

M.Roggenbach, CS-135 – Lab Class 1 – Monday 3.2.2020

- To be solved in groups of two.
- To be ticked off in one of the labs of your house on Monday, 3.2., or Monday, 10.2.
- For being ticked off on Monday, 10.2., you need to have your solution ready at the begin of the labclass.
- You can obtain two marks by solving this sheet.
- Each completed task gives you one mark.
- All group participants need to be present to be ticked off.

Note that there are computer instruction at the end of this lab sheet.

This lab is about producing comments with the javadoc tool.

☐ Task 1.1

This task is on using the javadoc tool in order to produce documentary comments.

- Download, study, compile, and run the Java stack program, found under
`http://cs.swan.ac.uk/~csmarkus/Tools/Stack.java`
- Run javadoc on the code as it is. Explore the result in a web-browser.
- Add the documentary comments as discussed in the lecture to the program code – make sure that author and version show in the produced documentation.

Material to shown when getting ticked off: A screen shot of the web page produced by javadoc, showing that *all* eight documentary comments are there, and the @author and @version tag have been used.

☐ Task 1.2

This task is on using the javadoc tool in order to produce comments for method declarations.

Use javadoc to comment all method declarations according to MR's commenting standard on method declarations as discussed in the lecture.

MR's standard requires you to write up to five different comments for each method declaration. Some of these comment types are supported with javadoc tags, some are not. You shall use javadoc tags as much as possible. You should use the @param tag to document each parameter. You should also use the @return tag to document all return values. For those comment types that javadoc does not support, you need to 'invent' some formatting. The excellent student will use HTML tags in order to format these comments.

For some methods, a comment type is not appropriate and should be omitted:

- @return tag should only be used if there is a return type different from void;

- Referential transparency should only be addressed if there is a return type different from `void`;
- `@param` tag should only be used if there is a parameter;

Furthermore note the following:

- It might not always be ‘easy’ to give a formula – in that case don’t add a formula.
- In this task, there is no need to add any comment to the declaration of the main method.

Material to shown when getting ticked off: A screen shot of the web page produced by javadoc, showing that all method comments are there.

Computer Instructions

Getting a terminal

Under the “Specialist Apps”, open the folder “College of Science”. Within this folder, open the folder “Computer Science”. There, you find various version of the “Java Development Kit”, which provides a terminal configured for Java. For our purposes, any of them will do.

Changing to the right directory

Typing `p:` changes the drive to the one with your home directory.

The **Change Directory - Select a Folder (and drive) command:**

Syntax

```
CD [/D] [drive:][path]
CD [..]
```

Key

`/D` : change the current DRIVE in addition to changing folder.

Examples

To change to the Desktop.

```
p:\> cd Desktop
```

To change to the parent directory.

```
p:\Desktop> cd ..
```

The Java commands

These commands can be typed in the command line of the terminal:

1. `javac <filename.java>` compiles a program.
2. `java <filename>` executes the .class file.
3. `javadoc -d <directory-name> -version -author <filename.java>` produces the documentation.

Recommended editor

Under the “Specialist Apps”, open the folder “College of Science”. Within this folder, open the folder “Computer Science”. There, you find the program “Notepad++”, which provides a reasonable environment to edit Java programs.

Making a screen shot

Click on ‘Start’, type ‘Snipping Tool’ in the search field, press ‘enter’. Use the tool.