

Softwarepraktikum

# Javadoc

Daniel Fritze Patrik Marschalik

15. Mai 2002

## 1 Was ist Javadoc?

*Javadoc* ist ein Tool, das aus Java-Quelltexten Dokumentationen im HTML-Format erstellt. Dazu verwendet es die öffentlichen Klassen-, Interface- und Methodendeklarationen und fügt zusätzliche Informationen aus eventuell vorhandenen Dokumentationskommentaren hinzu. Zusätzlich generiert Javadoc diverse Index- und Hilfsdateien, die das Navigieren in den Dokumentationsdateien erleichtern.

## 2 Syntax

Javadoc-Kommentare beginnen mit `/**` und enden beim nächsten `*/`. Javadoc-Kommentare werden nur dann erkannt, wenn sie direkt vor der Deklaration einer Klassen, eines Interfaces, eines Konstruktors, einer Methoden oder einer Variablen stehen. Der erste Satz in einem Java-Kommentar ist üblicherweise eine Kurzbeschreibung und endet beim ersten Punkt.

### 3 Tags

Tags sind Kommentare, die spezielle Informationen kennzeichnen. Alle Tags beginnen mit einem @ und stehen meist in einer eigenen Zeile.

#### Die wichtigsten Javadoc-Tags:

Tag	Beschreibung	zur Kommentierung von
@author <Name des Autors>	Der @author Tag spezifiziert einen Autor des Programms. Es ist auch möglich mehrere @author Tags zu verwenden.	Klassen, Interfaces
@version <Versionsnummer>	Der @version Tag gibt die Versionsnummer an und darf pro Klassen bzw. Interfaces nur einmal verwendet werden.	Klassen, Interfaces
@see <Referenz>	Der @see Tag erzeugt einen Querverweis auf eine andere Klassen, Methoden oder einen beliebigen Teil der Dokumentation.	Klassen, Interfaces, Instanzvariablen, Methoden
@link <Referenz>	Im Gegensatz zum @see Tag, der üblicherweise im Zusammenhang mit einem Hinweis am Ende der Dokumentation eingesetzt wird ("Siehe auch...") gehört der @link Tag zu einer Referenz, die mitten im Text aufgeführt wird.	Klassen, Interfaces, Instanzvariable, Methoden
@since <Version>	Der @since Tag gibt an, seit wann das beschriebene Feature existiert.	Klassen, Interfaces
@param <Parameter><Beschreibung>	@param beschreibt den Parameter der Methode, wobei die Beschreibung mehrzeilig sein kann.	Methoden
@return <Beschreibung>	Der @return Tag dokumentiert des Rückgabewert eines Methodenaufrufs.	Methoden
@deprecated <Hinweis>	@deprecated kennzeichnet eine alte Methode, die zukünftig nicht weiter verwendet werden sollte.	Methoden
@exception <Ausnahme><Kommentar>	Beschreibung einer Ausnahme, die von dieser Methode geworfen wird.	Methoden

## 4 HTML-Formatierung

Javadoc Kommentare können auch HTML-Tags beinhalten um den Text besser formatieren zu können. Doch dabei sei Vorsicht geboten. Da Javadoc ein HTML-Dokument mit einer Standard Struktur erzeugt, kann die Verwendung von HTML-Elementen, die die Dokumentstruktur ändern (wie z.B. zusätzliche Überschriften), zu Problemen führen.

Hier die wichtigsten HTML-Tags:

<code>&lt;i&gt;Text&lt;/i&gt;</code>	<i>kursive Schrift</i>
<code>&lt;b&gt;Text&lt;/b&gt;</code>	<b>fett gedruckte Schrift</b>
<code>&lt;sup&gt;Text&lt;/sup&gt;</code>	hochgestellte Schrift
<code>&lt;sub&gt;Text&lt;/sub&gt;</code>	tiefgestellte Schrift
<code>&lt;small&gt;Text&lt;/small&gt;</code>	Schrift eine Stufe kleiner
<code>&lt;big&gt;Text&lt;/big&gt;</code>	Schrift eine Stufe grösser
<code>&lt;center&gt;Text&lt;/center&gt;</code>	zentrierte Ausrichtung
<code>&lt;br&gt;Text&lt;/br&gt;</code>	Zeilenumbruch
<code>&lt;a href="http://www.torfu.de/"&gt;&lt;/a&gt;</code>	Link
<code>&lt;code&gt;Text&lt;/code&gt;</code>	Quellcode

## 5 Wie erstellt man eine Javadoc Dokumentation?

Javadoc wird über die Komandozeile aufgerufen und hat folgende Syntax:

```
javadoc [options] {package sourcefile}
```

Javadoc akzeptiert entweder eine Serie von Java Packages oder Dateinamen. Beim Aufruf von Javadoc, wird ein Klassenbaum aufgebaut und ein Index aller Klassen, die gefunden wurden. Standardmässig werden Elemente des Typs `package`, `public` und `protected` dokumentiert. Der Autor und die Version werden standardmässig nicht angezeigt. Um Einfluss auf die Erstellung der Dokumentation zu nehmen, gibt es unter anderem folgende Optionen.

### Wichtige Javadoc-Optionen:

-public	zeigt nur Elemente des Typ public
-protected	zeigt nur Elemente des Typ protected (Standard)
-private	zeigt alle Elemente
-package	zeigt nur Packages
-version	der @version Tag wird ausgewertet (nicht Standard)
-author	der @author Tag wird ausgewertet (nicht Standard)
-noindex	es wird kein Packageindex generiert
-notree	es wird kein Klassenbaum generiert
-classpath Pfad	gibt die Liste der Pfade zur Suche von Klassendateien an
-sourcepath Pfad	Pfad mit den Quelldateien an
-d Verzeichnis	Verzeichnis, in dem die generierten Dokumentationsdateien abgelegt werden. Standardmässig werden sie im aktuellen Verzeichnis angelegt.
-verbose	Ausgabe zusätzlicher Meldungen während der Dokumentationserstellung