**ES – Sprint 2 - Lines of Code Metrics**

This metrics is used to measure the size of a program by counting the number of lines in the text of the program’s source code, subdivided in Method metrics, Class metrics, Interface metrics, Package metrics, Module metrics, File type metrics and Project metrics. Each one is composed by the following metrics: CLOC (comment lines of code), JLOC (Javadoc lines of code), LOC (lines of code), NCLOC (Non-comment lines of code), RLOC (relative lines of code), CLOC (rec) – comment lines of code (recursive), JLOC (rec) – Javadoc lines of code (recursive), LOC(rec) – lines of code (recursive), LOCp (lines of product code, LOCp(rec) – lines of product code (recursive), LOCt (lines of test code), LOCt(rec) – lines of test code (recursive), NCLOCp (non-comment lines of code(product)), NCLOCp(rec) – non-comment lines of code (product, recursive), NCLOCt (non-comment lines of code (test)), L(Groovy) – lines of Groovy, L(HTML) – lines of HTML, L(J) – lines of Java, L(XML) – lines of XML, and the last one L(KT) – lines of Kotlin.

In this case we are only consider CLOC, JLOC and LOC in Class metrics.

* CLOC – comment lines of code

Consists in the number of lines of comments present in the class.

|  |  |  |
| --- | --- | --- |
| Class | CLOC | Path |
| HTMLUnicodeConversionMaps | 824 | src\main\java\org\jabref\logic\util\strings\HTMLUnicodeConversionMaps.java |
| BracketedPattern | 438 | src\main\java\org\jabref\logic\citationkeypattern\BracketedPattern.java |
| … | … | … |
| VersionPreferences | 0 | src\main\java\org\jabref\preferences\VersionPreferences.java |
| StyleTesterView | 0 | src\main\java\org\jabref\styletester\StyleTesterView.java |

This metric can indicate a code smell, showing that the class might be reviewed. The code smell could be, the class not having any comments, meaning that could be hard for someone else (or even the developers) to understand the code is doing or should be doing, or the class have too many comments, what could indicate that’s something wrong with the code or a bad design is being covered up. An example of this code smell was found and explain by the team member Martim Gouveia – 57482. When review the class, can be found comments that look like “reminder”, or comments that indicate that something needs to be done or that if a change is made to that section, another need to be updated, generating another code smell, that was found and explain by the team member Pedro Perdigão – 58165.

* JLOC – Javadoc lines of code

Consists in the number of lines of Javacode present in the class.

|  |  |  |
| --- | --- | --- |
| Class | JLOC | Path |
| BracketedPattern | 376 | src\main\java\org\jabref\logic\citationkeypattern\BracketedPattern.java |
| TreeNode | 300 | src\main\java\org\jabref\model\TreeNode.java |
| … | … | … |
| VersionPreferences | 0 | src\main\java\org\jabref\preferences\VersionPreferences.java |
| StyleTesterView | 0 | src\main\java\org\jabref\styletester\StyleTesterView.java |

The explanation in the last section applies to this, as well as the code smells explained and found.

* LOC – lines of code

Consists in the number of lines of code present in the class.

|  |  |  |
| --- | --- | --- |
| Class | LOC | Path |
| JabRefPreferences | 2291 | src\main\java\org\jabref\preferences\JabRefPreferences.java |
| BibtexParserTest | 1422 | src\main\java\org\jabref\logic\importer\fileformat\BibtexParserTest.java |
| … | … | … |
| JabrefIconProvider | 6 | src\main\java\org\jabref\gui\icon\JabrefIconProvider.java |
| LocalizationLocator | 6 | src\main\java\org\jabref\logic\l10n\LocalizationLocator.java |

This metric can indicate a code smell, showing that the class might be reviewed.

If this metric shows a hight number, like 2291 as we see on the previous table, maybe indicates a code smell for large classes, meaning that class has more responsibility that she needed. We can combat this code smell by creating new classes for pass some methods to those classes or just pass a few methods for other classes. This code smell was identified and explained by the team member Pedro Lourenço – 57577.

In another hand, if the metric shows a low number, like 6 as we see on the previous table, maybe indicates a code smell for a data class, meaning that class is too small, that means that only contain only data and no real functionality, only getter and setter methods. This code smell was identified and explained by the team member Martim Gouveia – 57482.