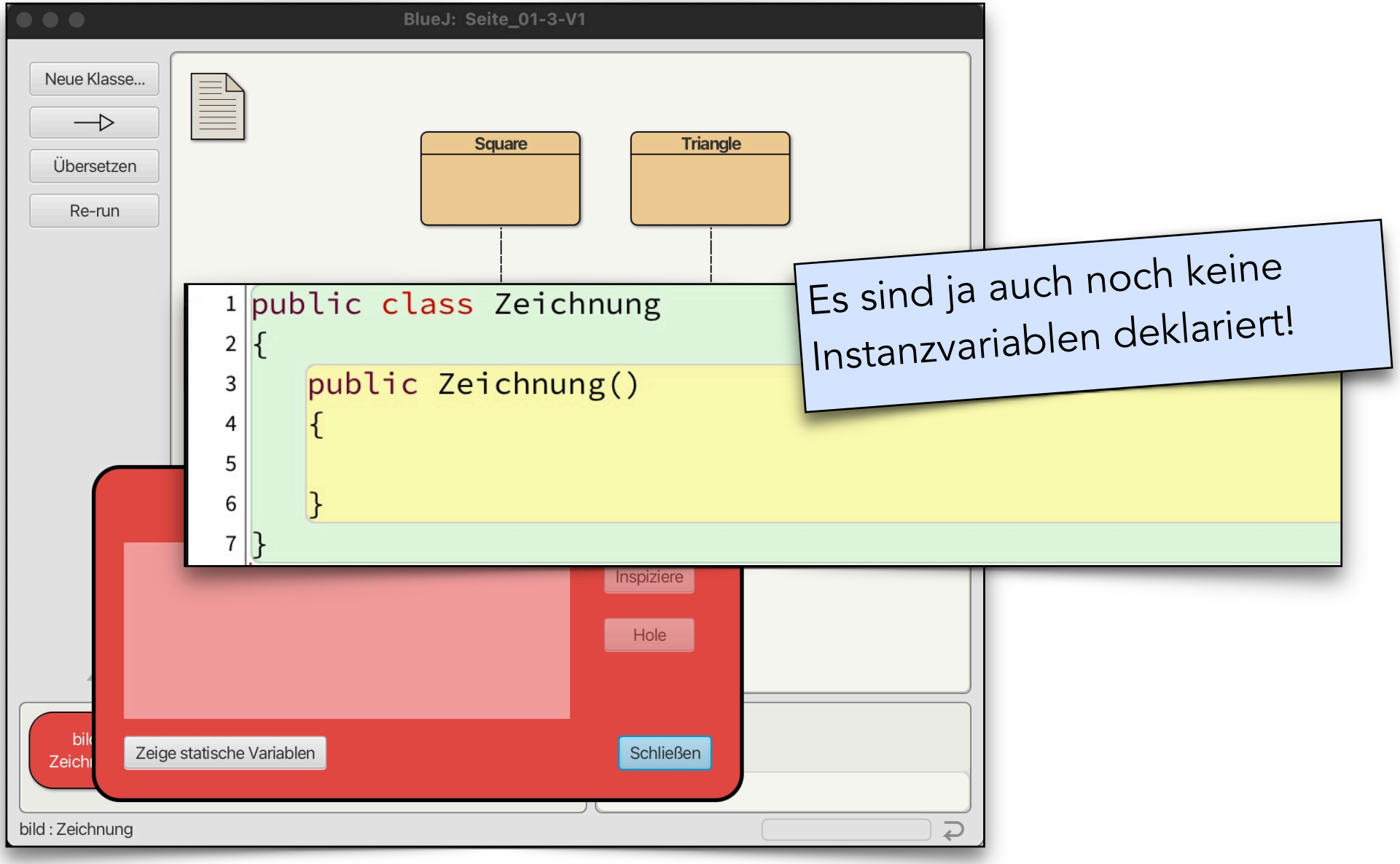


Der Objektinspektor zeigt keine Instanzvariablen an.

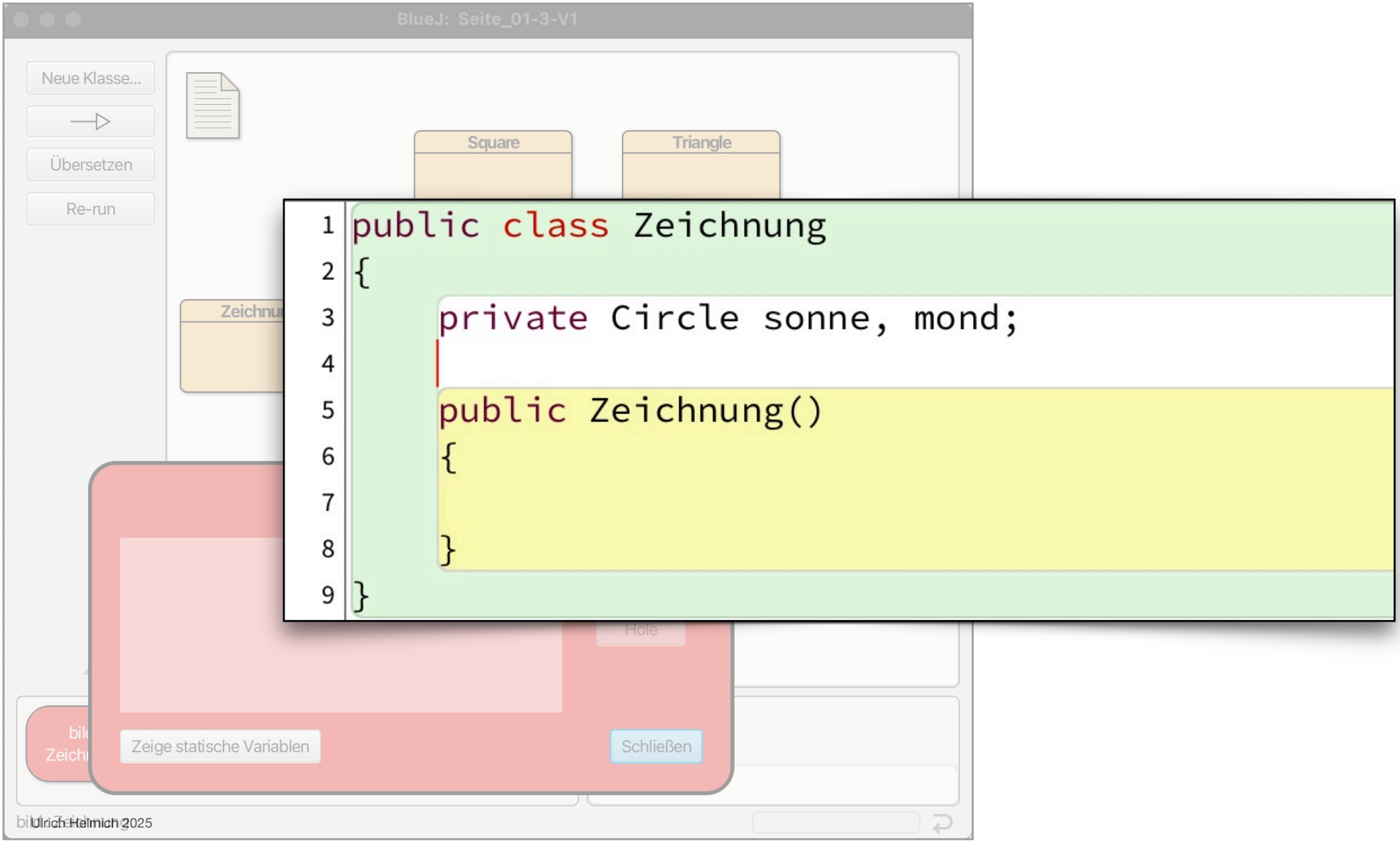
1.3.1 Das Shapes-Projekt, Fortsetzung



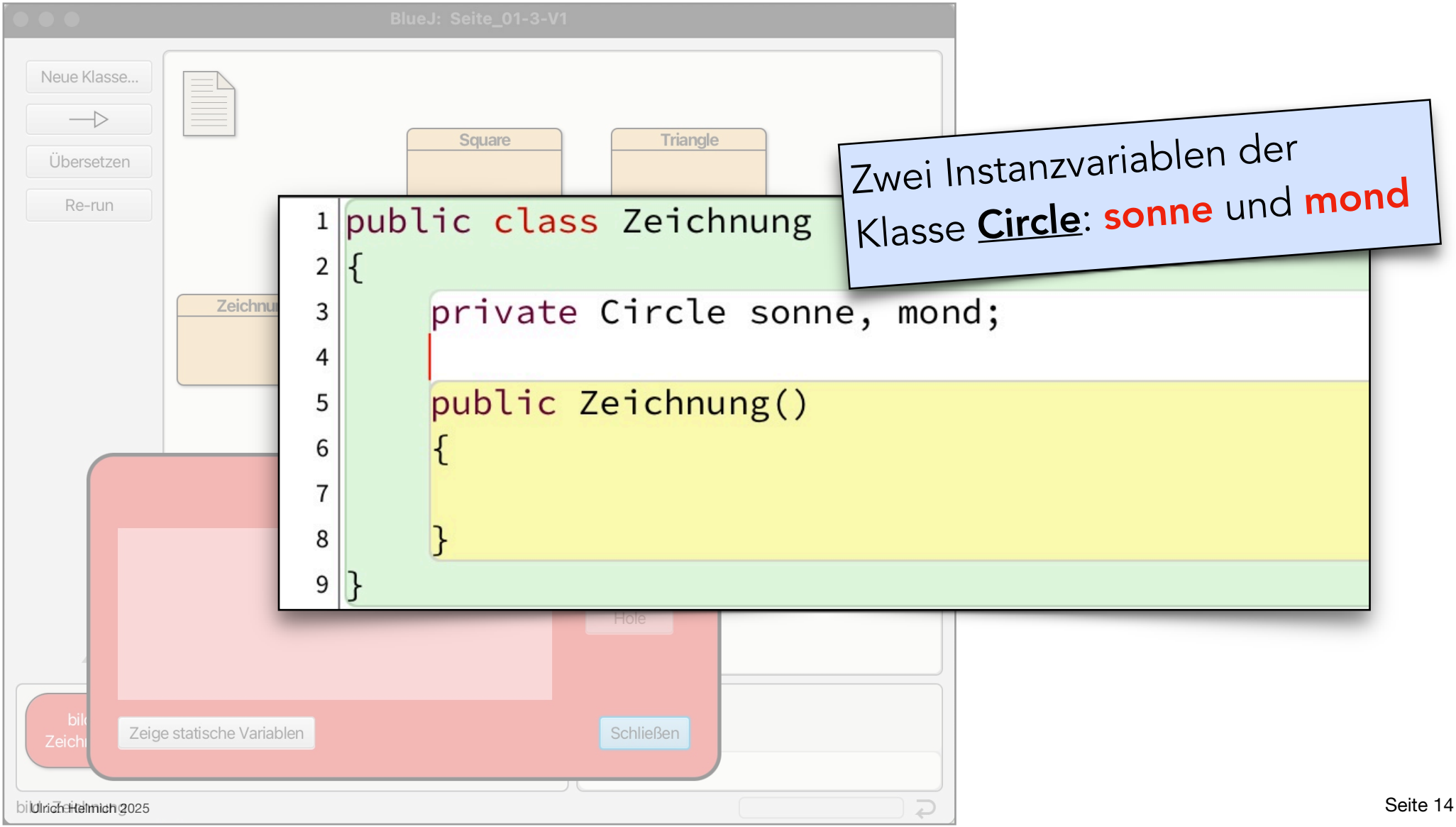
1.3.1 Das Shapes-Projekt, Fortsetzung



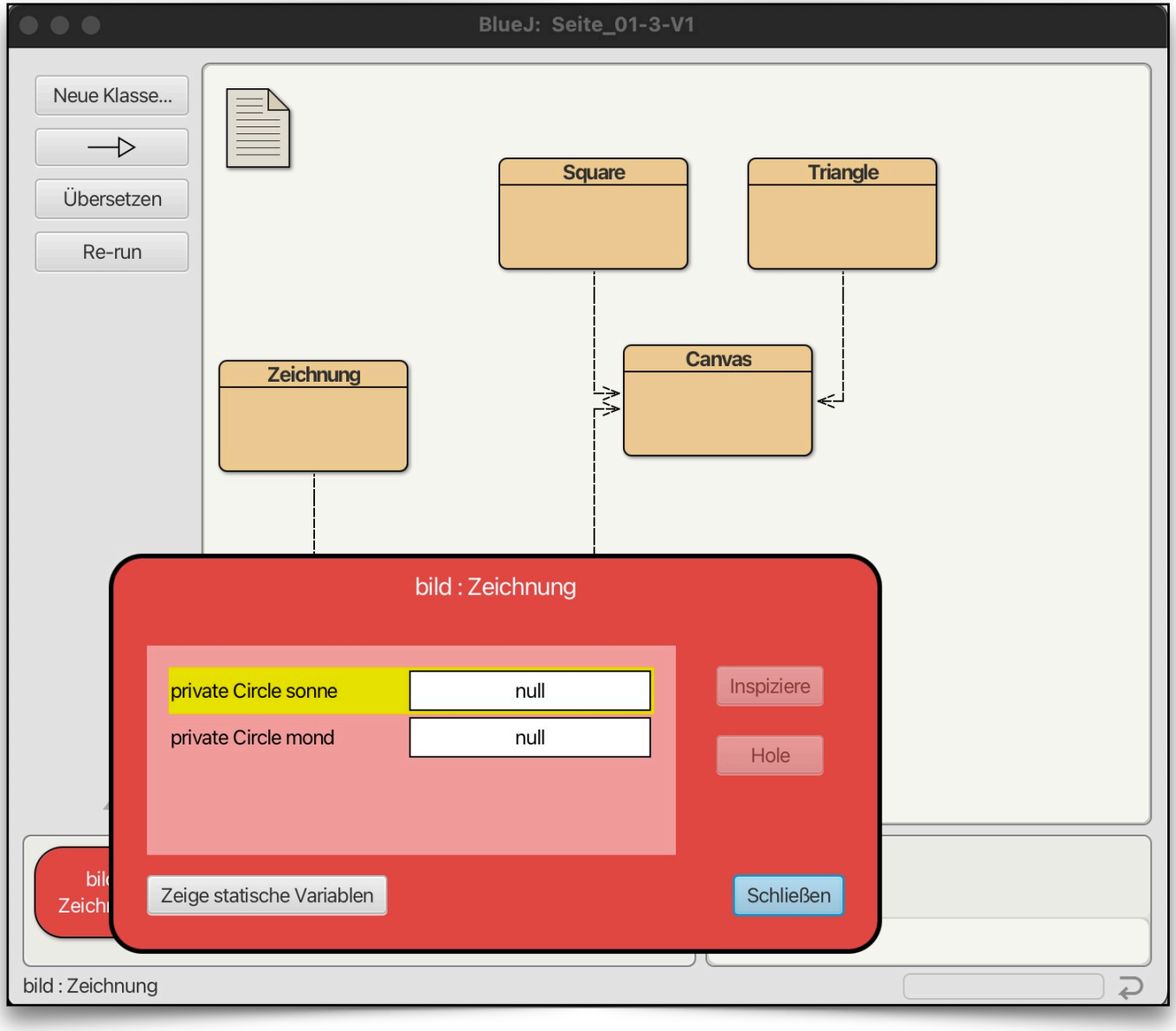
1.3.1 Das Shapes-Projekt, Fortsetzung



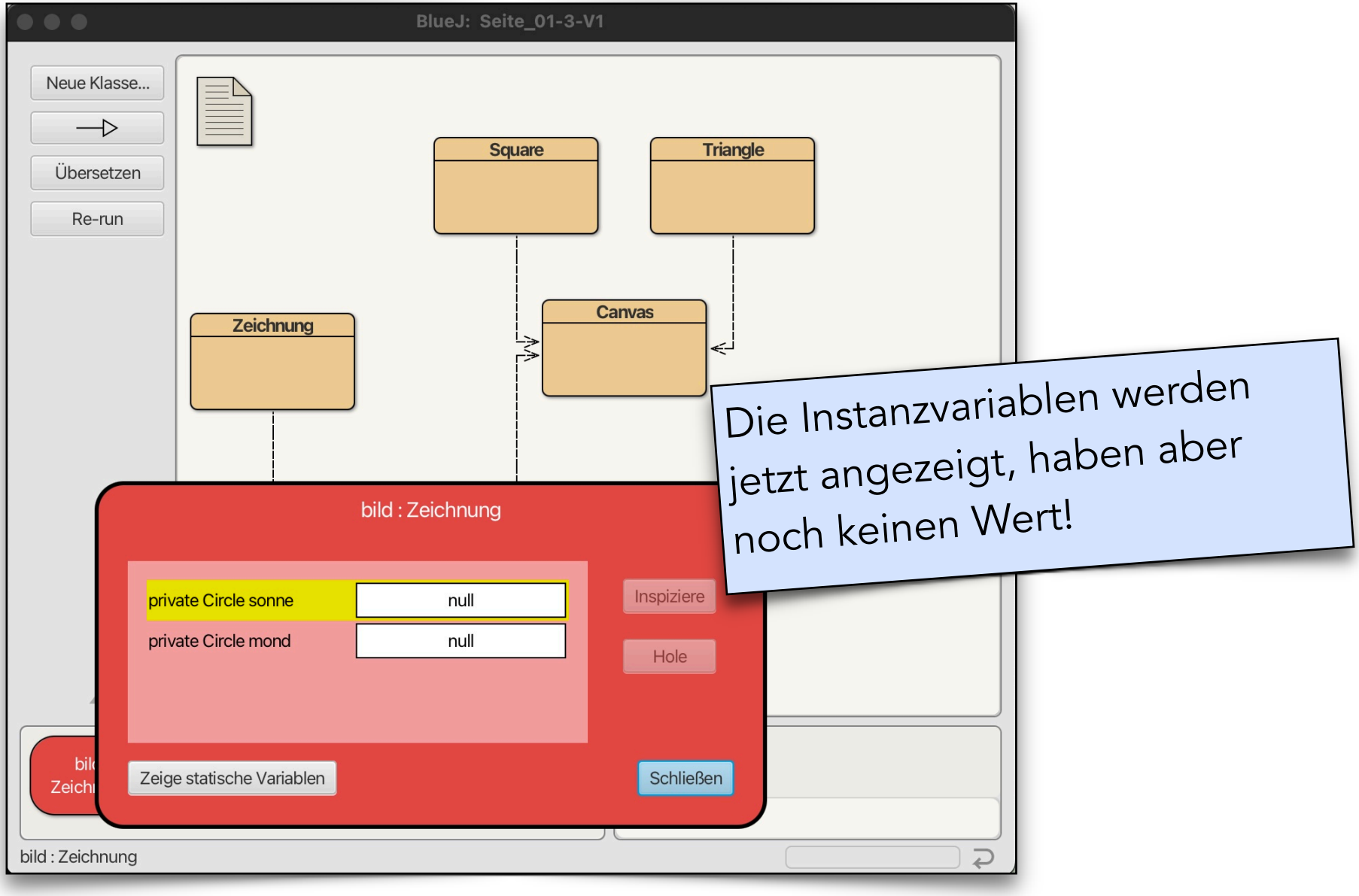
1.3.1 Das Shapes-Projekt, Fortsetzung



1.3.1 Das Shapes-Projekt, Fortsetzung

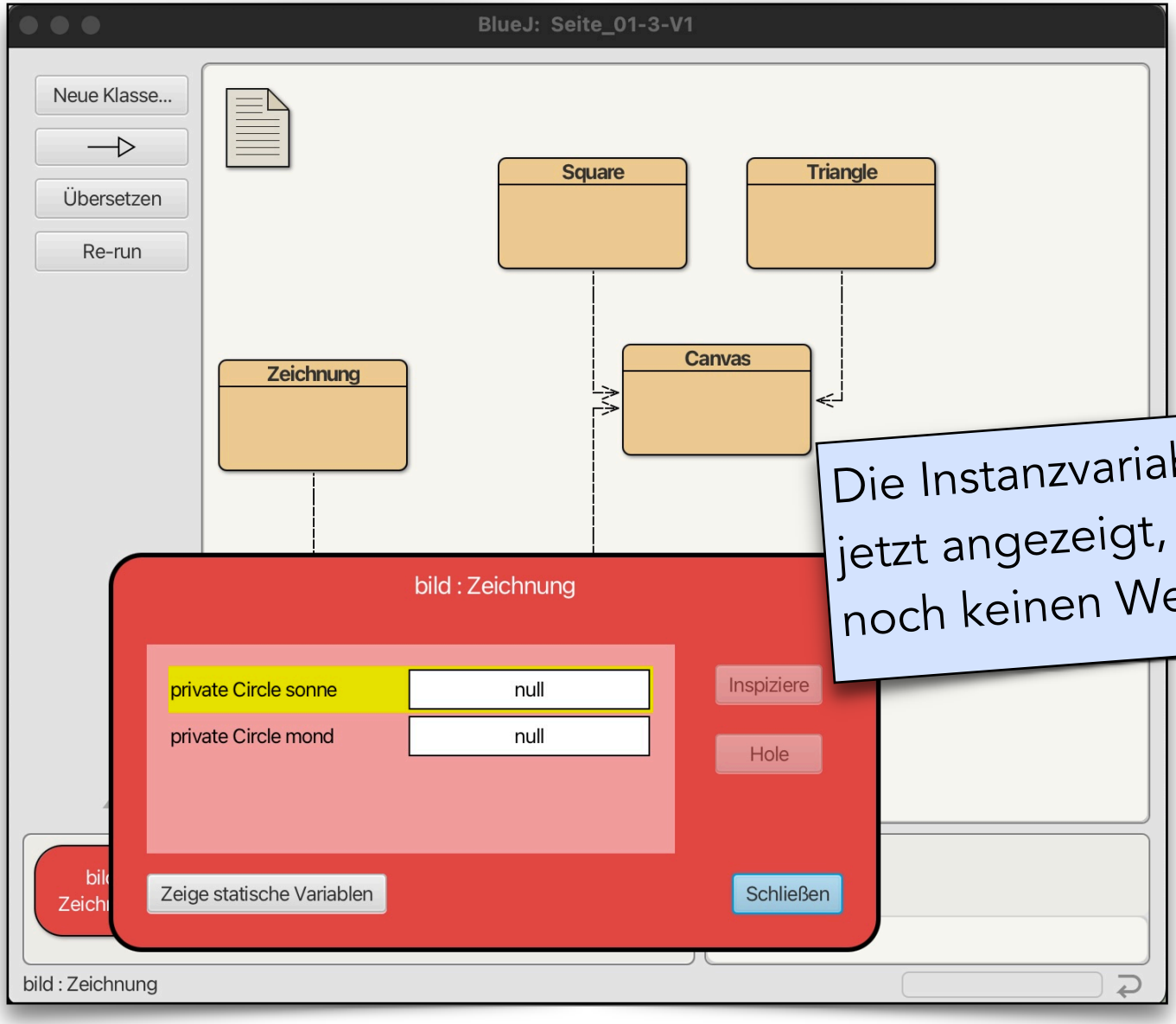


1.3.1 Das Shapes-Projekt, Fortsetzung



Die Instanzvariablen werden jetzt angezeigt, haben aber noch keinen Wert!

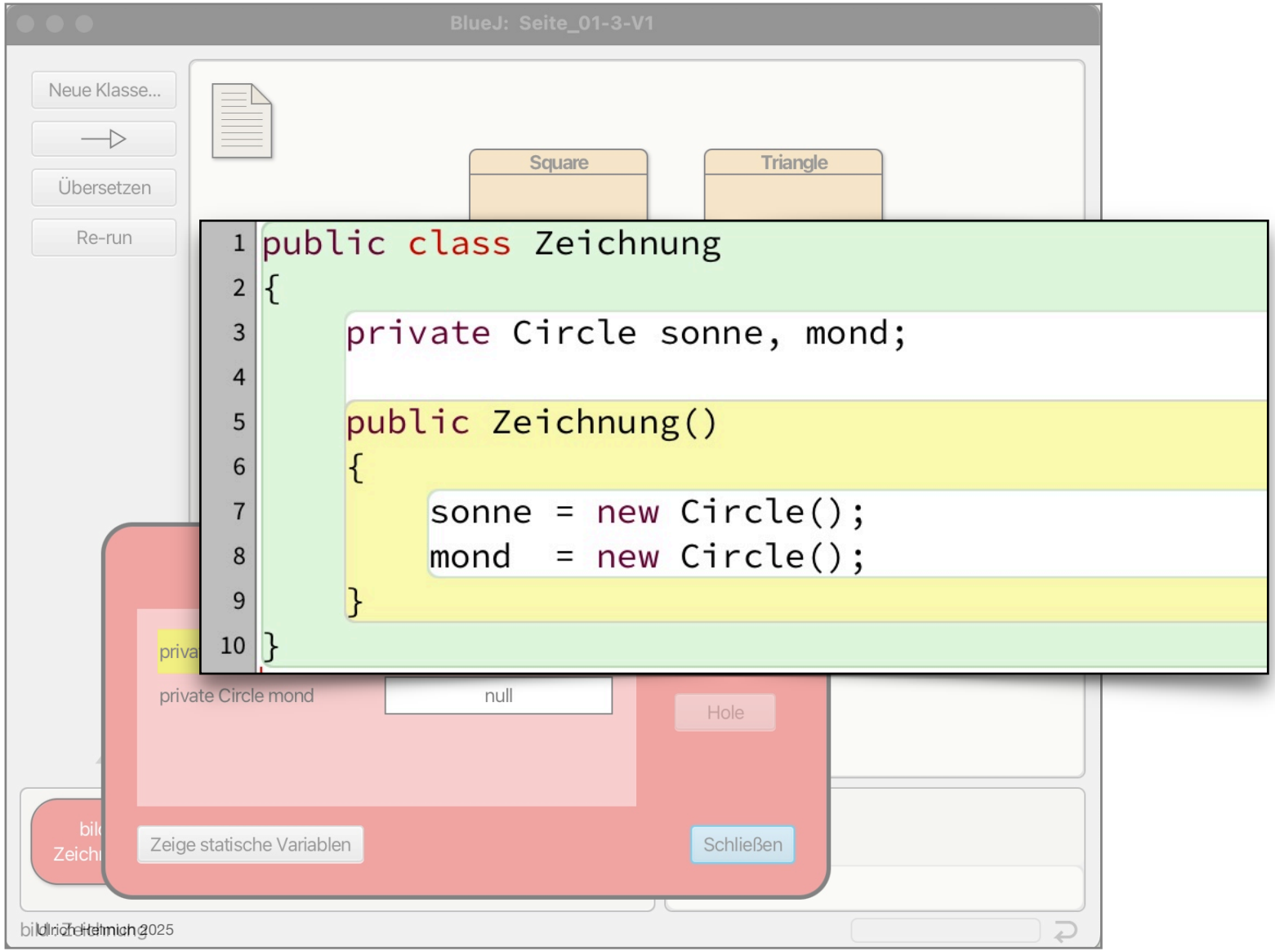
1.3.1 Das Shapes-Projekt, Fortsetzung



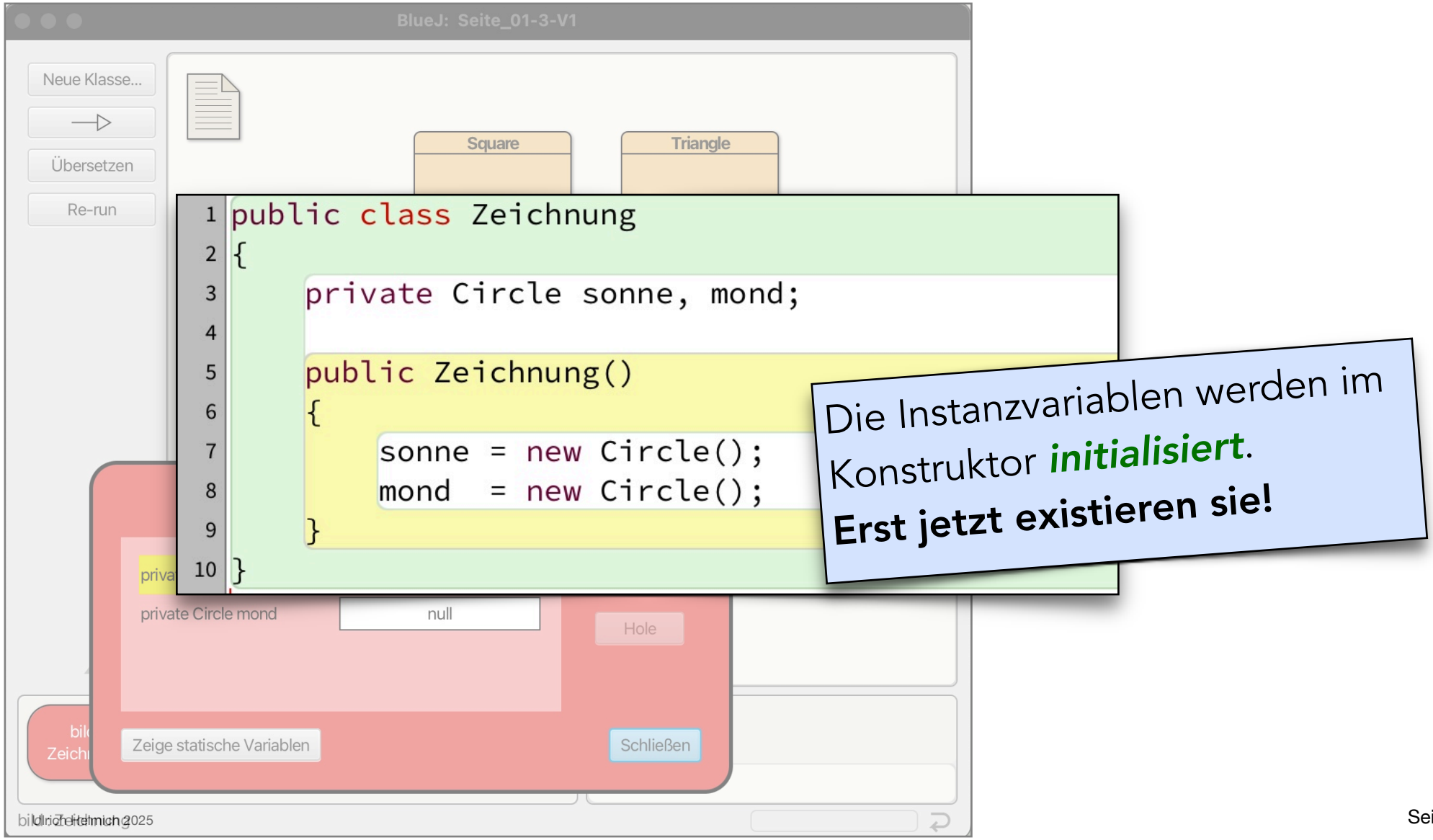
Die Instanzvariablen werden jetzt angezeigt, haben aber noch keinen Wert!

Sie müssen noch initialisiert werden!

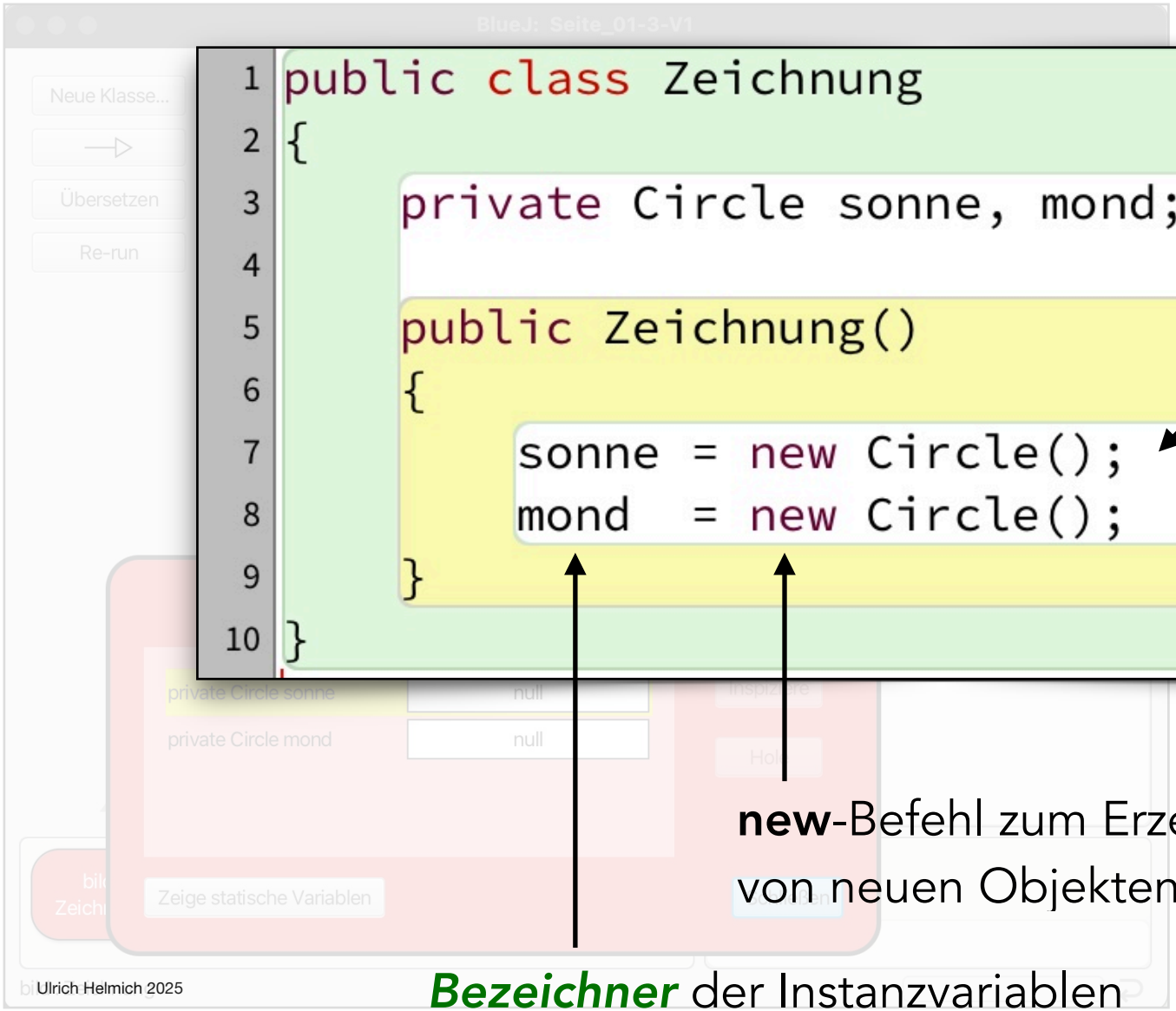
1.3.1 Das Shapes-Projekt, Fortsetzung



1.3.1 Das Shapes-Projekt, Fortsetzung



1.3.1 Das Shapes-Projekt, Fortsetzung



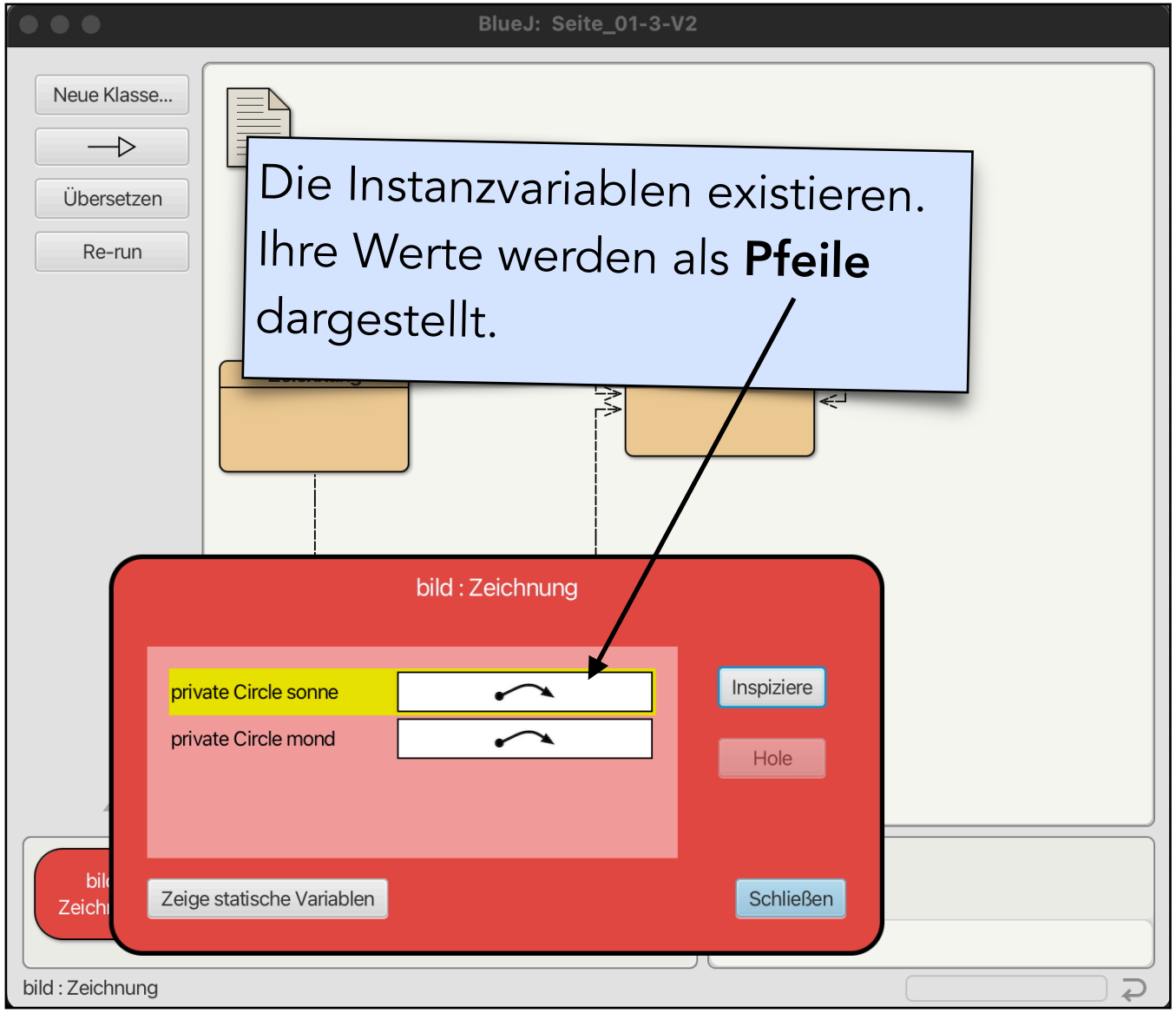

```
1  public class Zeichnung
2  {
3      private Circle sonne, mond;
4
5      public Zeichnung()
6      {
7          sonne = new Circle();
8          mond  = new Circle();
9      }
10 }
```

Angabe der Klasse, von der ein neues Objekt erzeugt werden soll

new-Befehl zum Erzeugen von neuen Objekten

Bezeichner der Instanzvariablen

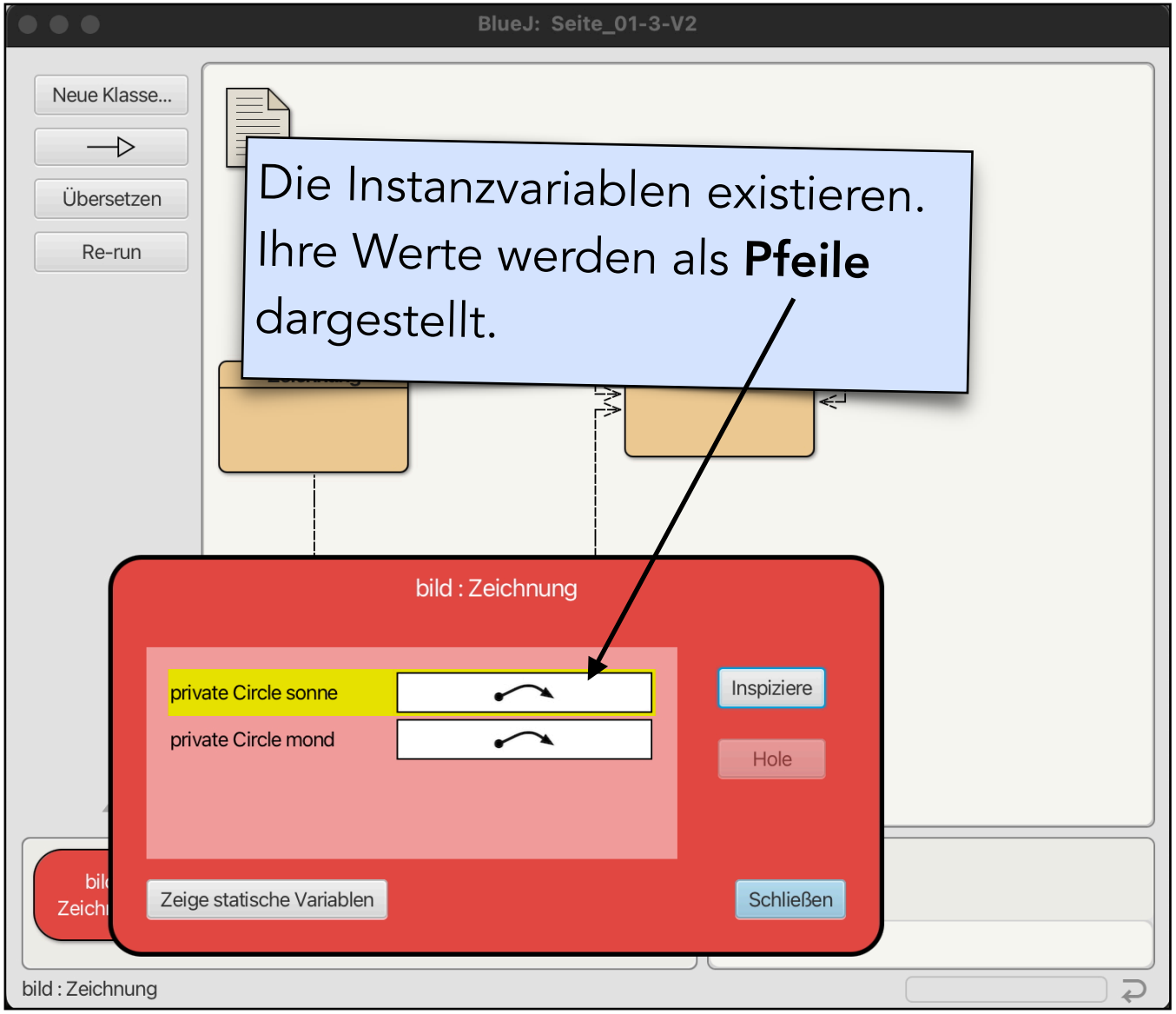
1.3.1 Das Shapes-Projekt, Fortsetzung



Instanzvariablen speichern keine Daten, sondern ***Adressen***!

Sie ***verweisen*** auf einen Bereich im Speicher, an dem sich die *eigentlichen Daten* der Objekte befinden.

1.3.1 Das Shapes-Projekt, Fortsetzung

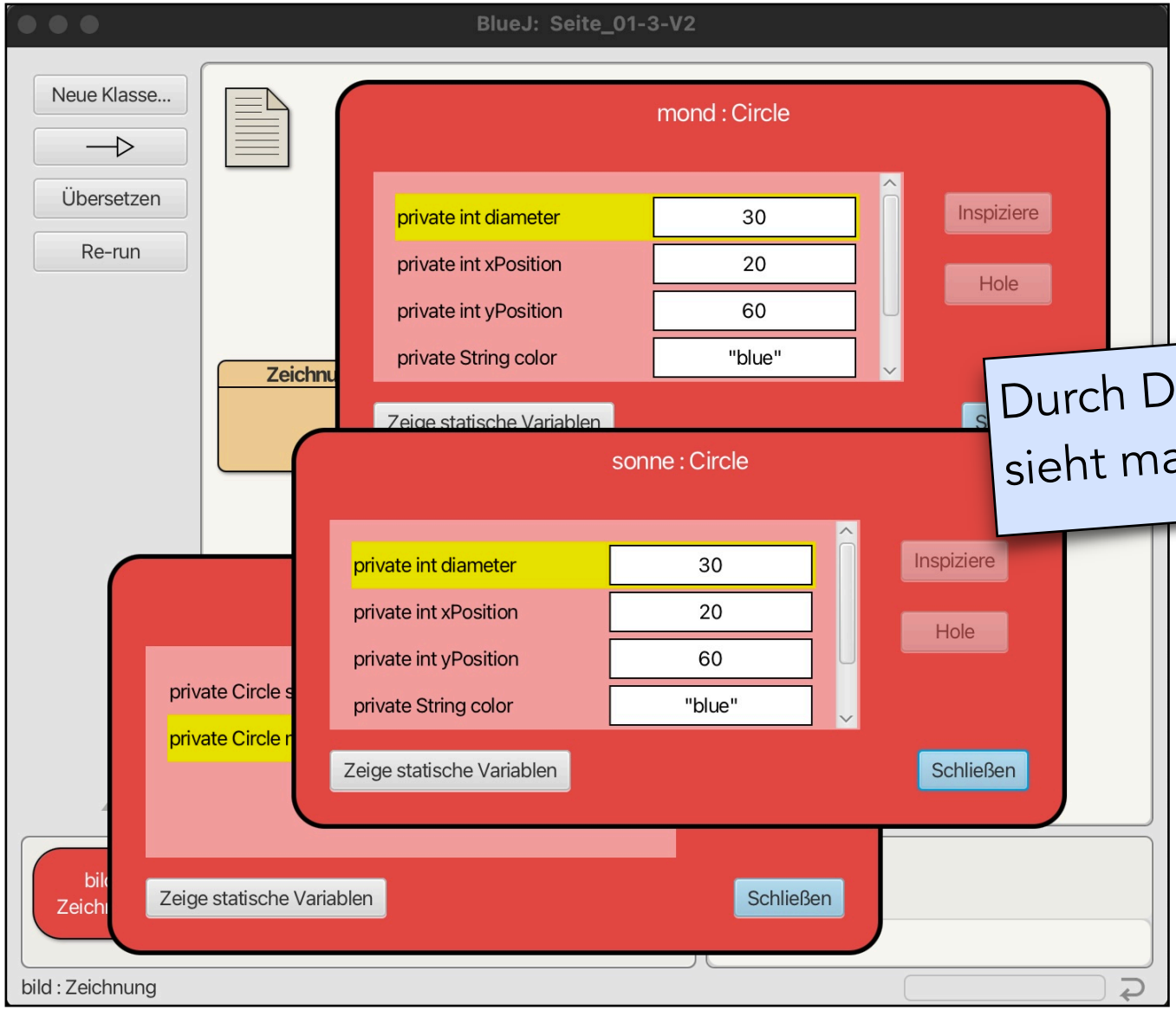


Instanzvariablen speichern keine Daten, sondern ***Adressen***!

Sie ***verweisen*** auf einen Bereich im Speicher, an dem sich die *eigentlichen Daten* der Objekte befinden.

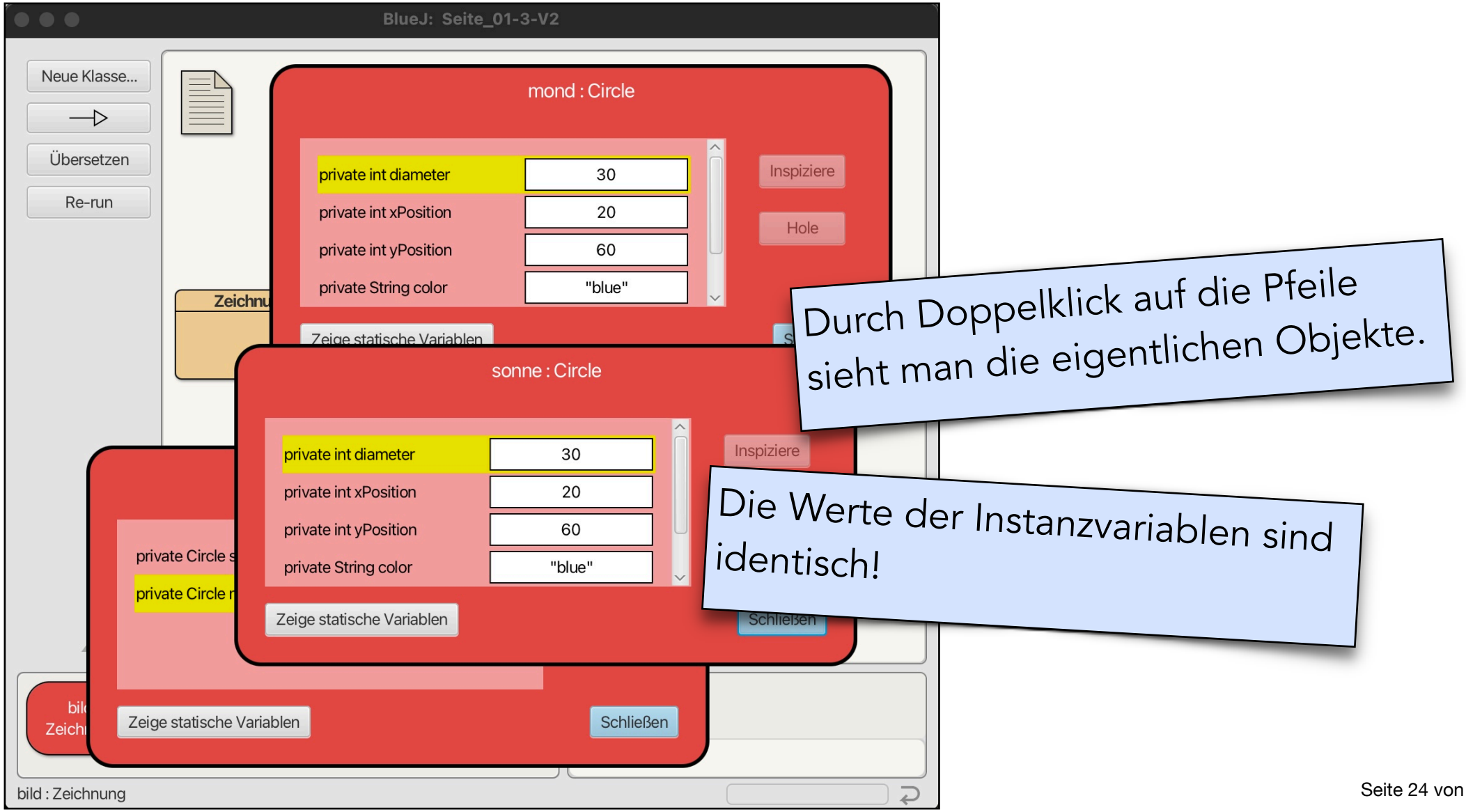
Solche Variablen werden als ***Referenzen*** bezeichnet.

1.3.1 Das Shapes-Projekt, Fortsetzung



Durch Doppelklick auf die Pfeile sieht man die eigentlichen Objekte.

1.3.1 Das Shapes-Projekt, Fortsetzung



1.3.2 Eigene Methoden erstellen

```
1  public class Zeichnung
2  {
3      private Circle sonne, mond;
4
5      public Zeichnung()
6      {
7          sonne = new Circle();
8          mond  = new Circle();
9      }
10
11     public void makeVisible()
12     {
13         sonne.makeVisible();
14         mond.makeVisible();
15     }
16 }
```

Eine eigene Methode
makeVisible()

1.3.2 Eigene Methoden erstellen

Sichtbarkeit der Methode

Rückgabetyp der Methode

Bezeichner der Methode

Parameter der Methode (hier: keine)

```
11      public void makeVisible()
12      {
13          sonne.makeVisible();
14          mond.makeVisible();
15      }
16  }
```

1.3.2 Eigene Methoden erstellen

Sichtbarkeit der Methode

Rückgabetyp der Methode

Bezeichner der Methode

Parameter der Methode (hier: keine)

```
11      public void makeVisible()
12      {
13          sonne.makeVisible();
14          mond.makeVisible();
15      }
16  }
```

Aufruf der Methode **makeVisible()** des Objekts **mond**

Bezeichner des Objekts

1.3.2 Eigene Methoden erstellen

Sichtbarkeit der Methode

Rückgabetyp der Methode

Bezeichner der Methode

Parameter der Methode (hier: keine)

```
11      public void makeVisible()
12      {
13          sonne.makeVisible();
14          mond.makeVisible();
15      }
16  }
```

Die Methoden einer Klasse können öffentliche (public) Methoden von Objekten anderer Klassen aufrufen.

Aufruf der Methode **makeVisible()** des Objekts **mond**

Bezeichner des Objekts

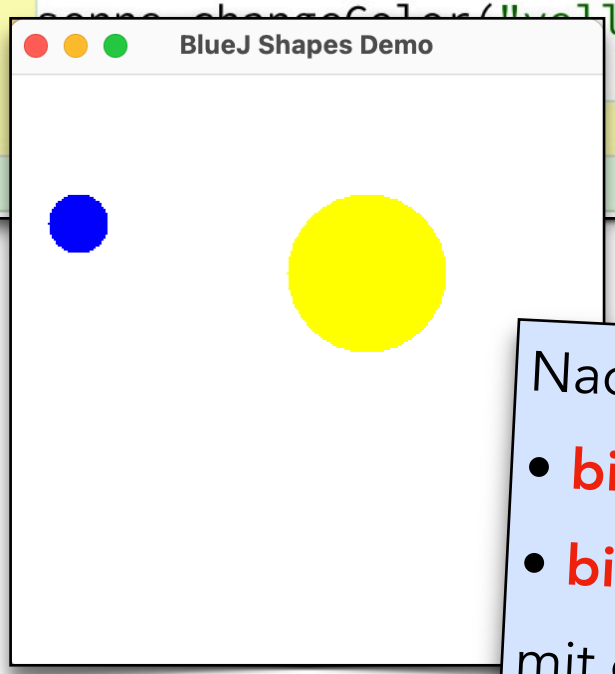
1.3.2 Eigene Methoden erstellen

```
17      public void draw()
18      {
19          sonne.moveHorizontal(120);
20          sonne.changeColor("yellow");
21          sonne.changeSize(80);
22      }
23  }
```

Die neue Methode draw()

1.3.2 Eigene Methoden erstellen

```
17      public void draw()
18      {
19          sonne.moveHorizontal(120);
20          sonne.changeColor("yellow");
21
22      }
23  }
```



Nach Aufrufen der Methoden

- bild.makeVisible()
- bild.draw()

mit der Maus in BlueJ

1.3.2 Eigene Methoden erstellen

```
5     public Zeichnung()
6     {
7         sonne = new Circle();
8         mond  = new Circle();
9         makeVisible();
10        draw();
11    }
```

Erweiterung des Konstruktors der Klasse Zeichnung.

makeVisible() und **draw()** werden jetzt bereits beim Erzeugen eines neuen Objekts der Klasse **Zeichnung** automatisch ausgeführt.

Keine weiteren Maus-Aktionen mehr notwendig!