CMPSC 290
Principles of Software Development
Fall 2013

Laboratory Assignment Four: Team-Based Implementation and Testing of a Program

Introduction

In the previous laboratory assignments, you have been learning about the tools that we will use this semester to specify, design, implement, test, and document Java programs. Moreover, our past class session have introduced you to the key concepts associated with software engineering. In this assignment, you and your team will follow the phases of the software development life cycle and employ the concepts that we have studied in class to implement and test a program. Two weeks from now, you will present your solution and demonstrate your test suite and program.

Specifying the Requirements

You are responsible for implementing a data generator that should take a list of numbers as input and produce a list of lists as output. You are given the following specification for the data generator.

> For an input list of objects, denoted L, the data generator must produce all of the lists that can be obtained by swapping two adjacent items in L.

For list $L = \{1, 2, 3, 4\}$, the customer wants the data generator to output $L = \{L_1, \ldots, L_6\}$ with

$L_1 = \{2, 1, 3, 4\}$

$L_2 = \{3, 2, 1, 4\}$

$L_3 = \{4, 2, 3, 1\}$

$L_3 = \{1, 3, 2, 4\}$

$L_4 = \{1, 4, 3, 2\}$

$L_6 = \{1, 2, 4, 3\}$

The customer knows that the component that you must create will be a part of a larger system that has not yet been fully implemented. You are responsible for implementing this data generator so that it functions according to the provided specification. However, please note that the stated requirements may not be entirely correct! It is the job of your team to interact with the customer to ensure that the system is implemented as desired. Using LaTeX, you should write a requirements document that fully explains the inputs, outputs, and behavior of the data generator.

Designing the System

Working with the members of your team and leveraging the content in the requirements document, you should create a design for your system. As you are finalizing the object-oriented design, you should try to develop answers to relevant questions such as: How many classes will you use? What will be the relationship between the classes? What methods will the classes have? What will be the inputs and outputs of the methods? Is the design testable? After answering these questions, you should use LaTeX to write a design document with text and diagrams that explain the system.

Implementing and Testing the Program

Using the requirements and design document, your team must implement and test the data generator. You should focus on implementing a program that is both correct and efficient. Just like in the previous laboratory assignment, your implementation must include the following:

1. A build system with rules for building, cleaning, testing, and running the program
2. A high-coverage test suite that effectively tests all of the classes in the program
3. A coverage report that was produced by the JaCoCo coverage monitoring tool
4. Fully documented Java source code that completely fulfills the requirements

Since you cannot exhaustively test this application, you must decide what types of inputs you will create in the test cases. You will also need to determine how you will know that the output of the data generator is correct. For instance, you should consider checking the following conditions:

1. The contents of the output list only contain entities from the input list
2. The output list has the correct number of sublists
3. The output list contains all of the specified sublists

Summary of the Required Deliverables

This assignment invites your team to submit one printed version of the following files:

1. A description of and justification for your team's chosen organization, roles, and tool support
2. A document that clearly specifies the inputs, outputs, and behavior of the data generator
3. A document that explains the data generator's design, with details about classes and methods
4. All of the implementation artifacts (e.g., build system, source code, and the coverage report)

You must also ensure that the instructor has read access to your Bitbucket repository that is named according to the convention `cs290F2013-lab4-team`*k*, with *k* representing the number of your assigned team. Your repository should contain all of the deliverables that you produced during the completion of this assignment. Please see the instructor if you have any questions.