

Javadoc Guidelines- Professor Hoskey

Overview

To generate Javadoc comments you must perform two tasks:

1. Add Javadoc comments to your code.
2. Run the Javadoc tool.

The Javadoc tool looks through all of your code for Javadoc comments and then uses those comments to generate HTML documents describing your code.

What Are Javadoc Comments Used For

Suppose we buy a new device and it comes with a user's manual. The user's manual will likely describe the following:

1. The general purpose of the device.
2. What each control on the device does.

This user's manual will not go into detail about the inner workings of the device. It will not be describing the details of what wires are connected on the inside. There may be another manual specifically designed for engineers that might describe the inner workings.

Javadoc comments are like a user's manual but for a class instead of a device. It should give a general description of the class ("the device") and it should give descriptions of each of the methods ("the controls on the device"). Javadoc comments should NOT be used to describe the inner workings of the methods. Comments such as "the while loop in the method is used to calculate the subtotal for all items" should be put in normal comments. Normal comments are like the engineer's manual. The user's manual and the engineer's manual are both important but they are written for different audiences.

Guidelines for Required Javadoc Comments

1. Add Javadoc comment right above every class definition.
2. Add Javadoc comment above every method definition.

Basic Javadoc Comments

Start a Javadoc comment with `/**`.

End a Javadoc comment with */.

For example:

```
/**
```

Here is a Javadoc comment.

```
*/
```

Javadoc Class Comments

All classes must have a Javadoc comment preceding them. There should be a description of the class followed by the @author, @version, and @since tags. For example:

```
/**
 * Contains main program code. This class generates a report for
 * state representatives.
 *
 * @author Arthur Hoskey
 * @version 1.0
 * @since 9/1/2020
 */
public class Driver { ... }
```

Class description comes first

Tags come after description

Javadoc Method Comments

All methods in all classes must have a Javadoc comment describing its purpose as well as usage. There should be a description of the method followed by the @param and @return tags.

@param – Used to describe a method parameter. Every method parameter should be given a separate one of these tags. You need to give the parameter name followed by a description of what it is.

@return – Used to describe the method return value. Every method should have one unless the return type is void.

For example:

```
/**
 * This method adds two numbers together and returns the result.
 *
 * @param num1 First number to add.
 * @param num2 Second number to add.
 * @return The sum of num1 and num2.
 */
public int Add(int num1, int num2)
{
    return num1 + num2;
}
```

Method description comes first

Tags come after description

Running the Javadoc Tool

To actually generate the HTML files containing your Javadoc comments you need to run the Javadoc tool. Here is how you do it:

1. Select the menu item Run|Generate Javadoc.
2. To view the documentation go to the target/site/apidocs directory within the project. This directory will have the .HTML documents that the Javadoc utility created. Open them to view the Javadoc comments.

View Javadoc

Go to Window|IDE Tools|Javadoc Documentation. This will open a tab named Javadoc in the bottom window inside NetBeans. If you put the cursor on a class or a method in your code (by left clicking, not hovering) it will show the Javadoc document for that item in the Javadoc tab. You can also open File Explorer and navigate to the target/site/apidocs directory within your project. The Javadoc .html files that were created when you ran the Javadoc tool will be there.

Javadoc Links

<http://www.oracle.com/technetwork/articles/java/index-137868.html>

<http://en.wikipedia.org/wiki/Javadoc>