

Generating documentation for Java programs

Martin Pola

1 Your first documentation

Find some of your classes from the previous exercises, preferably some with some methods that would make sense for someone else to use if you shared the class.

The `Book` and `Movie` classes are probably two good candidates. Ensure that both the class itself is public (e.g. `public class Book {...}`) and that all methods that are not for internal use, if any, are public too.

1. Create a new directory for documentation.
2. Open a terminal and navigate to your directory with the `cd` command.
3. Generate Javadoc for one of your classes and inspect the results by open the HTML file in your web browser. Use a command like this one:

```
$ javadoc C:\path\to\code\Book.java
```

4. Generate Javadoc for the rest of your classes in the same folder. You can do that by replacing `Book.java` by `*.java` (an asterisk).

2 Write about your classes

Now that you've been able to generate some documentation, it's time to explore how you can use plain English to write about your classes. While the automatically generated documentation usually serves you well to begin with, other people would likely prefer if all those formal definitions in the HTML page were accompanied by some sentences with details about what they are used for.

Add Javadoc comments (`/** some information */`) right before your public class, method, enum and property definitions. Use annotation (e.g. `@author` or `@param`) whenever reasonable to structure the information a bit.

To get inspiration, you could look at the following two links.

- [Sample class with comments and annotation](#)
- [Javadoc for the ArrayList class](#)

Re-generate your documentation. Where did your comments go? How are the annotations presented? Do you think that your documentation is clear? What can be improved? Look at your neighbours' results and reason with them!