CODEBASE METRICS

Lines of code Metrics

Lines of Code or LOC (also known as Source Lines of Code - SLOC) is a quantitative measurement in computer programming for files that contains code from a computer programming language, in text form. The number of lines indicates the size of a given file and gives some indication of the work involved.

LOC is literally the count of the number of lines of text in a file or directory.

The total number of lines in the project is 1230593.

An example of how the metrics are captured:



The count of lines is divided in categories:

* Method
* Class
* Interface
* Module
* Package
* FileType
* Project

The measures that are included are:

|  |  |
| --- | --- |
| **Comment lines of code** | **52788** |

|  |  |
| --- | --- |
| **Javadoc lines of code** | **42222** |

|  |  |
| --- | --- |
| **Lines of code** | **1230593** |

|  |  |
| --- | --- |
| **Lines of Groovy** | **1456** |

|  |  |
| --- | --- |
| **Lines of HTML** | **0** |
| **Lines of Java** | **173641** |
|  |  |
| **Lines of Kotlin** | **0** |
| **Lines of product code** | **221235** |
|  |  |
| **Lines of test code** | **0** |
| **Lines of XML** | **41144** |
|  |  |
| **Non-comment lines of code** | **1153319** |
|  |  |
| **Non-comment lines of code (product)** | **165004** |
|  |  |
| **Non-comment lines of code (test)** | **0** |

Complexity Metrics

A complexity measure is a [**cyclomatic complexity**](https://www.geeksforgeeks.org/cyclomatic-complexity/) in which the complexity of a module is the number of independent cycles in the [flow graph of the module](https://www.geeksforgeeks.org/software-engineering-control-flow-graph-cfg/). A complexity measure tries to capture the level of difficulty in understanding a module.

The value is given by this formula:



n1 (number of unique operators)

n2 (number of unique operands)

N2 (total frequency of operands)

An example of how the metrics are captured:



Here is a resume of all the measures taken divided by categories:

|  |  |
| --- | --- |
| **Class** | **28013,56702** |
| **Average operation complexity** | **2919,567022** |
| OCavg | 2919,567022 |
| **Maximum operation complexity** | **5527** |
| OCmax | 5527 |
| **Weighted method complexity** | **19567** |
| WMC | 19567 |
| **Method** | **71635** |
| **Cognitive complexity** | **15352** |
| CogC | 15352 |
| **Cyclomatic complexity** | **22195** |
| v(G) | 22195 |
| **Design complexity** | **19490** |
| iv(G) | 19490 |
| **Essential cyclomatic complexity** | **14598** |
| ev(G) | 14598 |
| **Module** | **22201,89145** |
| **Average cyclomatic complexity** | **1,891454375** |
| v(G)avg | 1,891454375 |
| **Total cyclomatic complexity** | **22200** |
| v(G)tot | 22200 |
| **Package** | **22552,78331** |
| **Average cyclomatic complexity** | **352,7833105** |
| v(G)avg | 352,7833105 |
| **Total cyclomatic complexity** | **22200** |
| v(G)tot | 22200 |
| **Project** | **22201,89145** |
| **Average cyclomatic complexity** | **1,891454375** |
| v(G)avg | 1,891454375 |
| **Total cyclomatic complexity** | **22200** |
| v(G)tot | 22200 |
| **Total general** | **166605,1332** |