


Individual Homework 3

Start Assignment

- Due Mar 19 by 11:59pm
- Points 100
- Submitting a file upload
- File Types pdf
- Available Jan 13 at 12am - Mar 21 at 11:59pm

Introduction

Using the provided materials and the team's TP2 submission, each student must:

- work with the rest of the team to divide the already existing automated tests in the team's TP2 submission so that each member has at least five tests and **document** which existing tests were assigned to which team member. If there are not enough automated tests to give each member five different tests, select tests to be implemented by more than one member of the team. Allocate these tests so each member has five tests, and each member has roughly the same number of duplicate tests as the others.
- design, **document**, create and possibly extend, and then test the JUnit tests allocated to you.
- explore using Javadoc, find a professional-looking Javadoc example (e.g., Oracle's documentation of the [Java Collections Framework](https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html) ) (<https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html>) that you believe is worthy of emulation, update the internal documentation to produce similar Javadoc output for the updated TP2 submission, and produce a PDF of the generated Javadoc output.

You are expected to engage with others in the class, directly or via Ed Discussion, to identify good information sources on JUnit testing and good examples of generated Javadoc output. Do not blindly implement what you believe others are doing, and most definitely, **do not copy** their work!

With the requirements satisfied, you must create a short screencast that shows that the application performs each requirement and successfully performs the five newly created JUnit tests. Use the provided MS Word template, fill it out, and submit it as a PDF where links can be copied and/or clicked to access your screencast and code within your personal GitHub. The detailed requirements for this homework are specified below.

You are encouraged to get help from your classmates if you have issues with JUnit or Javadoc and offer to help others if you have been successful in using them.

Tasks

This homework requires you to perform the following tasks.

1. Fill in the cover page of the template as required.
2. Collaboratively with your teammates, partition and **document** a set of five test cases as described above.
3. Design, create, test, and **document** the set of five JUnit test cases allocated to you as you update the Team's TP2 submission. Rename your update of this application to HW3.
4. Identify the source that you believe shows what professional-looking documentation should be. Update and extend the internal **documentation** to produce professional-looking Javadoc output of your HW3 application.
5. Produce a screencast that covers the following:
 - the implementation of the original five automated tests and how they work
 - the implementation of JUnit replacement tests and how they work
 - the results of running the JUnit tests and the output produced using Javadoc
 - Store your screencast and HW3 in your personal GitHub repository. Make sure that only you and your grader have access to your repository.
6. Produce a PDF that contains
 - the list of five test cases allocated to you in Task 1
 - the JUnit test cases described in Task 2.
 - A link to your HW3 application as described in Task 3.
 - a link to the source you found that shows what you believe professional-looking Javadoc output should be. Following that should be the Javadoc report output from HW3, as described in Task 4.
 - Provide links that can be copied and/or clicked to access your HW3 code and your screencast from within your GitHub repository.
7. Submit this PDF before the deadline so there is enough time for the upload of the submission to finish, for Canvas to process the submission, and for Canvas to add it to its data repository before the deadline. Just **starting** the upload before the deadline is **not adequate**!

Deliverables

A PDF document must be produced that covers the following items. The links in the PDF must be copyable or clickable to access the GitHub copies of the code and screencast.

- Use the following Template for your PDF submission. Fill in this template with the results of the following tasks and submit it. (5%)

HW3-Submission-Template.docx

- Task 1: Cover page complete with your name. (5%)
- Task 2: A list of the five tests allocated to you. (5%)
- Task 3: Design, create, test, and **document** the set of five JUnit tests. The application must be named HW3. (20%)
 - Five JUnit tests that implement or enhance the implementation of the allocated automated tests. (10%)
 - Test the five JUnit tests to show they work as required. (5%)

- The five JUnit tests have internal documentation equivalent to the rest of the HW3. (5%)
- Task 4: The internal documentation has been updated to produce Javadoc output. (15%)
 - A link to a source showing the professional-looking Javadoc material and an explanation for what was compelling about this example. (5%)
 - Include the Javadoc output generated by the five JUnit tests. (5%)
 - Include the Javadoc output generated by the entire HW3. (5%)
- Task 5: Store your HW3 Screencast in your personal GitHub Repository. (35%)
 - Screencast shows and explains the code and that the code is in alignment with the external and internal documentation. (10%)
 - The code is readable, and the explanations are clear and audible. (10%)
 - The screencast explains and demonstrates how the JUnit tests work. (10%)
 - Store your screencast and HW3 in your personal GitHub Repository. (5%)
- The grader must be able to examine the code in your GitHub, see that it is nicely formatted with internal documentation, and determine that it is consistent with the provided code so most people would assume it had been written by the same author. (10%)