



|   |  |
|---|--|
| <a href="http://ovm-kassel.de">http://ovm-kassel.de</a>   Information |  |
| Information IT-AE-JA-INFO-3.5<br>Quellcode dokumentieren              |   |
| Code  | IT-AE-JA-INFO-3.5  |
| Autor   | André Bauer<br><a(dot)bauer(at)ovm-kassel(dot)de>  |
| Datum   | 25. Februar 2018   |
| Links   | <ul style="list-style-type: none"> <li>• <a href="#">Wikipedia: Javadoc</a></li> <li>• <a href="#">How to Write Doc Comments for the Javadoc Tool</a></li> <li>• <a href="#">Screencast: Die API-Dokumentation und Javadoc</a></li> <li>• <a href="#">Google Java Style Guide</a></li> <li>• <a href="#">Java API Dokumentation</a></li> </ul> |
| Verwandte Literatur   |  |
| Lizenz  | <br>Dieses Werk ist lizenziert unter einer <a href="#">Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz</a> .   |

## Quellcode dokumentieren

Quellcode sollte so verfasst werden, dass er gut lesbar ist. Dazu sollten die Vereinbarungen, wie in einer Programmiersprache Quellcode gestaltet werden soll, eingehalten werden wie der [Google Java Style Guide](#). Zu diesen Vereinbarungen zählen u. a. die korrekte Einrückung der Zeilen und die Verwendung gut gewählter Namen für Klassen, Variablen usw.

Ein Teil der Dokumentation einer Software besteht auch in der Verwendung von Kommentaren im Quellcode. Mit ihnen kann man ausgewählte Teile des Quellcode erläutern.

In Java beginnen mehrzeilige Kommentare mit `/*` und enden mit `*/`. Einzeilige Kommentare beginnen mit `//` und wirken bis zum Zeilenende. Kommentare werden vom Java-Compiler `javac` ignoriert, man kann sie daher auch dazu verwenden, um bestimmte Teile eines Programms „ein- und auszuschalten“, um z. B. die Wirkung derselben zu untersuchen und damit u. a. Fehler zu finden.

### 1. Javadoc-Kommentare und -Tags

In Java wird mit `/**` ein mehrzeiliger Kommentar eingeleitet, der von dem Programm `javadoc` zur automatischen Generierung der HTML-Quellcode-Dokumentation verwendet wird. So ist die `Java`

API Dokumentation mit `javadoc` erstellt. Ein Javadoc-Kommentar endet auch mit `*/`; dazwischen können und sollen auch Javadoc-Tags verwendet werden, wie z. B.:

#### *Häufig verwendete Javadoc-Tags*

`@author`

Angabe des Autors

`@version`

Angabe der Version

`@return`

Beschreibung des Rückgabewertes einer Methode

`@param name`

Beschreibung des Parameters mit dem Bezeichner `name`.

Die Tags `@author` und `@version` werden in dem Javadoc-Kommentar verwendet, der eine Klasse oder eine Schnittstelle beschreibt, sie werden daher üblicherweise zu Beginn einer Java-Datei verwendet. Neben diesen gibt es noch einige weitere Tags.

### Quellcode 1. Beispiel zur Verwendung von Javadoc-Kommentaren

```
/**
 * The class switch is the model of a switch with the states ON and OFF.
 * You can connect multiple Lamp with it. ❶
 *
 * @author André Bauer
 * @version 1.0
 */
public class Switch {

    /**
     * The current state of the switch.
     */
    private State state;

    /**
     * Connects a lamp with the switch. ❷
     *
     * @param lamp The lamp which should become connected. ❸
     */
    public void connect(Lamp lamp) {
        lamps.add(lamp);
    }

    /**
     * Returns the current state of the switch.
     *
     * @return The current state of the switch. ❹
     */
    public State getState() {
        return state;
    }

    ...
}
```

- ❶ Beschreibung der Aufgabe der Klasse zu Beginn der Datei.
- ❷ Erläuterung der Aufgabe einer Methode. Bei Methoden, die als Rückgabe den Typ `void` haben, wird in der Dokumentation auf das Tag `@return` verzichtet.
- ❸ Der Bezeichner nach dem Tag `@param`, hier ist es `lamp`, muss mit dem Parameter in der dokumentierten Methode übereinstimmen.
- ❹ Da `getState()` einen Wert zurückliefert, wird dieser mit dem Tag `@return` dokumentiert. Da diese Methode keine Parameter hat, wird auf das Tag `@param` verzichtet.

## 2. Der Javadoc-Übersetzer

Javadoc ist ein Kommandozeilen-Werkzeug. Bei einfachen Projekten, bei denen alle Quellcode-

Dateien in einem Verzeichnis liegen, wird mit dem folgenden Aufruf die HTML-Dokumentation aller Klassen und Schnittstellen erzeugt:

```
$ javadoc -author -version -private *.java  
  
$ firefox index.html &
```

In einigen Entwicklungsumgebungen ist Javadoc integriert. So kann in [BlueJ](#) unter [bearbeiten](#) zwischen [Quelltext](#) und [Dokumentation](#) gewechselt werden. In einem [Screencast zur API-Dokumentation mit Javadoc](#) wird die Verwendung von Javadoc anhand von Eclipse vorgeführt.