**Code Metrics:**

Complexity metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | CogC | ev(G) | iv(G) | v(G) |
| **org.jabref.cli.ArgumentProcessor.fetch(String)** | 9 | 4 | 6 | 6 |
| **org.jabref.cli.ArgumentProcessor.importAndOpenFiles()** | 28 | 1 | 16 | 16 |
| **org.jabref.cli.ArgumentProcessor.processArguments()** | 39 | 5 | 33 | 33 |

**Legend:**

CogC – Cognitive complexity

ev(G) – Essential cyclomatic complexity

iv(G) – Design complexity

v(G) – Cyclomatic complexity

Analysis of the collected metrics:

As we may observe, the first method has the lowest values, having a Cognitive complexity of 9, an Essential cyclomatic complexity of 4, a Design complexity of 6 and a Cyclomatic complexity of 6.

Cognitive Complexity is a measure of how difficult a unit of code is to intuitively understand. In the last two methods, the cognitive complexity is higher which means that the lines of code are more difficult to read. This could be a trouble spot because the code needs to be easy to read not just for who made the code itself but for other people to read it.

The second method has the lowest value for Essential cyclomatic complexity (1), Essential complexity is the measure of the degree to which a module contains unstructured constructs.

The numbers of Design complexity, a measure of the module’s decision structure as it relates to calls to other modules, and Cyclomatic complexity, measures the number of linearly independent pathsthrough a given program, are the same in each method.

Regarding the identified code smells, none of them reflect on this metrics.