

Java Standard: Variablen und Bezeichner

Aus Wikibooks



[Zurück zu Datenstrukturen](#) |



[Hoch zu Java Inhaltsverzeichnis](#) |



[Vor zu Operatoren](#)

Inhaltsverzeichnis

- 1 Bezeichner
 - 1.1 Java Code Konventionen
 - 1.1.1 Klassen und Interfaces
 - 1.1.2 Methoden
 - 1.1.3 Variablen
 - 1.1.4 Konstanten
 - 1.2 Variablendeklaration
 - 1.2.1 Sichtbarkeits- und Zugriffsmodifizierer
 - 1.3 Die Initialisierung

Bezeichner

Bezeichner (engl.: Identifier) bezeichnen in Java Klassen, Interfaces, Variablen (bzw. Objekte), Methoden und Konstanten. Bezeichner sind also, einfach gesprochen die Namen von den zuvor genannten Elementen.

Mit Hilfe von Bezeichnern kann man die diesen Namen zugeordneten Speicherbereiche ansprechen und nutzen.

Legale Identifier können in Java aus folgenden Zeichen bestehen:

- Unicode Zeichen
- Nummern
- Währungssymbole
- Verbindungszeichen (Bsp.: Unterstrich)

Gültige Identifier:

- starten mit einem Zeichen, Währungssymbol oder Verbindungszeichen
- starten **nicht** mit einer Nummer
- können nach dem ersten Zeichen alle Kombinationen von Zeichen, Nummern, Währungssymbolen und Verbindungszeichen enthalten
- theoretisch ist die Anzahl der Zeichen pro Identifier unbegrenzt
- sind keine Java Schlüsselwörter wie **new**
- Identifier sind case-sensitive, sprich Java unterscheidet zwischen Groß- und Kleinschreibung

Java Code Konventionen

Ähnlich wie in der "realen Welt" sollte man sich bei der Programmierung ebenfalls um einen guten Ton bemühen.

```
esisteinunterschiedobichalleskleinundzusammenschreibe ODER
```

```
_leserlich_formatiere_sodass_mein_Code_besser_nachvollzogen_werden_kann
```

Grundlegende Regel hierbei ist die Verständlichkeit und leichte Nachvollziehbarkeit des verfassten Codes (Und die Kommentare nicht vergessen). Aus diesem Grund wurden die Java Code Konventionen geschaffen:

Klassen und Interfaces

- Erster Buchstabe immer groß
- "CamelCase" anwenden. Wenn mehrere Worte in einem Identifier verbunden werden, sollten die Anfangsbuchstaben groß geschrieben werden.
- Für Klassen Nomen verwenden (Hund, Katze...)
- Für Interfaces Adjektive verwenden (Essbar...)

Methoden

- erster Buchstabe immer klein.
- "camelCase" anwenden.
- Methodennamen sollten Kombinationen aus Verben und Nomen darstellen. So kann am besten beschrieben werden, wozu die Methode dient. Gute Beispiele findet man natürlich in der Java API.
- Bsp.: getTasche(), fuelleTasche(...)

Variablen

- wie bei Methoden "camelCase" und erster Buchstabe immer klein.

Konstanten

- Konstanten werden erzeugt, indem man Variablen static und final deklariert.
- Sie werden nur in Großbuchstaben geschrieben und einzelne Worte werden durch Unterstrich voneinander getrennt.
- Bsp.: MAX_MENGE

Variablendeklaration

Variablen in Java sind immer lokale Variablen. Bevor man eine Variable benutzen kann, muss sie erst deklariert werden. Dies kann an beliebiger Stelle innerhalb einer Methode, in der statischen Initialisierung einer Klasse oder im Initialisierungsteil einer for-Schleife geschehen. Nur in diesem Bereich ist die Variable sichtbar. Der Name der Variablen muss innerhalb dieses Sichtbarkeitsbereichs eindeutig sein.

Da Java eine streng typisierte Sprache ist, muss bei Deklaration einer Variablen immer ein Datentyp mit angegeben werden. Dies kann ein primitiver Datentyp sein (boolean, char, byte, short, int, long, float, double) oder ein Referenztyp (eine Klasse oder Interface).

Optional kann eine Variable bei Deklaration durch ein Ausdruck zu Laufzeit mit einem Wert initialisiert werden.

Mit dem ebenfalls optionalen Schlüsselwort `final` wird verhindert, dass der Wert einer Variablen nach dessen Deklaration geändert werden kann; lediglich eine erstmalige Zuweisung im Konstruktor einer Klasse für Instanzvariablen ist möglich.

Die Syntax einer Variablendeklaration:

```
[ final ] [private|protected|public] Datentyp VARIABLENNAME [ = Ausdruck ] ;
```

zum Beispiel:

```
public class Beispiel {
    static {
        int i = 0;
        // etwas Sinnvolles mit i anstellen
    }
    public void machWas() {
        long grosseGanzeZahl;
        String name;
        final float PI = 3.1415;
        boolean [] wahrheitswerte = new boolean[10];
        // etwas Sinnvolles mit grosseGanzeZahl, name und PI anstellen
        // Die Zuweisung PI = 1; verursacht einen Compilerfehler
        // Ein Zugriff auf i verursacht einen Compilerfehler
        for (int i=0; i < wahrheitswerte.length; i++) {
            // hier existiert i (ein anderes als obiges im static Block)
        }
    }
}
```

Sichtbarkeits- und Zugriffsmodifizierer

Die drei Schlüsselwörter `private`, `protected` und `public` dienen der Modifikation der Sichtbarkeit und des Zugriffs.

- `private` erlaubt nur der eigenen Klasse den Zugriff.
- `protected` erlaubt der eigenen Klasse, der Paketklasse und der Elternklasse den Zugriff.
- `public` erlaubt jeder Klasse, auch fremden Klassen, den Zugriff.

Ist eine Methode als `private static final` gekennzeichnet, ist der Zugriff am schnellsten.

Um Fehlern vorzubeugen, sollte der Zugriff so stark wie möglich eingeschränkt werden.

Die Initialisierung

Vor der Verwendung der Variablen muss ihnen ein Wert zugewiesen werden, der dem Datentyp entspricht. So kann man einem Integerwert(`int`) nicht den String "Hallo" zuweisen (zumindest nicht ohne vorherige Umwandlung)

```
int i;
i = 12; //i muss vorher deklariert werden
```

oder:

```
int i = 12;
```



[Zurück zu Datenstrukturen](#) |



[Hoch zu Java Inhaltsverzeichnis](#) |



[Vor zu Operatoren](#)

Von „https://de.wikibooks.org/w/index.php?title=Java_Standard:_Variablen_und_Bezeichner&oldid=732436“

- Diese Seite wurde zuletzt am 19. Februar 2015 um 12:39 Uhr geändert.
- Der Text ist unter der Lizenz "Creative Commons" „Namensnennung – Weitergabe unter gleichen Bedingungen“ verfügbar. Zusätzliche Bedingungen können gelten. Einzelheiten sind in den Nutzungsbedingungen beschrieben.