UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Software Engineering and IT Management

Report

Professor

Dario Di Nucci

Student

Riccardo Imparato 0522501613

ACADEMIC YEAR 2023/2024

Index

1 Beginning

After exploring the site https://commons.apache.org/, the project about the CSV component has been selected, and forked.

Original project: https://github.com/apache/commons-csv

Forked project: https://github.com/r4004/commons-csv

2 Building the project

After adding all the missing dependencies in the pom.xml file, the project became buildable locally. After the bug fixing, the project will be built in CI/CD with GitHub Actions.

One of the needed dependencies was not from the Maven Repository, forcing us to include the library locally as Maven does not permit anymore downloads from external sources.

3 Bug fixing

The tool SonarCloud has been used to find the following problems:

- 7 bugs: 4 high, 3 medium.
- 658 code smells: 37 high, 69 medium, 552 low.

After many iterations between bug fixing and checking with SonarCloud, this is the final state of the program:

- 4 bugs: 1 high, 3 medium.
- 157 code smells: 37 high, 70 medium, 50 low.

It is worth to note that the majority of the problems revealed by SonarCloud are not actual issues. Here the explanations of the bugs ignored.

Bug	**Why it has been ignored**
Use or store the value returned from "read" instead of throwing it away.	To resolve this, we need to store the thrown value into an unused variable, but an unused variable will make the project unbuildable.
A "NullPointerException" could be thrown; "getHeaderMapRaw()" can return null.	The project uses the null values as feature. Changing this will require to refactor several code, and even to change some of the .txt files used for the tests.
Update this scope and remove the "systemPath".	Previously has been already mentioned the need to include a library locally. There is no way to avoid this.

4 Code coverage

JaCoCo has been used to create the coverage report, and sent the results to the site Codecov. The test coverage is of 98,35%.

5 Mutation test

With PiTest, 709 mutations were created, of which 677 (95%) were killed.

6 Performance test

Java Microbenchmark Harness has been used to benchmark. This is the benchmark created:

Benchmark	**Mode**	**Cnt**	**Score**	**Error**	**Units**
CSVBenchmark.parseCommonsCSV	avgt	20	3359,297	± 21,839	ms/op
CSVBenchmark.parseGenJavaCSV	avgt	20	3035,217	± 6,604	ms/op
CSVBenchmark.parseJavaCSV	avgt	20	1280,663	± 33,184	ms/op
CSVBenchmark.parseOpenCSV	avgt	20	1521,555	± 37,987	ms/op
CSVBenchmark.parseSkifeCSV	avgt	20	2080,360	± 45,026	ms/op
CSVBenchmark.parseSuperCSV	avgt	20	1712,673	± 37,787	ms/op
CSVBenchmark.read	avgt	20	198,199	± 4,087	ms/op
CSVBenchmark.scan	avgt	20	1158,116	± 31,365	ms/op
CSVBenchmark.split	avgt	20	971,231	± 23,498	ms/op

All of these benchmarks perform the same task: they all read the same file, and count the number of records within it. The only difference is the library used. The reason is obvious: the speed of the libraries is being compared.

7 Tests generated automatically

For the tests generated automatically, EvoSuite and Randoop have been used. Because these tools have been used outside Maven, the terminal commands will be shown here.

7.1 EvoSuite

EvoSuite terminal command was easy to use.

```
C:\Users\Omega\IdeaProjects>java -jar evosuite-1.0.6.jar -target commons-csv\target\classes
```

This is very easy to comprehend. These are the statistics created:

TARGET_CLASS	**criterion**	**Coverage**	**Total_Goals**	**Covered_Goals**
org.apache.commons.csv.Constants	*	0,0	2	0
org.apache.commons.csv.CSVFormat	*	0,830	3668	2788
org.apache.commons.csv.CSVParser	*	0,643	746	388
org.apache.commons.csv.CSVPrinter	*	0,703	351	224
org.apache.commons.csv.CSVRecord	*	0,743	315	205
org.apache.commons.csv.DuplicateHeaderMode	*	1,0	0	0
org.apache.commons.csv.ExtendedBufferedReader	*	0,838	649	496
org.apache.commons.csv.Lexer	*	0,709	1075	569
org.apache.commons.csv.QuoteMode	*	1,0	0	0
org.apache.commons.csv.Token	*	0,917	29	27

*the criterion is: LINE;BRANCH;EXCEPTION;WEAKMUTATION;OUTPUT;METHOD;METHODNOEXCEPTION;CBRANCH

7.2 Randoop

Randoop, instead, is much more convoluted.

```
C:\Program Files\Java\jdk1.8.0_202\bin>java -classpath C:\Users\Omega\IdeaProjects\commons-csv\target\classes;randoop-all-4.3.2.jar randoop.main.Main gentests --classlist=C:\Users\Omega\IdeaProjects\myClassList.txt --junit-output-dir=randoop-tests
```

Let’s analyze it bit by bit.

`C:\Program Files\Java\jdk1.8.0_202\bin>`	This is where the JDK is located. If the terminal was not positioned here, tools.jar could not be found. You can access your JDK position easily by typing “cd %JAVA_HOME%/bin”. Also, the terminal needed the administrator privileges.
`java -classpath C:\Users\Omega\IdeaProjects\commons-csv\target\classes;randoop-all-4.3.2.jar`	Here, classpath is getting 2 arguments separated by a semicolon. The first one is about the .class files location, while the second one is about the .jar file location.
`randoop.main.Main gentests --classlist=C:\Users\Omega\IdeaProjects\myClassList.txt --junit-output-dir=randoop-tests`	These are the commands given to the .jar file. The text file myClassList.txt contains the list of the classes.

This is the content of the file myClassList.txt:

```
org.apache.commons.csv.Constants
org.apache.commons.csv.CSVFormat
org.apache.commons.csv.CSVParser
org.apache.commons.csv.CSVPrinter
org.apache.commons.csv.CSVRecord
org.apache.commons.csv.DuplicateHeaderMode
org.apache.commons.csv.ExtendedBufferedReader
org.apache.commons.csv.Lexer
org.apache.commons.csv.QuoteMode
org.apache.commons.csv.Token
```

Randoop generated 6 regression tests.

8 Project security

For the project security analysis, FindSecBugs and OWASP DC have been used. Because these tools have been used outside Maven, the terminal commands will be shown here.

8.1 FindSecBugs

FindSecBugs terminal command was relatively easy.

```
PS C:\Users\Omega\IdeaProjects\FindSecBugs> .\findsecbugs.bat -low -progress -html -output ..\findsecbugs-report\report.html ..\commons-csv\target\classes
```

FindSecBugs has created an .html report. There was only 1 medium security warning.

8.2 OWASP DC

OWASP DC terminal command was easy.

```
PS C:\Users\Omega\IdeaProjects\dependency-check\bin> .\dependency-check.sh -s ..\..\commons-csv\target\
```

OWASP DC has created an .html report. There were 4 medium vulnerabilities, all from jquery-ui.min.js file.