

## Beschreibung der Methoden eines Objekts

dieKaengurus
name = "DIE KAENGURUS" kurs = "11D1" trikotFarbe = "BLAU"
gibNamen() gibKurs() gibTrikotfarbe() setzeNamen() setzeTrikotfarbe()

Die obige Objektkarte zeigt das Objekt **dieKaengurus** inklusive einiger Methoden. Dieses Arbeitsblatt informiert darüber, wie eine Beschreibung der Methoden aussehen kann.

Um funktionieren zu können, benötigen Methoden häufig Daten, die ihnen der Aufrufer der Methode zur Verfügung stellt. Diese Daten werden der Methode als *Parameter* (oder *Argumente*) übergeben.

Methoden können nicht nur Objekte manipulieren (in der Regel das Objekt, zu welchem die Methode gehört), sondern auch Daten an den Aufrufer zurückgeben – dann spricht man vom *Rückgabewert*.

Man könnte sich beispielsweise ein Objekt **rechner** vorstellen, das mathematische Berechnungen ausführt. Die Methode `pi()` des Objekts **rechner** kann bei jedem Aufruf den Wert 3,14159 zurückgeben. Einer Methode wie `quadratwurzel()` müsste jedoch die Zahl, für welche die Quadratwurzelberechnung ausgeführt werden soll, als Parameter mitgegeben werden: `rechner.quadratwurzel(9)` hätte den Rückgabewert 3.

Die Beschreibung einer Methode sollte zumindest eine Übersicht der Parameter und (sofern vorhanden) des Rückgabewerts umfassen. Eine kurze verbale Beschreibung der Arbeitsweise ist äußerst sinnvoll. Zwei Beispiele:

- Methode **gibNamen()**

Parameter: keine

Rückgabewert: Wert des Attributs **name**

Beschreibung: Die Methode liest den im Attribut **name** gespeicherten Wert und gibt diesen zurück.

- Methode **setzeNamen(neuerName)**

Parameter: **neuerName** – der neue Mannschaftsname

Rückgabewert: keiner

Beschreibung: Die Methode weist dem Attribut **name** den übergebenen Wert zu. Der alte Wert wird überschrieben.

In vielen objektorientierten Programmiersprachen (hier am Beispiel von Python) können die Methoden nach folgendem Schema genutzt werden:

<code>print dieKaengurus.gibNamen()</code>	Gibt zu Beginn "DIE KAENGURUS" aus
<code>dieKaengurus.setzeNamen("SKIPPY-TEAM")</code>	Überschreibt den Wert des Attributs <b>name</b> durch "SKIPPY-TEAM"
<code>print dieKaengurus.gibNamen()</code>	Gibt nun "SKIPPY-TEAM" aus

An diesem Beispiel erkennt man gut, dass der Objektbezeichner **dieKaengurus** im Prinzip nichts über den Inhalt des Objekts aussagt. Der Bezeichner könnte genauso gut **meineMannschaft** heissen.

Ein direkter Zugriff auf das Attribut **name** anstelle des „Umwegs“ über Methoden wäre prinzipiell möglich, es ist jedoch „guter Stil“, nicht von außen direkt auf Attribute zuzugreifen, damit sich jedes Objekt eigenständig verwalten kann (Geheimnisprinzip). So lassen sich Teile eines komplexen Programms ändern, ohne dass die Auswirkungen auf andere Programmteile unüberschaubar werden.