

Code-Konventionen bei der Programmierung in Java

Im folgenden sollen einige Konventionen bei der Erstellung von Java-Programmen vorgestellt werden. Die offiziellen *Code Conventions for the Java Programming Language* sind zu finden unter:

<http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>

Code-Konventionen sind für Programmierer aus verschiedenen Gründen wichtig:

- Die Verwaltung einer Software nimmt auf lange Sicht ca. 80% des Zeitaufwandes in Anspruch.
- Fast keine Software wird während ihres gesamten Bestehens vom Ersteller verwaltet.
- Code-Konventionen verbessern die Lesbarkeit von Software und erlauben neuen Code schneller und besser zu verstehen.

Layout

- **Einrücken**
Rücken Sie den Code für jeden neuen Block um 4 Leerzeichen ein.
- **Zeilenlänge**
Vermeiden Sie Zeilen, die länger als 80 Zeichen sind, da sie von vielen Terminals und Tools nicht vernünftig verarbeitet werden.
- **Zeilenumbrüche**
Wenn ein Ausdruck nicht auf eine einzelne Zeile paßt, brechen Sie sie entsprechend den folgenden Grundregeln um:
 - Umbruch nach einem Komma
 - Umbruch vor einem Operator
 - Richten Sie die neue Zeile mit dem Anfang des Ausdruckes auf der gleichen Stufe der vorhergehenden Zeile aus.

Hier einige Beispiele:

```
someMethod(longExpression1, longExpression2, longExpression3,
           longExpression4, longExpression5);

var = someMethod1(longExpression1,
                  someMethod2(longExpression2,
                              longExpression3));
```

Zeilenumbrüche in if-Statements sollten generell 8 Leerzeichen eingerückt werden, da 4 Leerzeichen die Erkennung des folgenden Blocks erschweren. Zum Beispiel:

```
//DON'T USE THIS INDENTATION
if ((condition1 && condition2)
    || (condition3 && condition4)
    ||!(condition5 && condition6)) { //BAD WRAPS
    doSomethingAboutIt();           //MAKE THIS LINE EASY TO MISS
}

//USE THIS INDENTATION INSTEAD
if ((condition1 && condition2)
    || (condition3 && condition4)
    ||!(condition5 && condition6)) {
    doSomethingAboutIt();
}
```

Deklarationen

- **Variablennamen**

Wählen Sie Variablennamen mit semantischer Bedeutung und vermeiden Sie Variablen, die aus nur einem Zeichen bestehen. Verwenden sie pro Deklaration eine Zeile:

```
int level; // indentation level
int size;  // size of table
```

- **Initialisierung**

Versuchen Sie lokale Variablen bei der Deklaration zu initialisieren. Dies ist nur dann nicht möglich, wenn der Initialisierungswert erst berechnet werden muss.

- **Plazierung**

Plazieren Sie Deklarationen immer am Beginn eines Blocks.

```

void myMethod() {
    int int1 = 0;          // beginning of method block

    if (condition) {
        int int2 = 0;    // beginning of "if" block
        ...
    }
}

```

Statements

- Jede Zeile sollte höchstens ein Statement enthalten:

```

argv++;           // Correct
argc--;          // Correct
argv++; argc--;  // AVOID!

```

- **if, if-else Statements**

```

if (condition) {
    statements;
}

if (condition) {
    statements;
} else if (condition) {
    statements;
} else {
    statements;
}

```

- **for Statements**

```

for (initialization; condition; update) {
    statements;
}

```

- **while Statements**

```

while (condition) {
    statements;
}

```

- **do-while Statements**

```
do {
    statements;
} while (condition);
```

- **switch Statements**

```
switch (condition) {
case ABC:
    statements;
    /* falls through */

case DEF:
    statements;
    break;

case XYZ:
    statements;
    break;

default:
    statements;
    break;
}
```

Namenskonventionen

Durch Namenskonventionen werden Programme leichter lesbar und damit leichter verständlich. Folgende Namenskonventionen sollten eingehalten werden:

Typ	Namenskonvention	Beispiel
Klassen	Klassennamen sollten Substantive sein, wobei der erste Buchstabe jedes internen Wortes gross geschrieben wird.	<code>class Raster;</code> <code>class ImageSprite;</code>
Schnittstellen	wie Klassen	<code>interface RasterDelegate;</code> <code>interface Storing;</code>
Methoden	Methoden sollten Verben sein, der erste Buchstabe klein, interne Worte gross geschrieben	<code>run();</code> <code>runFast();</code> <code>getBackground();</code>
Variablen	wie Methoden	<code>int i;</code> <code>char c;</code>
Konstanten	Variablenamen sollten kurz aber aussagekräftig sein. bestehen ausschliesslich aus Grossbuchstaben, einzelne Worte durch '_' getrennt	<code>float myWidth;</code> <code>static final int MIN_WIDTH = 4;</code> <code>static final int MAX_WIDTH = 999;</code>