Software Dependability Project Presentation

Apache Commons Text

Calabrese Carmine
0522501853

What is Commons-Text?

A set of functions and components to process and manipulate Text

https://github.com/apache/commons-text

Part of the Apache Commons, a group of project focused on reusable components.

Active since 2017

Counting a total of 62 contributors and 244 forks

Goal of the Project

Project Analysis

Library analysis using external tools and fix of them.

Testing

Improvements of test cases, coverage and performance testing.

Containerization

Real usage of the library inside a web-app using a docker image.

Project Analysis I – SonarCloud

High-Severity Issue

Bugs

Total: 1

Code Smells

Total: 44

Project Analysis I – SonarCloud

Bugs Fix

Bugs

Total: 1

Code Smells

Total: 44

[1] Possible ArrayIndexOutOfBoundException

Added a control to the return statement

Project Analysis I – SonarCloud

Code Smells Fix

Bugs

Total: 1

Code Smells

Total: 44

[19] Refactor this method to reduce its Cognitive Complexity

[12] Test Cases should include assertions

[5] Duplicated string literals

[3] Constant should comply with naming conventions

[5] Method should not be empty

Project Analysis I – SonarCloud

Code Smells Fix

[19] Refactor this method to reduce its Cognitive Complexity

Method with high CC got splitted. Easy when global variables were used, harder when multiple local primitive variables were used

[12] Test Cases should include assertions

No assertions added, all the test-cases relied on throwing exceptions or the correct execution of the method

[5] Duplicated string literals

Fixed by defining a dedicated constant instead of literals

Project Analysis I – SonarCloud

Code Smells Fix

[3] Constant should comply with naming conventions

False Positive. No refactor needed, the class rely on a strong check style so constant should comply the checkstyle imposed by Apache commons

[5] Method should not be empty

Empty Constructor for java bean initialization. Added a comment inside telling the reason of the empty methods.

Project Analysis II – Vulnerabilities

Dependency Check

Apache Commons prefers library with a minimal number of dependencies

Total Dependencies: 5

No vulnerabilities found

2 of the dependencies added during the testing with JMH

Project Analysis II – Vulnerabilities

FindSecBugs

Code analyzed using Spotbugs Plugins for IntelliJ IDE

Dodgy Code	Security
Total: 9	Total: 7

Project Analysis II – Vulnerabilities

FindSecBugs

Code analyzed using Spotbugs Plugins for IntelliJ IDE

Dodgy Code

Total: 9

Security

Total: 7

Only one fixed. Added during the process of the CC Fix.

Other 8 were part of the JMH source code, so out of scope for our analysis

Project Analysis II – Vulnerabilities

FindSecBugs

Code analyzed using Spotbugs Plugins for IntelliJ IDE

Dodgy Code	Security
Total: 9	Total: 7

Vulnerabilities that required a more in-depth study of the entire code base.

Most of them about dev's choice about random generators or methods with important parameters.

Testing I – Coverage Analysis

JaCoco

Jacoco already used for Coverage Analysis inside pipeline

High Standards by Apache Commons

Not respected because of the class added during Analysis

An external pom blocked exclusion of new classed from coverage analysis

All the packages with a coverage value of > 95%

Testing II – Improving Coverage

Automatic test-case generation

Github Copilot **Randoop**

Both tools used to create test-cases, randoop was not able to generate good test-cases.

Most of the class had a very high coverage.

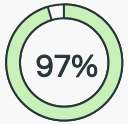
Github copilot helped with the test-cases for the new classes, but still this test-cases presented a lot of problems

Manual modification needed

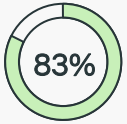
Testing II – Mutation Testing

PiTest

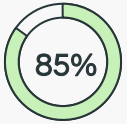
Mutation testing using default config for mutant operators



Line Coverage



Mutation Coverage



Test Strength

Strong Coverage, almost all the line covered with TC

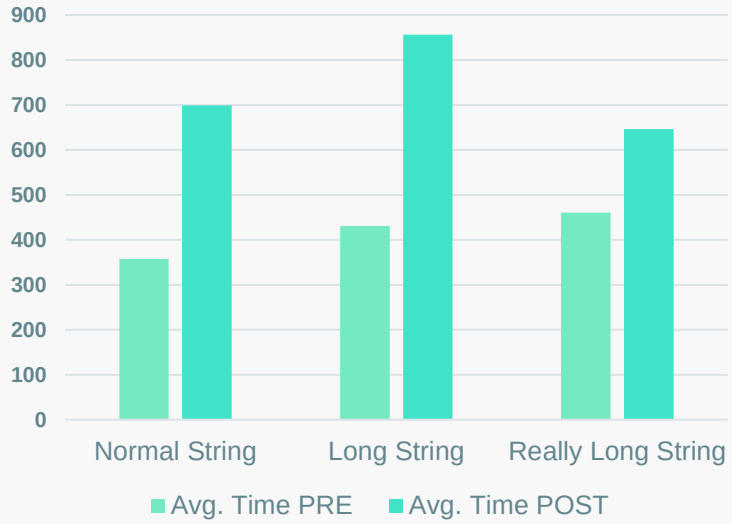
High robustness of test-cases able to catch errors

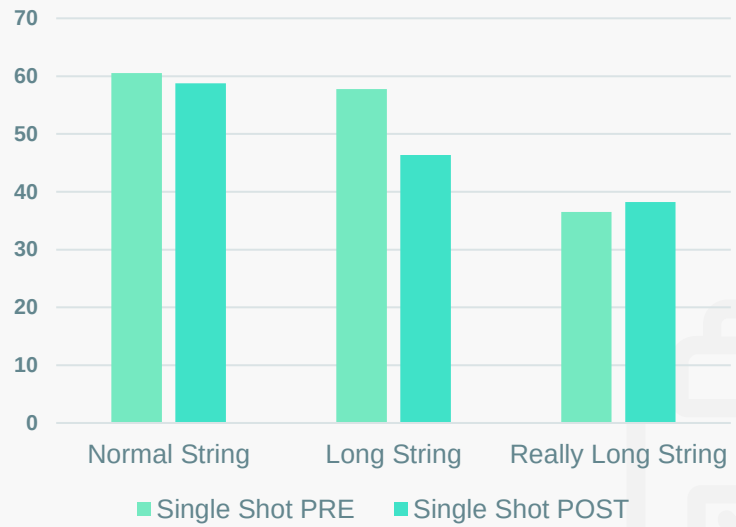
Test-cases are able to detect changes inside the code

Testing III – Performance Testing

JMH

Tested the performance of one of the method with high CC

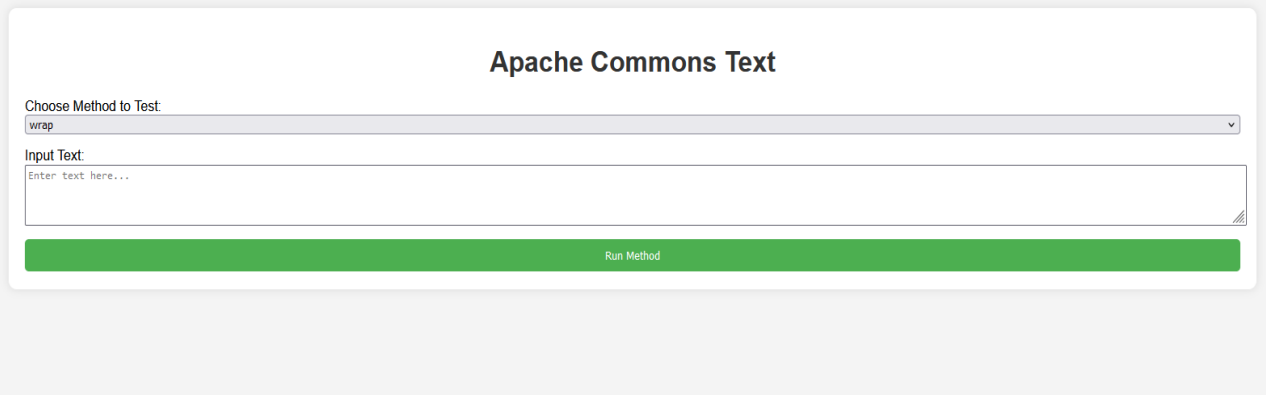




Containerization

Docker Image

Developed a Web-App to test methods of the library



Thanks!

Do you have any questions?

C.CALABRESE31@STUDENTI.UNISA.IT

https://github.com/Carmineh/commons-text/