

Javadoc



The Plan

- What is Javadoc?
- Writing Javadoc comments
- Using the Javadoc tool
- Demo
- Practice

What is Javadoc?

Javadoc is a way to comment your code that enables automatic generation of web pages that document your code.

Why use Javadoc?

- It's much faster than generating webpages documenting your code.
- It's standard documentation which means it's easy to use and the structure is given.

Writing Javadoc Comment

- Javadoc comments start with `/**` and end with `*/`
- The placement of the comment is important.
- The following can be commented:
 - classes
 - methods
 - instance variables
 - static variables

Commenting a Class

- Put the comment immediately before the class declaration.
- Briefly describe the purpose of the class in 2-3 sentences.
- Optionally include
 - @author tag
 - @version tag
 - others

Commenting a Method

- Put the comment immediately before the method declaration.
- Briefly describe the purpose of the method in a short phrase or 2-3 sentences. Include more detail if necessary
- Include these tags if needed
 - @param name – describes parameter
 - @return – describes the return value

Commenting Instance and Static Variables

- Put the comment immediately before the variable declaration.
- Briefly describe the purpose of the method in a short phrase. Include more detail only if absolutely necessary.
- No tags needed.

For more information...

Visit the article:

How to Write Doc Comments for the Javadoc Tool

<http://java.sun.com/j2se/javadoc/writingdoccomments/index.html>

Using the Javadoc Tool

- On the command line, change into the directory where your code is.
- Type

```
javadoc -subpackages [source_directory]  
-d [target_directory] -author
```

all on the same line

- This generates web page documentation in `target_directory` by reading the Java source code in `source_directory`

Using the Javadoc Tool


- If your source code is in a jar file (like ours normally is), you'll first need to expand the jar file. To do this:
 - change directories until you are in the directory containing the jar file
 - list the directory's contents to make sure the jar file is in the current directory
 - type
jar -xvf file.jar

Using the Javadoc Tool

- If your source code is in the current directory, use dot for the filename:

```
javadoc -subpackages .  
-d [target_directory] -author
```

Don't forget the dot!



Using the Javadoc Tool

- Instructions particular to our lab:
 - To get to the command line, use Java ssh:
<http://www.oit.duke.edu/sa/security/ssh.html>
 - At the command line use
 - **cd** to change directories
 - **ls** to list a directory's contents
 - **pwd** to print the working directory
 - **cd ..** to go back a directory
 - Be sure code and documentation are in a subdirectory of `public_html`

Demo

1. Save Blast.jar to the Desktop.
2. Copy into public_html/cps4/homework/hwX.
3. Expand Blast.jar
4. Generate web pages using Javadoc
5. View from a web browser
6. Add Javadoc comment in Eclipse
7. Export as jar file.
8. Repeat 2-5 once with added comment.

Practice

- Put Javadoc comments in one of the previous homework assignment's source code.
- Export the source code to a jar file and put the jar file in the proper directory on your account (somewhere within public_html).
- Expand the jar file.
- Use the javadoc command to generate web page documentation.
- View the documentation on a web browser.