

Definitionen zum Lernen

Eine **Klasse** ist eine **Vorschrift zur Konstruktion von Objekten**. Eine Klasse ist immer etwas Abstraktes. Eine Klasse kann nicht angezeigt werden. Dazu müssen erst Objekte der Klasse erzeugt werden.

Beispiel Die Klasse „Kreis“ enthält einen Bauplan eines Kreises- mit diesem können konkrete Kreise angelegt werden.

Konvention Klassennamen werden immer groß geschrieben. (z.B. `Kreis`).

Objekte sind konkrete, aus den Klassen abgeleitete oder **erzeugte Instanzen**, die sich durch einen bestimmten **Zustand** auszeichnen. Aus einer Klasse können mehrere unterscheidbare Objekte gebildet werden.

Beispiel BlueJ benennt Objekte selbst. Es wird der Klassenname verwendet und eine Zahl angehängt. `kreis1`, `kreis2`, `kreis3`... Der Name kann auch selbst gewählt werden.

Konvention Objektamen werden immer klein geschrieben (z.B. `sonne`).

Methoden sind **Operationen**, die entweder den Zustand eines Objektes ändern (**verändernde Methoden**) oder Informationen über den Zustand eines Objektes zurückliefern (**sondierende Methoden**).

Beispiel Ein „Kreis“ enthält eine verändernde Methode `farbeAendern()`. Der Zustand der Eigenschaft `Farbe` kann damit verändert werden.

Konvention Methoden werden klein geschrieben, mehrere Wörter werden zusammen geschrieben. Dabei beginnt jedes Wort in der Mitte mit einem Großbuchstaben (z.B. `farbeAendern()`)

Methoden

Wir können nur mit Objekten kommunizieren (oder Objekte miteinander), indem Methoden aufgerufen werden.

Verändernde Methoden	Sondierende Methoden
Methoden, die Attributwerte verändern, nennt man verändernde Methoden.	Sondierende Methoden verändern den Attributwert nicht. Der Status des Objektes wird nur abgefragt, aber nicht verändert.
Da der Status eines Objektes als Gesamtheit der Attributwerte definiert ist, die ein Objekt hat, bedeutet eine Veränderung eines Attributwertes gleichzeitig auch eine Veränderung des Objektstatus .	Sondierende Methoden liefern den Wert eines Attributes zurück.

Parameter

Methoden können **Parameter** haben, mit denen zusätzliche Informationen für die Erfüllung einer Aufgabe angegeben werden. **Parameter sind Variablen oder Werte, die in der Parameterliste stehen.**

Der Kopf einer Methode wird als **Signatur** bezeichnet. Sie benennt die benötigten Informationen für einen Aufruf der Methode. → `void methodenBezeichner(ParameterTyp parameterBezeichner)`

Stehen in der Signatur einer Methode nur **leere Klammern ()**, so besitzt die Methode **keine Parameter**.