

2IP15 Programming Methods

From Small to Large Programs

Tom Verhoeff

Eindhoven University of Technology

Department of Mathematics & Computer Science

Software Engineering & Technology Group

www.win.tue.nl/~wstomv/edu/2ip15

Overview

- Javadoc

Javadoc Tool

Name: javadoc

Purpose: generate documentation for Java software

Input: documented Java code source files (.java) in a package

Output: hyperlinked HTML files documenting the Java software

In DrJava: button (top right)

In NetBeans: right-click on project >

Javadoc syntax

- Uses specially formatted “doc comments” in Java source files:

```
/**  
 * This comment will be processed by javadoc.  
 */
```

- Are placed *immediately before* the relevant element:
class, interface, constructor, method, or field declarations

- May include HTML markup and special tags marked by @:

```
/**  
 * Returns the correctly rounded positive square root of  
 * a {@code double} value. ... <ul><li>...<li>...<li>...</ul> ...  
 *  
 * @param x a value.  
 * @return the positive square root of {@code x}. ...  
 */  
public static double sqrt(double x) { ... }
```

Javadoc rendered output: Summary form

Method Summary	
...
static double	<u>sqrt</u> (double x) Returns the correctly rounded positive square root of a double value.
...

Javadoc rendered output: Full form

sqrt

```
public static double sqrt(double x)
```

Returns the correctly rounded positive square root of a double value. Special cases:

- If the argument is NaN or less than zero, then the result is NaN.
- If the argument is positive infinity, then the result is positive infinity.
- If the argument is positive zero or negative zero, then the result is the same as the argument.

Otherwise, the result is the double value closest to the true mathematical square root of the argument value.

Parameters:

`x` - a value.

Returns:

the positive square root of `x`. If the argument is NaN or less than zero, the result is NaN.

Javadoc Comment Format

Main description followed by tag section

First sentence of main description ...

- ... should be a summary sentence, containing a concise but *complete* description of the declared entity;
- ... ends at the first period that is followed by a blank, tab, or line terminator, or at the first block tag;
- ... is copied to the member summary at the top of the HTML page.

Block tags and in-line tags

Block tags appear in the `tag section` *following* the main description.

Form: `@tag ...` followed by one or more arguments (texts)

In-line tags appear in the main description or parameter texts of block tags.

Form: `{@tag ...}`

E.g.: `{@code ...}` for typewriter font

Tags for constructors and methods

- `@param` *name purpose*
- `@return` *value_as_function_of_parameters_and_fields*
- `@throws` *exception condition*
- `@pre` *precondition* (not standard)
- `@modifies` *list_of_parameters_and_fields* (not standard)
- `@post` *postcondition* (not standard)
- `@bound` *integer_bound_function_for_termination* (not standard)
(Especially for recursion)

Tags for classes

- @since *date*
- @version *version_number*
- @author *author_(identification_number)*
- @see *"text"* or `label`
- @inv *name predicate* (not standard)

For public class invariants

Tags for fields

- *None*

There is just a main description. It is used to state private invariants.

Note: Do not declare multiple fields in a single declaration, because there can be only one doc comment for each declaration.

Style Hints for Javadoc Comments

- Use `{@code ...}` for Java keywords and names
- Okay to use phrases instead of complete sentences
- Use 3rd person (descriptive), *not* 2nd person (prescriptive)
Preferred: “Gets the label ...”; Avoid: “Get this label ...”
- Method descriptions begin with a verb phrase
Avoid: “This method gets the label ...”
- Use “this” instead of “the” when referring to an object created from the current class
- Add description beyond what the name already tells

Documentation about Javadoc

- *Javadoc Tool Reference Page*

See docs.oracle.com/javase/7/docs/technotes/guides/javadoc

Read: *Documentation Comments: Commenting the Source Code*

- *How to Write Doc Comments for the Javadoc Tool*

See www.oracle.com/technetwork/java/javase/documentation/index-137868.html

Read: *Descriptions, A Style Guide, Tag Conventions*

N.B. Use `{@code ...}` instead of `<code>...</code>`

- *Requirements for Writing Java API Specifications*

See www.oracle.com/technetwork/java/javase/documentation/index-137868.html

Javadoc and NetBeans

- Activate: Right-click on project > Generate Javadoc
- Additional options: Right-click on project > Properties
Categories: Build > Documenting > Additional Javadoc Options

Example: `-tag pre:cm:"Precondition:"`

`-tag modifies:cm:"Modifies:"`

`-tag post:cm:"Postcondition:"`

Or: `-docletpath ...Doclet.jar -doclet ...Doclet`

- Tools > Options > Editor > Hints > Javadoc
 - Create Javadoc
 - Error in Javadoc

Summary

- Javadoc