

Der Java-Konstruktor

In einer Klasse ist der Konstruktor eine Methode, die den gleichen Namen wie die Klasse hat. Du benötigst diesen, um ein neues Objekt einer Klasse zu erstellen. Das ist auch der Grund, weshalb jede Klasse einen solchen Konstruktor haben muss. Programmierst du diesen nicht selbst, gibt der Compiler von Java einen sogenannten Standardkonstruktor vor, der im Hintergrund eines Programms abläuft und keinen Inhalt hat. Wir haben für dich extra Videos auch zu Klassen und Methoden erstellt und damit du das Thema komplett verstehen kannst. Ein Standardkonstruktor einer Klasse Tier würde in Java so aussehen. Zusätzlich haben wir noch die zwei Attribute Name und Art programmiert. Diese brauchen wir später. Der Standardkonstruktor besteht zunächst aus einem Modifizier. In unserem Fall ist das public. Public regelt, dass du auch von außerhalb der Klasse auf die Methode zugreifen kannst. Was du ja zum Erzeugen eines Objekts der Klasse Tier machen musst. Den Vorgang kannst du auch als Instanziierung bezeichnen. Eine Instanziierung unserer Klasse Tier in der Main-Methode programmierst und nun mit dem Schlüsselwort new und anschließend kommt in unserem Fall hier der Aufruf des leeren Konstruktors Tier. Prima! Jetzt hast du schon dein erstes Objekt erzeugt, welches du in der Variablen Hund vom Typ Tier speichern kannst. Nun kannst du den Konstruktor auch Werte mitgeben, die du am Anfang schon kennst, dafür musst du die Objekterzeugung und den Konstruktor aber noch um die Möglichkeit von Parametern erweitern. Nehmen wir an, du möchtest jetzt eine spezielle Tierart programmieren und du weißt, dass die Art Vögel sein wird. Dann kannst du die Erzeugung des Objektes in der Main Klasse anpassen, indem du bei der Erzeugung eines Tier-Objekts einfach den String Vögel mit übergibst. Das Programm sucht daraufhin nach einem passenden Konstruktor in der Klasse Tier. **Dieser muss Tier heißen und die Möglichkeit besitzen einen String Parameter zu empfangen, damit das funktioniert musst du deinen Standardkonstruktor anpassen. Das sieht dann konkret so aus, dass du ihm einen Parameter vom Typ String gibst. Diese nennen wir hier Art. Mit dem Befehl this. A = a** zockst du lediglich dafür, dass die Variable a der Klasse von der zweiten Zeile gleich dem übergebenen Parameter Art ist. Als nächstes wollen wir uns einmal gemeinsam das überladen von Konstruktoren genauer anschauen. Wie normale Methoden kannst du auch diese überladen. Dabei spielt die Signatur welche sich aus dem Namen der Methode und den Parameter zusammensetzt eine wichtige Rolle. Angenommen du möchtest ein Objekt Vögel und ein Objekt Säugetiere programmieren. Weiterhin gehen wir davon aus, dass du den Namen des Säugetiers schon kennst. es soll ein Mensch sein. Von dem Vogel hingegen weißt du keinen Namen weshalb die entsprechende Variable auf unbekannt gesetzt werden soll. Wenn du nun in der main-Methode zweimal die Klasse Tier aufrufst einmal mit nur einem zu übergebenen Parameter und einmal mit zwei dann sucht das Programm nach zwei verschiedenen Konstruktoren. Zwei heißen beide gleich können aber nicht die gleichen Werte empfangen. Der erste Aufruf von Tier landet in dem ersten Konstruktor. Der 2 Aufruf mit zwei möglichen Parametern führt zu den Konstruktor 2. Dabei spricht man vom überladen von Konstruktoren. Jetzt hast du zwei verschiedene Konstruktoren, aber eigentlich machen doch beide das gleiche oder? Sie weisen den Attributen der Klasse Name und Art einen unterschiedlichen Wert zu und doppelten Programmcode zu vermeiden kannst du die zwei Konstruktoren miteinander verketteten. Das würde mit dem Operator this () funktionieren. Umgesetzt könnte das dann so aussehen, dass du im ersten Konstruktor das Schlüsselwort this verwendest. Übergibst du bei der Klassen Instanziierung aus der Main Klasse nur einen Parameter, so landest du im Konstruktor eins. Dieser Verweis dann durch Wyss weiter auf den Konstruktor 2 und ruft ihn mit der Variablen Art und dem Spring Wert unbekannt auf. Dann werden die Variablen Art und Name mit ihren jeweiligen Werten belegt. Merken musst

du dir das der `this ()` Befehl immer an erster Stelle im Konstruktor kommen muss. Nun kannst du nicht nur Objekte einer Klasse erzeugen, sondern weißt auch wie man Konstruktoren anpasst und miteinander verkettet. Viel Spaß beim weiteren Programmieren.