

Konstruktoren der Klasse	
①	<code>public class DemoClass {</code>
②	<code>private Datentyp attr;</code>
③	<code>:</code>
④	<code>public Datentyp methode1(){...};</code>
⑤	<code>:</code>
⑥	<code>/* Konstruktor(en) Variante 1 Überladung */</code>
⑦	<code>public DemoClass(){attr=Wert;}</code>
⑧	<code>public DemoClass(Typ Var){</code>
⑨	<code>this.attr= Var ; }</code>
⑩	<code>}</code>
DemoClass.java	

Der Konstruktor ist eine spezielle Klasse- Methode. Er wird bei der Objektinstanziierung (mit dem **new**-Operator) aufgerufen und dient zur Initialisierung der Attribute.

Hinweise:

- Konstruktor haben immer den Klassennamen, sollte **public** sein und haben keine Rückgabe.
- Wird **kein** Konstruktor implementiert dann wird automatisch der parameterlose leere Default-Konstruktor erzeugt.
- Der Konstruktor wird **genau einmal** bei der Objekterzeugung aufgerufen.
- Der Konstruktor kann auch (nach den bekannten Regeln) überladen werden.

DemoClass obj1 = new DemoClass(); Hier wird der parameterlose Konstruktor ⑦ aufgerufen	DemoClass obj2 = new DemoClass(Wert); Und so der parametrisierte Konstruktor in ⑧
---	--

## Eine besondere Möglichkeit bietet die Konstruktor Weiterleitung (Delegieren).

Dabei sind folgende Regeln zu beachten:

- Die Weiterleitung erfolgt mit der this()-Methode und muss die erste Anweisung im Konstruktor sein.
- jeder Konstruktor wird höchstens einmal ausgeführt
- Konstruktor-Aufrufe dürfen keinen Zyklus bilden
- jeder Konstruktor darf höchstens an einen Konstruktor delegieren

Für das nebenstehende Beispiel mit der Weiterleitung zeigt die Tabelle die verschiedenen Möglichkeiten der Objekterzeugung.

Konstruktoren der Klasse	
①	<code>public class DemoClass {</code>
②	<code>private Datentyp attr1;</code>
③	<code>private Datentyp attr2;</code>
	<code>:</code>
	<code>/* Konstruktor(en) Variante 2 Weiterleitung */</code>
④	<code>public DemoClass(){</code>
⑤	<code>this( 0 );</code>
	<code>/* mögliche weitere Anweisungen */ }</code>
⑥	<code>public DemoClass(Typ Var){</code>
⑦	<code>this(Var , 0 );</code>
	<code>/* mögliche weitere Anweisungen */ }</code>
⑧	<code>public DemoClass(Typ Var<sub>1</sub>, Typ Var<sub>2</sub>){</code>
⑨	<code>this.attr1 = Var<sub>1</sub>;</code>
⑩	<code>this.attr2 = Var<sub>2</sub> ; }</code>
⑪	<code>}</code>
DemoClass.java	

	Konstruktor Aufruf	Reihenfolge	erzeugte Obj.
①	DemoClass obj1 = new DemoClass();	④ -> ⑥ -> ⑧ DemoClass() -> this(0) -> this(0,0)	objekt1 attribut1    attribut2 0            0
②	DemoClass obj2 = new DemoClass(2);	⑥ -> ⑧ DemoClass(2) -> this(2,0)	objekt2 attribut1    attribut2 2            0
③	DemoClass obj3 = new DemoClass(3,4);	⑧ DemoClass(3,4)	objekt3 attribut1    attribut2 3            4