

Im Skript und auf den Merkblättern, aber auch in anderen Quellen, findet ihr oft sogenannte Methoden-Signaturen. Als Beispiel nehme ich den Abschnitt auf dem Java-Merkblatt zu den Standard-Dialogen:

Die Klasse `JOptionPane` besitzt die Methoden:

```
public static void showMessageDialog(Component parentComponent, Object message)
public static String showInputDialog(Component parentComponent, Object message)
public static int showConfirmDialog(Component parentComponent, Object message)
```

Wie kann man daraus erkennen, wie die Methoden zu benutzen sind?

Zunächst einmal ist es wichtig zu verstehen, welche Informationen in solch einer Methoden-Signatur enthalten sind:

- Ganz wichtig: der Name der Klasse. Hier: `JOptionPane`
- Der Name der Methode. Hier beispielsweise `showMessageDialog()`
- Der Datentyp des Rückgabewertes: Beispielsweise liefert die Methode `showInputDialog()` ein Ergebnis vom Datentyp `String`. Ein Rückgabewert vom Typ `void` bedeutet, dass die Methode nichts zurückliefert.
- Die Parameterliste: Die drei Methode aus dem Beispiel oben erwarten jeweils zwei Parameter:
 1. Einen Parameter vom Datentyp `Component`
 2. Einen Parameter vom Datentyp `Object`
- Ebenfalls wichtig: Handelt es sich um eine „normale“ Methode, oder um eine statische Methode? Das Schlüsselwort `static` vor dem Typ des Rückgabewertes verrät uns, dass es sich um eine statische Methode handelt. Fehlt dieses Schlüsselwort, dann ist es eine „normale“ Methode.
- Die Sichtbarkeit von Methoden auf den Merkblättern ist typischerweise `public`. Manchmal wird das in der Signatur weggelassen (eben weil es eigentlich selbstverständlich ist).

Nun zur Anwendung:

Wenn es sich um eine statische Methode handelt, dann kann man die Methode verwenden, ohne zuvor ein Objekt der Klasse definieren zu müssen. Statt dessen schreibt man den Klassennamen, gefolgt von einem Punkt und dann den Methodennamen hin. Beispielsweise:

```
JOptionPane.showMessageDialog(this, "Dies ist ein Beispiedialog");
```

Schauen wir uns ein weiteres Beispiel für eine Methodensignatur an:

Die Klasse `Random` aus dem Package `java.util` besitzt unter anderem die Methode:

```
public int nextInt(int range)
```

In dieser Methoden-Signatur taucht das Schlüsselwort `static` nicht auf. Es handelt sich also um eine „normale“ Methode. Um sie zu benutzen braucht man ein Objekt der Klasse (hier: `Random`):

```
Random zufall;
int zufallszahl;
zufall = new Random();
zufallszahl = zufall.nextInt(10); // Erzeugt Zufallszahl im Bereich von 0 bis 9
```