BLG 475E - Software Quality and Testing
2021-22 Spring Term
Homework #1

Due: May, 12th 23:59

"If you think your users are idiots, only idiots will use it."

— Linus Torvalds

Introduction

In this assignment, you will learn how to implement and create unit test cases using JUnit framework.

For any issues regarding the assignment, please contact Doğukan Arslan (**arslan.dogukan@itu.edu.tr**).

Implementation Notes

The implementation details below are included in the grading:

1. Please follow a consistent coding style (indentation, variable names, etc.) with comments.
2. Please follow unit testing best practices summarized in the lecture notes. e.g. Test method names must be meaningful, and they should provide a clear idea what the test is for.

Submission Notes

1. You should compress all your necessary files into an archive file (`.zip`). Please write your name and ID on the top of each document that you will upload as in following format:
```
/* @Author
* Student Name:  <student_name>
* Student ID: <student_id>
*/
```
2. Submissions are made through Ninova system **only**, and have a strict deadline. Assignments submitted after the deadline will not be accepted.
3. This is not a group assignment and getting involved in any kind of cheating is subject to disciplinary actions. Your homework should not include any copy-paste material.

1 Questions

1.1 Warm-up: ISBN (30 pts.)

The International Standard Book Number (ISBN) is a numeric commercial book identifier that is intended to be unique. A 13-digit ISBN can be separated into its parts as prefix element, registration group, registrant, publication and check digit. The ISBN-13 check digit, which is the last digit of the ISBN, must range from 0 to 9 and must be such that the sum of all the thirteen digits, each multiplied by its (integer) weight, alternating between 1 and 3, is a multiple of 10 [1].

$$(x_1 + 3x_2 + x_3 + 3x_4 + x_5 + 3x_6 + x_7 + 3x_8 + x_9 + 3x_{10} + x_{11} + 3x_{12} + x_{13}) = 0\,(mod\,10)$$

You are given a source file called ISBN.java that checks whether an ISBN number is valid or not according to the formula above, and a test file called ISBNTest.java for the corresponding source file. **Analyze** ISBNTest.java for any bad practices or unusual convention examples, **correct** them, and **briefly state** (adding one-line comments) why it is fixed.

Evaluation criteria: Your test file will be evaluated based on the manual review on unit tests' quality.

1.2 Unit testing: Doubly Linked List (70 pts.)

You are given a source file called DoublyLinkedList.java that includes methods to utilize various doubly linked list features. Investigate the given Java class and **implement** test cases for the given methods of this class considering both black box (equivalence class partitioning) and white box testing strategies (coverage) as well as good testing practices summarized in the lecture notes.

Evaluation criteria: Your test file will be evaluated based on the achieved branch coverage and manual assessment of unit tests' quality and effectiveness in detecting bugs. Test files with less than **70%** branch coverage will get only partial points.

Info: The source code given in this assignment is implemented using Intellij IDEA (as an IDE) and JUnit5 (as a testing framework). In order to observe branch coverage in Intellij IDEA, don't forget to activate tracing option and run your tests *with coverage*. See documentation for more information.

References

[1] https://en.wikipedia.org/wiki/International_Standard_Book_Number