

Vorlesung 02b: Javadoc

Peter Thiemann

Universität Freiburg, Germany

SS 2010

Kommentare

- ▶ Bisher: Freistil-Kommentare für Klassen und Methoden
- ▶ Ab jetzt: Kommentare nach der Javadoc-Konvention

Javadoc

- ▶ Javadoc ist ein Werkzeug der Java-Plattform
- ▶ Anwendungsbereich: Erstellen von Dokumentation
- ▶ Verwendung
 - ▶ Einfügen von strukturierten Kommentaren im Quellcode des Programms
 - ▶ Extrahieren dieser Kommentare als eigenständige Dokumentation (Werkzeug)
- ▶ Vorteil: Dokumentation ist immer synchronisiert mit dem Quellcode

Javadoc-Kommentare

Ein Javadoc-Kommentar

- ▶ bezieht sich auf eine Deklaration einer Klasse, eines Felds, eines Konstruktors oder einer Methode;
- ▶ steht direkt *vor* der Deklaration, auf die er sich bezieht;
- ▶ ist in HTML geschrieben und verwendet spezielle (eigene) Tags für spezifische Funktionen.

Beispiel: Javadoc-Kommentar

```
/**  
 * Returns a {@link Square} that fully covers the shape. Needs to be a rectangle  
 * if it is desired to return the smallest enclosing rectangular shape.  
 *  
 * @return the smallest square containing the shape  
 */  
public Square bb();  
  
/**  
 * Returns true if the argument point lies inside the shape.  
 *  
 * @param p the point to be checked (must not be null)  
 * @return true if p is inside the shape  
 */  
public boolean in(CartPt p);
```

Bemerkungen

- ▶ Die erste Zeile startet einen Javadoc-Kommentar: `/**`
- ▶ Beendet durch `*/`
- ▶ Automatische Formattierung und Einfügung durch Eclipse durch Eintippen der ersten Zeile
- ▶ Der erste Satz sollte einen kurzen Überblick über die Funktionalität der Methode geben. Er wird automatisch in die Übersichtstabelle übernommen.
- ▶ Das Tag `{@link Square}` wird in einen Hyperlink auf die entsprechende Klassendefinition umgewandelt.
- ▶ Mehrere Paragraphen werden in der Beschreibung durch `<p>` getrennt.
- ▶ Leerzeile vor der abschliessenden Tagliste!
- ▶ Die erste Zeile, die mit `@` beginnt, beendet die Beschreibung
- ▶ Eclipse erkennt die Javadoc-Kommentare und verwendet sie in seiner Hilfsfunktion.

Richtlinien

Formulierung von Javadoc-Kommentaren

- ▶ Beschreibe die Funktion einer Methode, Klasse, etc. Verwende nicht den Imperativ.
 - ▶ OK: Returns true if ..., Computes a number such that ...
 - ▶ Bad: Return true if ..., Compute a number such that ...
- ▶ Methodenbeschreibung beginnen mit einem Verb.
 - ▶ OK: Computes a number ...
 - ▶ Bad: This method computes a number ...
- ▶ Verwende `this` um über das Empfängerobjekt einer Methode zu reden.
- ▶ Wiederhole nicht einfach den Namen als Beschreibung.

Tags zur Beschreibung von Methoden und Konstruktoren

@param

- * @param ch the character to be tested
- * @param observer the image observer to be notified

- ▶ Reihenfolge entsprechend der Reihenfolge in der Parameterliste
- ▶ Das erste Wort nach dem Tag ist der Name des Parameters, nicht der Typ.
- ▶ Rest: weitere Beschreibung
- ▶ Das erste Nomen in der Beschreibung bezeichnet den Datentyp.
Ausnahme: int wird meist weggelassen
- ▶ Parameternamen werden klein geschrieben.
- ▶ Die Schreibweise der Beschreibung folgt der Grammatik:
 - ▶ Falls ganzer Satz, groß anfangen und mit Punkt beenden.
 - ▶ Falls kein Satz, beende nur dann mit Punkt, falls weitere Teile der Beschreibung folgen.

Tags zur Beschreibung von Methoden

@return

- ▶ Weglassen bei Konstruktoren und Methoden, die `void` liefern.
- ▶ Sonst: verwenden, auch wenn der Inhalt redundant ist (Wdhl aus der Methodenbeschreibung)
- ▶ Erkläre die Rückgabewerte für Spezialfälle
- ▶ Konventionen für die Beschreibung wie bei `@param`

Tags zur Beschreibung von Klassen

@author

- ▶ Gibt den Autor einer Klasse (oder eines signifikanten Teils) an
- ▶ Nicht verpflichtend, mehrere Autoren möglich
- ▶ Konvention: Autor der ersten Version zuerst, weitere Autoren in zeitlicher Reihenfolge (falls überhaupt)
- ▶ Nur im Quellcode, nicht in der erzeugten Dokumentation.