

COM4506/COM6506 Testing Assessment

Setup

This work will be submitted via GitHub. For this assignment you will submit it using Github classroom. You can set up your repository by following this link: <https://classroom.github.com/a/0zSyw3TS>. Use this repository as a basis for working on your project.

Task

The task is use Metamorphic testing to test some sorting and searching capabilities implemented within the Oracle Java SDK. Specifically, we are interested in the following methods:

- Collections.sort(List list)
<https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html#sort-java.util.List->
- Collections.rotate(List list, int distance)
<https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html#rotate-java.util.List-int->
- Collections.min(Collection coll)
<https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html#min-java.util.Collection->

Tasks

1. Start by applying the Category Partition method to identify categories for the test inputs. For each type of parameter (List, distance and Collection), identify 5-8 categories that ought to probe the behaviour of the functions under test in some way.

Note: It is possible that two categories can apply to the same input at the same time.

Do not select categories that will trivially fail or throw exceptions, such as syntactically invalid inputs. The goal will be to use the categories for Metamorphic test cases; if an input always fails, it is not a suitable basis for metamorphic testing.

To submit: In your written report, include a section called "Category Partition Method". For each category, give it an ID number, a sentence to clearly describe the corresponding inputs, and 1-2 sentences to justify how you chose it (referencing relevant documentation where relevant).

2. Combine the categories to identify three distinct test cases for each function and derive the corresponding concrete input.

For each test case, in the comments above the test case refer to the identifiers for each of the categories that has been used. Do not use the same combination of categories twice.

To submit: In your written report, include a section called “Test cases” where, for each function, you list the combination of category IDs used, and the choice of values.

3. For each function, identify two Metamorphic relations. Think - in what ways can you “transform” the various input parameters, and how can you expect these transformations to affect the output?

To submit: In your written report, write a small section called “Metamorphic Relations”. For each relation, describe it as: (1) the transformation from ‘original’ input to ‘transformed’ input, and (2) the expected relationship between the resulting outputs. For example (with respect to our Sin example from the lectures):

(a) Transformation from original (x) to transformed (x'): $x' = \pi - x$

(b) Expected relationship between $z = \sin(x)$ and $z' = \sin(x')$: $z == z'$

4. Combine your Category Partition test cases with the metamorphic relations to produce JUnit test cases in Java.

To submit: For each function, create a separate JUnit test class. For every combination of metamorphic relation and test input, generate a separate test method (you should have created three test inputs and two metamorphic relations per function, resulting in six test methods per JUnit class).

In a code comment above each test method, indicate which category-combinations and which metamorphic relations are being tested.

5. Add an automated test generator.

To submit: Select two of the test methods you created in the previous step. Copy the test method and add the word “Automated” to the end of the copied method name. Amend the test code so that the metamorphic relation is tested 1000 times, but with a different randomly generated test input at every execution.

6. Include a brief writeup (approximately 200 words), summarising the outcome of the tests. Which test cases (if any) failed? If so, was this because there was a potential mistake in the documentation for the parser? Or could it be a genuine fault in the code?

To submit: Include a section called “Remarks” with the writeup in the written report.

Submission instructions

For the Java and JUnit libraries, we are assuming that you are using the Oracle Java JDK version 8 or higher and JUnit Jupiter (version 5.x).

The submitted version will count as the version at the time of the deadline. Include all of your Java JUnit classes in a directory called “src”. Include the report as a *PDF* in the base directory of the assignment.

Deadline: The deadline for this assignment is the 13th December, 23:59 GMT. The submitted version will be your last push to the GitHub repository. Any pushes to GitHub after this time will not be accepted.

Assessment

Assessment will be carried out according to the following factors:

- The appropriateness and justification for the chosen categories (where possible backed up by relevant reference to documentation). 20%
- The appropriateness of the test sets derived from the categories, and their implementation as JUnit tests. 10%
- The choice of appropriate Metemorphic relationships. 30%
- The implementation Metamorphic tests in JUnit, and their description. 30%
- The ability to generate and execute random metamorphic test inputs. 10%