Objektorientierung Fachbegriffe

**Klasse:**

Wir alle kennen primitive Datentypen, wie zum Beispiel Integer, Float, Boolean. Unter eine Klasse kann man sich einen „Datentyp on Steroids“ vorstellen. Eine Klasse stellt also umgangssprachlich einen neuen Datentyp, der viel mehr kann als ein primitiver Datentyp, zur Verfügung.

Wir alle haben schon mit Klassen gearbeitet. In Ruby sind z.B. Arrays Klassen, denn sie können viel mehr als primitive Datentypen. Hat man ein Array a, kann man z.B. die Methoden

a.pop() (gibt das letzte Element zurück und entfernt es aus dem Array) oder  
a.length() (gibt die Länge des Arrays zurück)

aufrufen.

**Objekt einer Klasse:**

Eine Klasse stellen wir uns also als „Datentyp on Steroids“ vor – ein Objekt einer Klasse ist einer Variablen zugewiesen und wird in der Regel mit dem Befehl new erzeugt.

x = 4 (primitiver Datentyp, kein New, ganze Zahl 4 ist in der Variable gespeichert)  
a = Array.new() (Ein Objekt der Klasse Array wurde erstellt und in der Variable a gespeichert)

Genau wie bei den primitiven Datentypen, kann man auch mehrere Objekte der selben Klasse haben.

b = Array.new() (Ein zweites Objekt der Klasse Array wurde erstellt und in der Variable b gespeichert)

**Instanzvariablen (Attribute):**

Zusätzliche Funktionalität erfordert zusätzliche Variablen. Instanzvariablen sind Variablen einer Klasse, die für jedes Objekt separat erstellt werden – im obigen Beispiel haben also a und b eigene Instanzvariablen – den Inhalt des Arrays. Im Arbeitsspeicher hat jedes Objekt seinen eigenen Platz für Instanzvariablen.

**Klassenvariablen (statische Attribute):**

Klassenvariablen sind Variablen, die sich alle Objekte der Klasse teilen. Wird so eine Variable von einem Objekt der Klasse verändert, verändert sie sich für alle anderen Objekte der Klasse mit. Im Arbeitsspeicher existiert eine Klassenvariable für alle Objekte nur einmal. Die Klasse Array hat keine Klassenvariablen.

**Kapselung:**

Klassen haben Instanz- und Klassenvariablen. In der Regel sollten diese Variablen von außerhalb der Klasse weder lesbar noch schreibbar sein. Möchte man Instanz- bzw. Klassenvariablen lesbar oder schreibbar machen, geschieht das durch sogenannte SETTER und GETTER. Man spricht in diesem Zusammenhang von Kapselung.

**Methoden:**

Um zusätzliche Funktionalität zur Verfügung zu stellen, werden neben Variablen Methoden verwendet. Eine Methode kann von einem Objekt der Klasse aufgerufen werden und führt dann etwas aus. Man unterscheidet zwischen **mutierenden Methoden** – das sind Methoden, die das Objekt selbst verändern und **nicht mutierenden Methoden**, bei denen das Objekt nicht verändert wird. Nehmen wir an, dass in der Variable a ein Objekt der Klasse Array gespeichert ist, dann ist

a.pop() eine mutierende Methode, da das Array a im Anschluss kürzer ist  
a.length() eine nicht-mutierende Methode, da sich das Array a nicht verändert hat.

**Konstruktor:**

Der Konstruktor ist eine besondere Methode. Sie wird beim Erstellen eines Objekts aufgerufen. Der Konstruktor der Klasse Array wird durch die Zuweisung a = Array.new() aufgerufen.