Prof. Lichter

Andreas Steffens, Konrad Fögen

SQA
2019

Exercise 2

sqa@swc.rwth-aachen.de

Issued: 07.05.2018 Submission: 21.05.2019 Discussion: 24.05.2019

1.1. Task: Apply Metrics to a complex software project

Measurements using software metrics support the assessment of software products.
You have to analyze the following software repositories:

1. JHotDraw Framework
   https://supp.swc.rwth-aachen.de/stash/projects/TEACH/repos/swc.oosc.swcarchitect
2. XWiki
   https://dev.xwiki.org/xwiki/bin/view/Community/SourceRepository
3. OpenSource E-Commerce Software
   https://github.com/BroadleafCommerce/BroadleafCommerce
4. Collection of Design Pattern Implementations
   https://github.com/iluwatar/java-design-patterns

For the analysis you may use some of the following tools, which offer free/trial licenses:

1. SonarGraph Explorer: https://www.hello2morrow.com/products/sonargraph/explorer
2. Parasoft JTest: https://software.parasoft.com/jtest/
3. JArchitect: https://www.jarchitect.com/
4. Sourcemeter: https://www.sourcemeter.com/

a) Download the projects. Use the tools to analyze each project. You may need to compile/build some of the projects to get meaningful results. Consult the documentation of the projects for how to build the projects.

b) Describe and discuss your findings of the metrics introduced in the lecture:
   a. McCabe Cyclometric Complexity
   b. Metric Suite by Chidamber & Kemerer
   c. Robert Martin Metrics

   In some tools the metrics are renamed or slightly modified.

c) Provide at least one concrete interesting example from the projects for each metric.

d) The GoF design pattern are best practices for object-oriented design. They are meant to improve maintainability and flexibility of the code.
   Choose three GoF Patterns from the last repository and discuss the metric results for these patterns, how does the pattern or its architecture influence the applied metrics?

To answer the questions you may provide screenshots, images, exported reports and code snippets from the tools and projects.

1.2. Task: Develop a JUnit 5 Tests

Under Test is a library for arithmetic expressions like "`4*(5+6)`". It consists of two parts. First, a *parser* to translate expressions as strings into trees where each node of the tree represents a number or an operator of the expression. Second, the library provides a means to *evaluate* the expressions.

We provide you two different implementations of the library. You can download them as a Maven project from https://git.rwth-aachen.de/swc/sqa-2018-arithm. Everything is setup and an initial test case is provided for each implementation.

Please download the project and copy it into your git repository. Place the tests for implementation `a` into the …`arithma` package and the tests for implementation `b` into the `arithmb` package.

The implementations are expected to fulfil the following requirements.

- The expression library should support expressions with integers including zero, negative and positive numbers but no floating-point numbers.
- The expression library should support addition, subtraction, multiplication, division and modulo operations as well as parentheses. Of course, the ordering (operator precedence) should be respected.

Your task is to write tests to check for the implementation's conformity and, of course, to identify errors.

Starting with the second part of the library, please write tests to check the evaluation capabilities.

a) Before starting to implement, think about the various classes and methods that can be tested. Which ones should be used for testing the evaluation capabilities and why?
   Hint: You should not use the parser frontend in these test cases!
b) For both implementations, we expect at least 10 tests to check the evaluation capabilities.
c) What are your rationales behind choosing exactly that input data and expected values?

Continue with checking the functionality of the library's parser.

d) Before starting to implement, think about the various classes and methods that can be tested. Which ones should be used for testing the parsing capabilities and why?
e) For both implementations, we expect at least 10 tests to check the parsing capabilities.
f) What are your rationales behind choosing exactly that input data and expected values?

g) Please report some errors that you've found. (We expect you to find at least one. You should rethink the input data if you didn't find any.)

Please Note: This assignment is about familiarizing yourself with the JUnit 5 testing framework. Consider using lifecycle methods such as @Before, @After, etc. and also consider using parameterized tests when appropriate.

We would like to ask you to use the templates provided in the L²P room for answering the exercises. Your results must be handed in as a single PDF file or as a compressed ZIP file containing all necessary files, including your source code, and data named „**SQA2018_AssignmentX_GroupX.zip**"; replace the **X** with your group identifier.
Your submission should be submitted via L2P Learning Room.

.