



## Wiederholung: Statische Methoden und statische Attribute

Normalerweise sind die Attribute, die in einer Klasse definiert werden, dazu da, den Zustand einer Objektinstanz dieser Klasse zu beschreiben. Sie können nur im Kontext eines Objektes angesprochen werden, also nachdem das Objekt mit `new` erzeugt wurde.

Klassendefinition		Verwendung
<pre>public class A {     public int i; }</pre>	→	<pre>A a = new A(); a.i = 17;</pre>

Gleiches gilt für Methoden, die in einer Klasse definiert werden. Auch ihr Aufruf ist erst nach der Anlage des Objekts möglich. Innerhalb dieser Methoden kann dann auf die Objekteigenschaften zugegriffen werden.

Klassendefinition		Verwendung
<pre>public class A {     public int i;     public void incI()         { i = i + 1; } }</pre>	→	<pre>A a = new A(); a.i = 17; a.incI();</pre>

Manchmal wäre es hilfreich, wenn ein Attribut der Klasse unabhängig von Instanzen existieren könnte, insbesondere auch, wenn noch keine Instanz angelegt ist. Auch Methoden können sinnvoll sein, die unabhängig von Objektinstanzen ausgeführt werden. Sie können dann natürlich nicht auf Objekteigenschaften zugreifen – es könnte ja mehrere Objekte parallel geben oder evtl. auch gar keines. Solche Attribute und Methoden sind durch Voranstellen des Schlüsselworts `static` möglich:

Klassendefinition		Verwendung
<pre>public class A {     public static int s;     public int i;     public void static incS()         { s = s + 1; }     public void incI()         { i = i + s; } }</pre>	→	<pre>A.s = 2; // Zweierschritte A a1 = new A(); A a2 = new A(); a1.i = 17; a2.i = 0; a1.incI(); // i in a1 ist jetzt 19 a2.incI(); // i in a2 ist jetzt 2 A.incS(); // Dreierschritte a1.incI(); // i in a1 ist jetzt 22 a2.incI(); // i in a2 ist jetzt 5</pre>

Statische Methoden (auch: Klassenmethoden) und statische Attribute (auch: Klassenattribute) werden direkt über den Klassennamen angesprochen. Der Wert eines statischen Attributs ist in allen Instanzen gleich, da es nur einmal existiert.

## Aufgabe „exercise03“ (Abgabe vom 12.4. - 18.04.2016, bZv-relevant)

### Singleton

Eine Klasse wird als Singleton bezeichnet, wenn es maximal eine Instanz zu dieser Klasse geben darf. Diese Anforderung tritt in der Praxis häufig auf, wenn z.B. eine nur einmal vorhandene Ressource (Beispiel: Standarddrucker) als Objekt abgebildet werden soll.

In Java kann normalerweise nicht verhindert werden, dass eine Klasse mehrfach instanziiert wird. Man kann aber mit Hilfe eines Tricks dafür sorgen, dass dies nicht passiert: Wird in dieser Klasse nur ein Konstruktor implementiert und dieser auf `private` gesetzt, kann niemand außerhalb der Klasse eine Instanz bilden. Weiterhin aufgerufen werden können aber statische Methoden dieser Klasse, mit deren Hilfe die eine zulässige Instanz angelegt werden kann.

Im Verzeichnis Übungsaufgabe\_3 finden Sie eine unfertig implementierte Klasse `Singleton`. Das bereits vorhandene statische Attribut `mySingleton` soll später die eine zulässige Instanz aufnehmen.

### Ihre Aufgabe:

- a) Erweitern Sie die Klasse `Singleton` so, dass sie lediglich einen privaten Standardkonstruktor ohne Parameter besitzt. Der Rumpf des Konstruktors verbleibt leer.
- b) Ergänzen Sie eine öffentliche (`public`) Klassenmethode `getInstance()`, die
  - überprüft, ob im Attribut `mySingleton` bereits eine Instanz enthalten ist. Falls dies so ist, wird diese Instanz zurückgegeben und der Aufruf beendet
  - andernfalls wird mit Hilfe des Standardkonstruktors eine neue Instanz erzeugt, diese Instanz im Attribut `mySingleton` gespeichert und die neue Instanz zurückgegeben.

Die in der Klasse enthaltene `main`-Methode ist ein erster Anhaltspunkt, ob Ihre Implementierung die Anforderungen erfüllt.

### Hinweis:

Aufgrund fehlender Methoden ist die bereitgestellte Klasse `Singleton` zunächst nicht übersetzbar.