

## 1 Konventionen zur Quellcodeerstellung

1. Klassen-, Methoden- und Variablennamen sind in *Englisch* zu wählen.
2. Javadoc-Kommentare sind möglichst in *Englisch* zu verfassen; deutsch wird in Ausnahmefällen akzeptiert (deutsche Kommentare sind für 95% der Entwickler nutzlos).
3. Öffnende geschweifte Klammern werden *nicht* in eine neue Zeile gesetzt (siehe Beispiel).
4. Die Einrücktiefe entspricht vier Leerzeichen bei normaler Einrückung, acht Leerzeichen bei Einrückung zum Fortsetzen einer Zeile.
5. Falls eine Zeile umgebrochen werden muss, geschieht dies nach einem Operator oder nach einer öffnenden Klammer.
6. Vor einer öffnenden Klammer befindet sich bei Methoden kein Leerzeichen.
7. Nach einer öffnenden und vor einer schließenden Klammer befinden sich ebenfalls keine Leerzeichen.
8. Zur deutlichen Abgrenzung zu Methoden, wird nach **if**-, **switch**-, **catch**-, **while**-, **for**- und **return**-Statements die öffnende Klammer mit einem Leerzeichen abgetrennt.
9. Das **else**-Statement steht in der selben Zeile wie die vorangehende schließende geschweifte Klammer (abgetrennt durch ein Leerzeichen).
10. Binäre und der ternäre Operator(en) werden durch Leerzeichen abgetrennt (Ausnahme: Erhöhung oder Verminderung um eine Konstante).
11. Nach Kommata und Semikola ist jeweils ein Leerzeichen einzufügen.
12. Längerer Quellcode ist in mehrere logisch zusammenhängende Blöcke zu unterteilen die jeweils durch eine Leerzeile getrennt sind. Dies schließt auch die erste Zeile einer Klassen- oder Methodendefinition ein.
13. Nach dem Casting-Operator wird der Operand direkt ohne Leerzeichen angeschlossen.

Des Weiteren gelten die Regeln zur Quellcodeerstellung aus:

**Achut Reddy, Java™ Coding Style Guide, Sun Microsystems, Inc., 1998**  
(siehe Literatur-Ordner im cvs-Repository).

## 2 Beispiel

```
public class DerivedClass extends BaseClass implements InterfaceOne,
    InterfaceTwo, InterfaceThree {

    int memberOne = 1;
    double memberTwo = 3.14159d;

    public static int staticMethod(int intParam) {

        for (int i = 0; i < intParam+1; ++i) {
            if (intParam < 10) {
                memberOne *= memberTwo;
            } else {
                memberOne *= (memberTwo / 2.0d);
            }
        }
        return (memberOne + memberTwo);
    }

    public int memberOne() {

        return memberOne;
    }
}
```