УДК 004.412 DOI https://doi.org/10.35546/kntu2078-4481.2025.2.2.26

Л. М. МАКАРОВА
кандидат технічних наук, доцент,
доцент кафедри програмного забезпечення автоматизованих систем
Національний університет кораблебудування імені адмірала Макарова
ORCID: 0000-0003-2903-3001

М. А. ТАТАРЕНКО
аспірант кафедри програмного забезпечення автоматизованих систем
Національний університет кораблебудування імені адмірала Макарова
ORCID: 0009-0009-3330-0169

АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ ОТРИМАННЯ МЕТРИК ЧИДАМБЕРА І КЕМЕРЕРА З ПРОГРАМНИХ ПРОЕКТІВ, РОЗРОБЛЕНИХ МОВОЮ JAVA

Метрики Чидамбера та Кемерера (CK Metrics) – широко поширений набір метрик, призначений для вимірювання властивостей в об'єктно-орієнтованих програмних системах. Набір CK Metrics включає шість показників, кожен з яких дає уявлення про різні аспекти розробки та реалізації програмного забезпечення. Об'єктно-орієнтована мова програмування Java і досі залишається однією з основних для створення як серверних програмних систем, так і нативних мобільних застосунків для ОС Android та десктопних програми для стаціонарних ПК.

У даній роботі надається аналіз інструментів, які можна використовувати для отримання числових значень метрик з набору CK Metrics з програмних проектів, написаних із застосуванням мови програмування Java. Надано опис та коротку оцінку зручності використання наступних інструментів: ck, ckjm, ckjm-extended, CodeMR, Metrics Reloaded, Metrics Tree.

В основному представлені інструменти, які розповсюджуються безкоштовно, за виключенням CodeMR, який має обмежену безкоштовну та платну повнофункціональну версії. Представлені в аналізі інструменти можна поділити на дві групи с точки зору взаємодії з користувачем: через інтерфейс командного рядка і як плагіни для інтегрованого середовища розробки IntelliJ, яке де-факто можна назвати стандартом для розробки програмних систем, реалізованих мовою Java. Інструменти для взаємодії через графічний інтерфейс користувача в даному аналізі не представлені.

Результати аналізу допоможуть зробити правильний вибір інструментального засобу для отримання числових значень метрик з набору CK Metrics, виходячи з конкретних вимог розробника чи дослідника та досягти потрібного результату з меншою витратою часу на самостійне вивчення можливостей відповідного інструментального засобу.

Ключові слова: *метрики Чидамбера та Кемерера, об'єктно-орієнтовані метрики, метрики програмного забезпечення, вимірювання метрик програмного забезпечення, Java, Android, IntelliJ.*

L. M. MAKAROVA
PhD, Associate Professor at the Department of Automated Systems Software
Admiral Makarov National University of Shipbuilding
ORCID: 0000-0003-2903-3001

M. A. TATARENKO
Postgraduate Student at the Department of Automated Systems Software
Admiral Makarov National University of Shipbuilding
ORCID: 0009-0009-3330-0169

ANALYSIS OF TOOLS FOR OBTAINING CHIDAMBER AND KEMMERER METRICS FROM SOFTWARE PROJECTS DEVELOPED IN JAVA

Chidamber and Kemerer Metrics (CK Metrics) is a widely used set of metrics designed to measure properties in object-oriented software. The CK Metrics set includes six indicators, each of which provides representation into different aspects of software development and implementation. The object-oriented Java programming language still remains one of the main ones for creating both server-side software and native mobile applications for Android OS and desktop applications for stationary PCs.

This paper provides an analysis of tools that can be used to obtain numerical values of metrics from the CK Metrics set from software projects written using the Java programming language. A description and a brief assessment of the usability of the following tools are provided: ck, ckjm, ckjm-extended, CodeMR,, Metrics Reloaded, Metrics Tree.

The tools are mainly presented, which are distributed open source, with the exception of CodeMR, which has a limited free and paid full-featured version. The tools presented in the analysis can be divided into two groups in terms of user interaction: via the command line interface and as plugins for the IntelliJ integrated development environment, which can be de facto called the standard for developing software implemented in Java. Tools for interaction via the graphical user interface are not presented in this analysis.

The results of the analysis will help to make the right choice of a tool for obtaining numerical values of metrics from the CK Metrics set, based on the specific requirements of the developer or researcher and achieve the desired result with less time spent on independently studying the capabilities of the corresponding tool.

Key words: *CK Metrics, object-oriented metrics, software metrics, measuring software metrics, Java, Android, IntelliJ.*

Постановка проблеми

Отримання метрик програмного забезпечення (ПЗ) – дуже важливий і складний процес. Широко визнано, що визначення розміру чи прогнозування трудовитрат процесу розробки ПЗ – один із найбільш домінуючих аспектів оцінки витрат на його розробку. Вимірювання метрик ПЗ має важливе значення для досягнення основних цілей управління процесом розробки ПЗ: прогнозування, прогресу та покращення якості розробки.

Метрики ПЗ – це кількісні показники, які дозволяють виміряти різні властивості та характеристики ПЗ [1, 2]. Правильно обрані метрики можуть допомогти керівництву, інженерам, розробникам зосередитись на своїх цілях.

Обґрунтований вибір інструменту для отримання числових значень метрик з набору CK Metrics, може бути зроблений виходячи з конкретних вимог розробника чи дослідника.

Аналіз останніх досліджень і публікацій

В роботі [3] пропонується дослідження та розробка нового інструменту для розрахунку числових значень метрик для програм, розроблених мовою програмування Java. Інструмент реалізує розрахунок метрик з набору CK Metrics, метрик Абдіна (Abdeen), Дюкасса (Ducasse) та Сахрауї (Sahraoui), а також метрик модульності. Наголошується, що розроблюваний інструмент також повинен дозволяти користувачеві легко розширювати його шляхом додавання нових метрик. Запропонований інструмент в [3] заснований на бібліотеці аналізу байт-коду BCEL, яка призначена для того, щоб надати зручний спосіб аналізу, створення та маніпулювання бінарними файлами класів Java (тими, що мають розширення.class).

В роботі [4] надається короткий аналіз інструментальних засобів для отримання числових значень метрик, в основному розміру вихідного коду. Це такі інструменти, як: Analyst4j, Loc Metric, Code Counter, Source Monitor, Code Analyzer, Ecilpse Metric Plugin. В роботі можна побачити, що Analyst4j дозволяє отримувати числові значення деяких метрик з набору CK metrics. Треба відзначити, що посилання на офіційні сторінки багатьох з цих інструментів не працюють, а деякі з них, як, наприклад, Analyst4j взагалі неможливо знайти за допомогою пошукової системи.

В роботі [5] запропоноване порівняльне дослідження, що має на меті дати уявлення про особливості та можливості різних інструментів для отримання числових значень об'єктно-орієнтованих метрик. Мета роботи полягає в тому, щоб допомогти у виборі найбільш підходящого інструменту. Представлений аналіз близько двадцяти інструментів. Розглядаються як комерційні рішення, так і програми з відкритим вихідним кодом. Порівняння відображене в таблицях за різними критеріями, наприклад: підтримка мов програмування, різних метрик, ліцензування, варіанти підтримки вхідних та вихідних даних.

Формулювання мети дослідження

Мета дослідження – проаналізувати та систематизувати інструментальні засоби для отримання числових значень метрик ПЗ, в першу чергу з набору метрик Чидамбера та Кемерера (Chidamber & Kemerer, набір CK metrics) з програмних проектів, розроблених із застосуванням мови програмування Java.

Представлений аналіз таких інструментів, як: ck, ckjm, ckjm-extended, CodeMR, Metrics Reloaded, Metrics Tree.

Дослідження може бути корисним для розширення можливостей вибору інструментарію для тих дослідників та розробників, яким потрібно виконувати вимірювання шляхом отримання числових значень з вихідного коду, написаного мовою програмування Java.

Викладення основного матеріалу дослідження

Метрика ПЗ є мірою певної властивості, характеристики або ефективності програмного коду. Основна мета розрахунку метрик ПЗ полягає в отриманні відтворюваних та підданих кількісній оцінці вимірювань, які можна використовувати для прогнозування дефектів ПЗ, оптимізації продуктивності, планування графіка робіт, забезпечення якості, оцінки витрат, оцінки розміру нового проекту, що розробляється на основі даних отриманих з минулих схожих проектів, наприклад на основі регресійного аналізу.

Об'єктно-орієнтована (ОО) парадигма є однією з найпопулярніших у сучасній розробці програмних систем. ОО метрики застосовуються для вимірювання властивостей об'єктно-орієнтованого ПЗ. Багато з них були

розроблені для парадигм ОО підходу (абстракція, клас, об'єкт, успадкування тощо), для обчислення різних атрибутів, таких як якість, зв'язність і т.д. ОО методології вимагають значних зусиль на ранніх етапах життєвого циклу програмного проекту для ідентифікації об'єктів та класів, атрибутів та операцій, відносин між об'єктами та класами, інкапсуляції, наслідування та поліморфізму. Все це вимагає від дизайнерів ПЗ ретельного структурування проекту та розгляду взаємодії між об'єктами.

Якість ОО проектування визначають за допомогою різних метрик. Одні з базових та найпоширеніших – метрики Чидамбера та Кемерера [6–9], які пропонують 6 метрик, що розраховуються для кожного класу в проекті. Часто в різних публікаціях можна зустріти скорочення CK, що і означає, що йдеться про набір метрик Чидамбера та Кемерера. Це такі метрики, як: WMC (Weighted Methods per Class) – зважені методи на клас, DIT (Depth of Inheritance Tree) – глибина дерева успадкування, NOC (Number Of Children) – кількість підкласів даного класу, CBO (Coupling Between Object classes) – зв'язаність між класами об'єктів, RFC (Response For a Class) – відповідь для класу, LCOM (Lack of Cohesion in Methods) – відсутність зв'язності методів.

Мова програмування Java – одна з популярних мов для створення серверного ПЗ рівня підприємства (Java Enterprise Edition), десктопних програми для стаціонарних ПК і для написання нативних мобільних застосунків для ОС Android [10]. Навіть незважаючи на те, що ще однією мовою є відносно молода для екосистеми Android мова програмування Kotlin, Java і досі залишається популярною, особливо для підтримки вже існуючих нативних мобільних Android застосунків.

Далі розглянуті інструменти, які дозволяють отримувати числові значення метрик з набору CK Metrics з програм, розроблених мовою програмування Java. Слід зазначити, що деякі інструменти дозволяють розраховувати лише частину набору CK-метрик. Оскільки середовище розробки IntelliJ IDEA для мови програмування Java та розробки нативних мобільних застосунків є де-факто стандартом, то наводяться також плагіни для цього середовища, які дозволяють отримати CK метрики.

CK від розробника Маурісіо Аніче (Maurício Aniche) – популярний інструмент для отримання багатьох різних метрик із вихідного коду Java програм [11]. Цей інструмент у певному сенсі можна назвати еталонним, через те, що він часто використовується науковцями у своїх роботах. Це проект з відкритим вихідним кодом (Open source). Взаємодія з програмою відбувається через інтерфейс командного рядка (Command Line Interface, CLI). Цей інструмент зберігає результати аналізу тільки в файл в CSV форматі. Тому результати аналізу потрібно обробляти вручну за допомогою табличного процесора (MS Excel, LibreOffice Calc), або спеціалізованих програм. CK можна використовувати як автономну програму так і у вигляді бібліотеки для підключення у власний проект. CK охоплює всі аспекти об'єктно-орієнтованого програмування. І головне, що він дозволяє розраховувати всі ООП метрики набору CK Metrics.

ckjm. Скорочення від Chidamber та Kemerer Java Metrics. Автор цього інструменту Діомідіс Спінелліс (Diomidis Spinellis) створив консольну програму для обчислення всіх метрик Чидамбера та Кемерера з Java програм [12]. Крім того, Спінелліс включив до свого інструменту дві додаткові метрики, які не є частиною набору CK: CA (аферентні зв'язки) та NPM (кількість загальнодоступних методів). ckjm визначає значення метрик для кожного класу окремо. Анонімні класи обробляються як автономні об'єкти.

Інструмент є автономною консольною програмою написаною на Java. Тому для її запуску потрібен Java компілятор. ckjm – це також Open source проект з відкритим вихідним кодом, який знаходиться у GIT репозиторії автора. Результати обчислень відображаються безпосередньо у стандартному виводі. ckjm можна використовувати як завдання ant, а також безпосередньо генерувати вихідні дані у форматі XML. Це дозволяє виконати постобробку виводу XML за допомогою XSLT для створення гарно оформлених звітів.

Важливою особливістю цієї програми, є те що в якості вхідних даних потрібно подавати файли байт-коду (.class) у форматі JVM. Це створює проблему для Android проектів, де вихідними файлами є формат Dalvik (байт-код Dex). Однак класи Java зазвичай спочатку компілюються у формат JVM, а потім перетворюються в байт-код Dex. Отже, можна використовувати проміжні файли байт-коду JVM. Крім того, вхідні файли для ckjm мають бути правильно представлені у командному рядку. Це може включати додаткову роботу з вбудованими UNIX інструментами, такими як find, sed, а також безпосередньо з компілятором java. Це описано на сторінці документації.

Як пише автор інструменту ckjm на сторінці цього проекту, що він написав ckjm через розчарування відсутності надійних програм для розрахунку об'єктно-орієнтованих метрик Чидамбера та Кемерера. Програми, які він знаходив, розраховували лише деякі CK метрики, або давали невірні результати, або були вкрай неефективними – вимагали величезних ресурсів ПК. Тому створений ним проект Ckjm – простий та аскетичний інструмент, який слідує традиції UNIX – добре робити щось одне. Він не переглядає рекурсивно всі каталоги в пошуках файлів, які потрібно виміряти, не пропонує графічного інтерфейсу користувача, однак виконує свою роботу ретельно та ефективно, пропонуючи також і високу продуктивність.

З цього опису можна зрозуміти простоту ckjm. Проте перед його застосуванням обов'язково потрібно ознайомитись із документацією, представленою на сторінці автора ckjm.

ckjm-extended. Інструмент ckjm отримав подальший розвиток під назвою ckjm-extended [13]. Автор ckjm-extended, Маріан Юреско (Marian Jureczko) зазначає, що цей інструмент можна вважати наступною версією ckjm. Розвиток полягає в тому, що ckjm-extended обчислює додаткові метрики [14]. В іншому відмінностей майже немає. В якості вхідних даних також потрібно подавати скомпіловані файли байт-коду (.class) у форматі JVM. Результати роботи програми відображаються у стандартний вихідний потік (в консоль), або з можливістю експорту у формат XML.

Даний інструмент це також Open source проект з відкритим вихідним кодом, який доступний у GIT репозиторії автора [15], а також на його сайті [13].

Далі перераховані додаткові метрики, які розраховує ckjm-extended [14]: ефферентні зв'язки (Efferent coupling, Ce), кількість публічних методів для класу (Number of Public Methods for a class, NPM), відсутність зв'язності методів (Lack of COhesion in Methods, LCOM3), кількість рядків вихідного коду (Lines Of Code, LOC), доступ до даних (Data Access Metric, DAM), міра агрегації (Measure Of Aggregation, MOA), міра функціональної абстракції (Measure of Functional Abstraction, MFA), згуртованість між методами класу (Cohesion Among Methods of class, CAM), зв'язаність успадкування (Inheritance Coupling, IC), зв'язаність між методами (Coupling Between Methods, CBM), середня складність методу (Average Method Complexity, AMC), цикломатична складність Маккейба (McCabe's Cyclomatic Complexity, CC).

CodeMR – плагін для IDE (Integrated Development Environment) IntelliJ і Eclipse, що дозволяє розраховувати різні метрики для програм, написаних мовами програмування Java, Kotlin, Scala [16]. CodeMR є платним плагіном. Вартість річної ліцензії на травень 2025 року для індивідуальних користувачів становить 149 доларів США. Доступна версія для ознайомлення з повним функціоналом. Період ознайомлення складає 7 днів.

CodeMR сумісний з IDE Eclipse [17], а також з багатьма продуктами IntelliJ Platform Community та Ultimate Edition [16]. На сторінці завантаження доступні різні версії плагіна, які сумісні з різними версіями платформи IntelliJ, починаючи з IntelliJ 2017 і вище.

Плагін доступний на офіційному сайті [18], а також у магазині плагінів Jetbrains Marketplace для IDE IntelliJ [16]. Цей інструмент дозволяє розраховувати велику кількість різноманітних метрик. Важливо, що цей плагін дозволяє розраховувати всі метрики набору CK Metrics. Плагін підтримує набори метрик на рівні проекту, пакета, класу та методу.

Далі наведено рівні та деякі метрики, які дозволяє розраховувати CodeMR.

Для всіх рівнів (всього проекту, пакета, класу, методу): кількість рядків коду (LOC), набір метрик Холстеда, зручність обслуговування (maintainability).

Рівень проекту: загальна кількість рядків коду проекту, кількість пакетів, зовнішніх пакетів, зовнішніх сутностей.

Рівень пакету: загальна кількість рядків коду пакета, еферентний та аферентний зв'язок, кількість інтерфейсів, класів, сутностей, абстрактність, нестабільність.

Рівень класу: кількість рядків коду, зважені методи на клас (WMC), глибина дерева успадкування (DIT), кількість дочірніх класів (NOC), зв'язаність між класами об'єктів (CBO), відповідь для класу (RFC), відсутність зв'язності методів (LCOM), зв'язок між методами, кількість полів, методів, статичних полів, статичних методів, перевизначених методів, доступ до зовнішніх даних, відсутність тісної зв'язності класів.

Рівень методу: кількість рядків коду, цикломатична складність Маккейба, глибина вкладених блоків, кількість параметрів, викликаних методів, доступних полів.

Як можна бачити, розрахунок набору CK-метрик підтримується в повному обсязі.

У вікні структурованого деревоподібного вигляду (Project Outline) показаного на рис. 1, відображаються числові значення метрик елементів (класів, інтерфейсів, і т.д.) програмного проекту відкритого у IDE IntelliJ IDEA. Їх можна сортувати за стовпцями різних метрик, швидко переглядати найпроблемніші елементи та переходити до вихідного коду вибраного елемента.

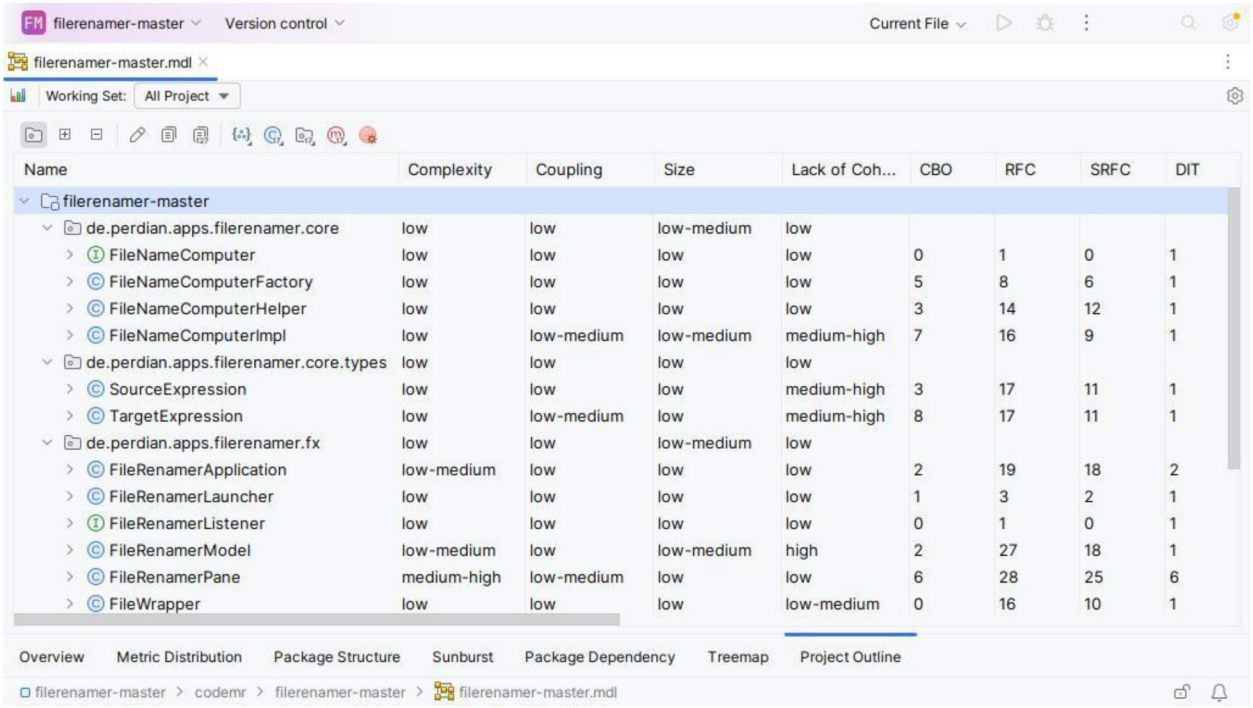
CodeMR представляє результати розрахунку метрик не тільки у вигляді числових значень, а й у вигляді різних варіантів візуалізацій. Далі наведено описи деяких варіантів візуалізацій.

– Структура пакета (Package Structure) у вигляді бульбашкової діаграми дозволяє розглянути повністю весь проект.

– Розподіл метрик (Metric Distribution) показує розподіл рівнів метрик у вигляді кругових діаграм. Метрики розподіляються за 5-ма рівнями: низький, низький-середній, середній-високий, високий, дуже-високий. На кругових діаграмах вони забарвлені різними кольорами. Зрізи кругових діаграм пропорційні до розміру коду класів відповідного рівня.

– Залежності пакетів (Package Dependency) дозволяє відповідно переглянути залежності між різними пакетами.

– Статистика проекту (Project Outline) надає можливість побачити всі метрики пакета, класу, методу та їх рівні якості у вигляді зведеної таблиці. Також можна вибрати лише потрібні метрики та переглядати інформацію щодо них.



Name	Complexity	Coupling	Size	Lack of Coh...	CBO	RFC	SRFC	DIT
filerenamer-master								
de.perdian.apps.filerenamer.core	low	low	low-medium	low				
FileNameComputer	low	low	low	low	0	1	0	1
FileNameComputerFactory	low	low	low	low	5	8	6	1
FileNameComputerHelper	low	low	low	low	3	14	12	1
FileNameComputerImpl	low	low-medium	low-medium	medium-high	7	16	9	1
de.perdian.apps.filerenamer.core.types	low	low	low	low				
SourceExpression	low	low	low	medium-high	3	17	11	1
TargetExpression	low	low-medium	low	medium-high	8	17	11	1
de.perdian.apps.filerenamer.fx	low	low	low-medium	low				
FileRenamerApplication	low-medium	low	low	low	2	19	18	2
FileRenamerLauncher	low	low	low	low	1	3	2	1
FileRenamerListener	low	low	low	low	0	1	0	1
FileRenamerModel	low-medium	low	low-medium	high	2	27	18	1
FileRenamerPane	medium-high	low-medium	low	low	6	28	25	6
FileWrapper	low	low	low	low-medium	0	16	10	1

Рис. 1. Вікно структурованого деревоподібного вигляду програмного проекту відкритого в IDE IntelliJ IDEA

– Візуалізація у вигляді графів (Graph Editor) спрощує розуміння та рефакторинг аналізуємої програмної системи.

Цікавою особливістю плагіна є відображення атрибутів якості таких як складність (complexity), зчеплення (coupling), перевикористання коду (reusability), читабельність (readability), зручності обслуговування (maintainability) у вигляді графа. Візуальними властивостями вузла графа є його колір, форма та розмір. Колір свідчить про складність сутності. Змінюється у колірній шкалі від синього до червоного. Чим вище складність – тим більше зміщення у бік червоного кольору. Коли значення атрибуту зв'язку класу збільшується, форма вузла графа змінюється на значок із великою кількістю кутів. Наприклад, трикутник, квадрат, п'яти кінцева зірка і т.д.. Клас з високим ступенем зв'язку має більше кутів, що вказують на більшу кількість точок взаємодії з іншими класами. Розмір вузла графа є розміром класу. Чим більший розмір класу, тим більше розмір вузла графа.

Даний плагін дозволяє генерувати HTML-звіти для всього проекту або кожного робочого набору (Working set). Робочий набір дозволяє розділити весь програмний проект на дрібні частини для більш наочного аналізу. У робочий набір можна помістити вибрані користувачем пакети та класи для подальшого аналізу. HTML звіт складається із чотирьох частин: інформаційна панель (Dashboard), докладний список метрик, пояснення метрик, графіки.

Metrics Reloaded. Плагін для IDE IntelliJ, що дозволяє розраховувати метрики кількості рядків коду та файлів для багатьох мов програмування, а також багато інших метрик для програм, написаних мовою програмування Java. Цей інструмент є проектом з відкритим вихідним кодом (Open Source) і доступний як у GitHub репозиторії автора [19], так і в магазині плагінів Jetbrains Marketplace для IDE IntelliJ [20].

Плагін сумісний з багатьма продуктами IntelliJ Platform Community та Ultimate Edition, серед яких IntelliJ IDEA, Android Studio, PhpStorm, WebStorm та інші. Для встановлення даного плагіну, в IDE IntelliJ в налаштуваннях конфігурації плагінів достатньо набрати MetricsReloaded. На сторінці завантаження доступні різні версії плагіна, які сумісні з різними версіями платформи IntelliJ, у тому числі досить старими.

Цей інструмент дозволяє розраховувати велику кількість різноманітних метрик. Важливо, що є можливість отримати значення всіх метрик з набору CK Metrics. Крім того, розраховуються й інші метрики, серед яких:

– Кількість класів (Class count). Інформація про загальну кількість класів, класів продукту (ті, що не є частиною тестів), та класів, які є частиною коду тестування;

– Залежності (Dependency). Детальні метрики по залежностям. Надається інформація про циклічні залежності та транзитивні залежності класів, а також про залежності кожного класу;

– Складність (Complexity). Подробиці цикломатичної складності, розширених цикломатичних залежностей методів;

– JUnit-тестування (JUnit Testing). Інформація про кількість тверджень (assertions) та методів тесту JUnit;

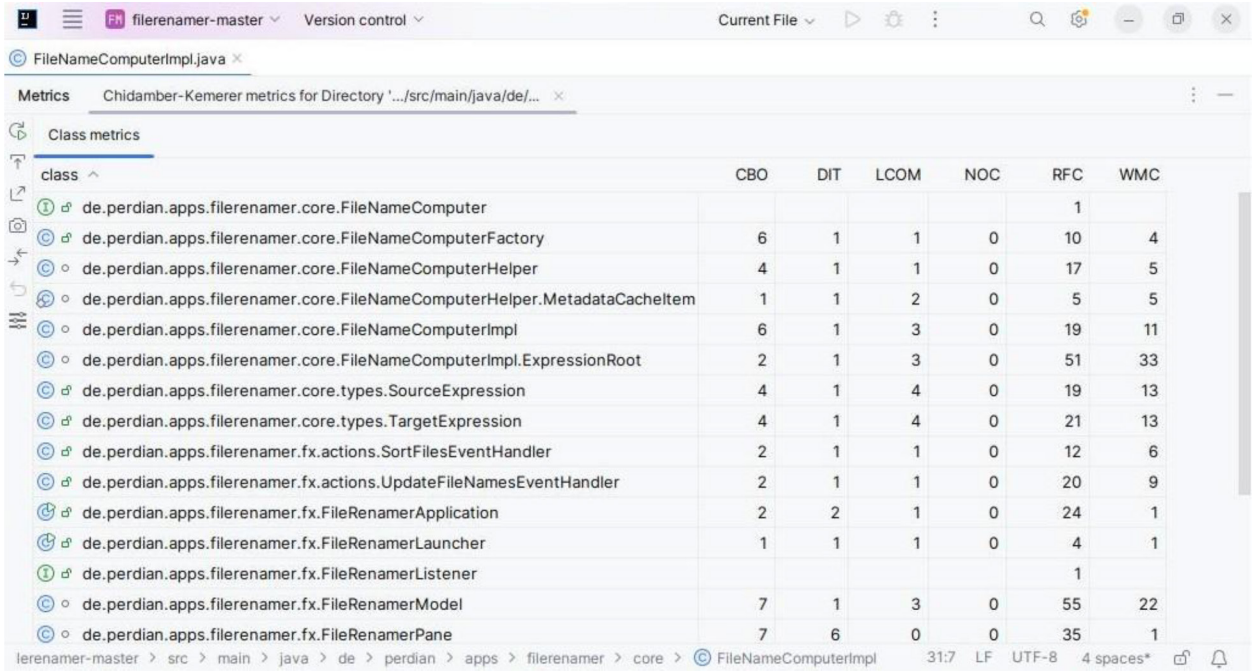
– Кількість документації Javadoc (Javadoc Coverage). Простий підрахунок кількості рядків документації Javadoc;

– Кількість рядів коду (Lines of Code, LOC). Підрахунок загальної кількості рядків коду, а також розподіл робочого (production) та тестового коду;

– Набір метрик MOOD (Model for Object Oriented Design), визначених Фернандо Бріто е Абреу (Fernando Brito e Abreu) і розроблених для того, щоб надати узагальнення загальної якості ОО проекту. Оригінальний набір метрик MOOD складається з 6 метрик;

– Метрики Роберта Мартіна. Вони відноситься до набору показників, які автор згадує у своїй книзі [21]. Це такі метрики, як: еферентне зчеплення (Efferent Coupling, Ce), аферентне зчеплення (Afferent Coupling, Ca), нестабільність (Instability, I), абстрактність (Abstractness, A), нормалізована відстань від головної послідовності (Normalized Distance from Main Sequence, D).

Metrics Reloaded дозволяє отримати метрики для всього проекту, конкретного модуля, поточного файлу або навіть для якоїсь частини проекту. Це робить його досить гнучким з огляду на те, що потрібно проаналізувати. На рис. 2 показний приклад отриманих числових значень набору CK Metrics для кожного класу проаналізованого проекту в IDE Intellij IDEA.



class	CBO	DIT	LCOM	NOC	RFC	WMC
de.perdian.apps.filerenamer.core.FileNameComputer					1	
de.perdian.apps.filerenamer.core.FileNameComputerFactory	6	1	1	0	10	4
de.perdian.apps.filerenamer.core.FileNameComputerHelper	4	1	1	0	17	5
de.perdian.apps.filerenamer.core.FileNameComputerHelper.MetadataCacheItem	1	1	2	0	5	5
de.perdian.apps.filerenamer.core.FileNameComputerImpl	6	1	3	0	19	11
de.perdian.apps.filerenamer.core.FileNameComputerImpl.ExpressionRoot	2	1	3	0	51	33
de.perdian.apps.filerenamer.core.types.SourceExpression	4	1	4	0	19	13
de.perdian.apps.filerenamer.core.types.TargetExpression	4	1	4	0	21	13
de.perdian.apps.filerenamer.fx.actions.SortFilesEventHandler	2	1	1	0	12	6
de.perdian.apps.filerenamer.fx.actions.UpdateFileNamesEventHandler	2	1	1	0	20	9
de.perdian.apps.filerenamer.fx.FileRenamerApplication	2	2	1	0	24	1
de.perdian.apps.filerenamer.fx.FileRenamerLauncher	1	1	1	0	4	1
de.perdian.apps.filerenamer.fx.FileRenamerListener					1	
de.perdian.apps.filerenamer.fx.FileRenamerModel	7	1	3	0	55	22
de.perdian.apps.filerenamer.fx.FileRenamerPane	7	6	0	0	35	1

Рис. 2. Вікно з отриманими числовими значеннями набору CK Metrics для кожного класу в IDE Intellij IDEA

Можна налаштувати свій набір лише тих метрик, які потрібно проаналізувати. Є можливість визначати порогові значення метрик, які можна використовувати для відображення тільки тих, які перевищують певний поріг (за замовчуванням відображається все). Розраховані числові значення метрик відображаються у табличному вигляді у самій IDE. Крім того, передбачена можливість переглянути діаграму розподілу або гістограму значень, кругову діаграму (там де це можливо).

Для того щоб розпочати роботу з Metrics Reloaded, перш за все потрібно виконати аналіз відкритого проекту у IDE Intellij IDEA. Для цього в меню вибираємо команду Code-Calculate metrics. Після цього в модальному вікні, в розділі Metrics scope вибираємо Цілий проект (Whole project), а в розділі Metrics profile, вибираємо один з предвстановлених профілів набору метрик. Профіль представляє собою просто набір тих чи інших метрик за якими програмний проект буде проаналізовано. Можна створити власний профіль додавши метрики, які підтримує плагін. На рис. 2 показані проаналізовані числові значення метрик з набору CK Metrics після вибору предвстановленого профілю Chidamber-Kemerer metrics.

Metrics Tree – плагін для IDE IntelliJ, що дозволяє розраховувати безліч різних метрик для програм, написаних мовою програмування Java. Цей інструмент є проектом з відкритим вихідним кодом (Open Source) і доступний як у GitHub репозиторії автора [22], так і в магазині плагінів Jetbrains Marketplace для IDE IntelliJ [23].

MetricsTree підтримує набори метрик на рівні проекту, пакета, класу та методу. Важливо, що цей інструмент дозволяє розраховувати значення всіх метрик із набору CK Metrics. Крім того, розраховуються і багато інших метрик коду.

Далі наведені рівні, на яких розраховуються ті чи інші метрики:

Для всіх рівнів (всього проекту, пакета, класу, методу): кількість рядків коду (LOC), набір метрик Холстеда, зручність обслуговування (index maintainability).

Для рівнів всього проекту та пакета: кількість не закоментованих ділянок коду, кількість звичайних, абстрактних, статичних класів, інтерфейсів.

Рівень проекту: набори метрик MOOD та QMOOD. Метрики QMOOD (Quality Model for Object Oriented Design) визначають зв'язок між атрибутами якості (такими як можливість повторного використання, функціональність, ефективність, зрозумілість, розширюваність, гнучкість).

Рівень пакету: набір метрик Роберта Мартіна.

Рівень класу: набори метрик Чідамбера-Кемерера, Лоренца-Кідда, Лі-Генрі, Ланца-Марінеску, Бімана-Канга.

Рівень методу: цикломатична складність МакКейба (CC), максимальна глибина вкладеності, глибина вкладеності циклів, глибина вкладеності умов, кількість циклів, локальність доступу до атрибутів (LAA), зовнішні постачальники даних (FDP), кількість доступних змінних (NOAV), інтенсивність зв'язку (CINT), дисперсія зв'язку (CDISP).

У вікні, показаному на рис. 3, відображаються числові значення різних метрик одного з класів проекту, відкритого у IDE IntelliJ IDEA.

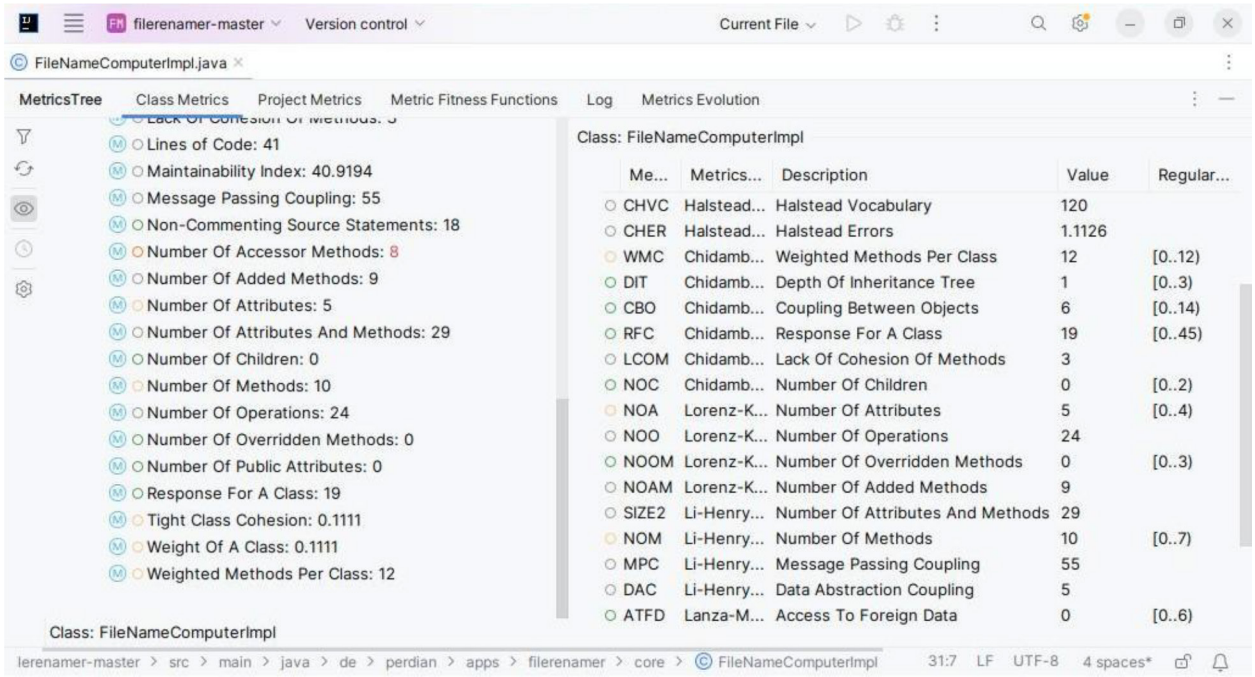


Рис. 3. Вікно з отриманими числовими значеннями різних метрик для одного класу в IDE Intellij IDEA

Даний інструмент представляє результати розрахунку метрик не тільки у вікні структурованого деревоподібного вигляду, як показано на рис. 3, а й у вигляді таблиць, дерев та деревоподібних карт. Підтримує контроль значень метрик, що обчислюються. Відображає різні властивості метрики (розподілу, кореляції) на діаграмах. Показує еволюцію значень метрик класу з урахуванням журналу git. Крім того, MetricsTree дозволяє знаходити загальні антипатерни і визначати нові.

Для того щоб розпочати роботу з Metrics Tree, перш за все потрібно відкрити панель інструментів плагіну Metrics Tree. Для цього в меню вибираємо команду View – Tool windows і в переліку знаходимо Metrics Tree. Даний плагін показує числові значення метрик з набору CK Metrics для відкритого Java класу в IDE Intellij IDEA. Попередній аналіз всього проекту, як у плагіні Metrics Reloaded робити не потрібно. Отже відкриваємо будь який Java клас з панелі проекту. У панелі інструментів плагіну, у вкладці Class metrics будуть відображені числові значення різних метрик, в тому числі з набору CK Metrics, як показано на рис. 3.

Передбачена можливість експорту числових значень метрик (в тому числі з набору CK Metrics) у CSV, або XML файл. Для цього переходимо на вкладку Project metrics. В цьому випадку потрібно виконати команду отримання метрик програмного проекту – натискаємо кнопку Calculate project metrics. Після цього можна виконувати експорт числових значень метрик.

Короткий порівняльний аналіз всіх інструментів представлений в таблиці 1.

Таблиця 1

Порівняльний аналіз всіх інструментів для отримання числових значень метрик набору CK Mertics

Назва інструменту	Підтримка мов програмування	Аналізуємі файли	Інтерфейс взаємодії з програмою	Перша версія інструменту (рік)	Ліцензія
CK	Java	Файли вихідного коду (.java)	CLI	2015	Apache 2.0 (Open source)
ckjm	Java	Байт код JVM (.class)	CLI	2005	Apache 2.0 (Open source)
ckjm-extended	Java	Байт код JVM (.class)	CLI	2009	Apache 2.0 (Open source)
CodeMR	Java, Kotlin, Scala	Проект в Intellij IDEA	Плагін Intellij IDEA	2018	Комерційна з пробною версією на 7 днів
Metrics Reloaded	Java	Проект в Intellij IDEA	Плагін Intellij IDEA	2004	Apache 2.0 (Open source)
Metrics Tree	Java	Проект в Intellij IDEA	Плагін Intellij IDEA	2020	Apache 2.0 (Open source)

Перш за все треба зазначити, що всі інструменти підтримують повний набір CK Metrics. З точки зору ліцензування, можна побачити, що всі програми, окрім CodeMR мають відкриту ліцензію і відносяться до категорії Open source, тобто є безкоштовними.

В науковому середовищі CK – достатньо розповсюджений інструмент. До переваг CK можна віднести велику спільноту розробників. Особливість CK в тому, що він зберігає результати аналізу тільки в файл в CSV форматі. Тому результати аналізу CK потрібно обробляти вручну за допомогою табличного процесора (MS Excel, LibreOffice Calc), або спеціалізованих програм.

До особливостей програм ckjm, ckjm-extended можна віднести формат вхідних даних – файли байт-коду (.class) у форматі JVM. Це може створити проблему при отриманні числових значень метрик з Android проектів. Та все ж незважаючи на це, треба звернути увагу на інструмент ckjm (ckjm-extended є продовженням ckjm) тому, що він був створений спеціально для розрахунку метрик з набору CK Metrics.

До недоліків інструментів CK, ckjm та ckjm-extended можна віднести взаємодію через інтерфейс командного рядка. Це може викликати труднощі у користувачів, які не звикли працювати з терміналом (або не мають досвіду роботи з терміналом).

CodeMR – платний (комерційний) плагін для IDE Intellij IDEA, але має повнофункціональну пробну версію на 7 днів. Переваги CodeMR: підтримка сучасних мов програмування, велика кількість підтримуваних метрик, надзвичайно велика кількість візуального відображення числових значень метрик, генерація HTML-звітів за результатами аналізу проекту, який відображається у веб-браузері. Треба наголосити, що CodeMR – достатньо потужний інструмент, і для його опанування потрібен деякий час.

Інструменти Metrics Reloaded та Metrics Tree приблизно однакові, але Metrics Tree має можливість візуального відображення числових значень метрик.

Висновки

В даній роботі акцент був зроблений на аналізі інструментів для отримання числових значень ОО метрик Чидамбера і Кемерера з програм, розроблених мовою програмування Java.

Результати дослідження можуть дозволити зробити обґрунтований вибір інструментального засобу для отримання числових значень метрик Чидамбера та Кемерера, виходячи з конкретних вимог розробника або дослідника. Також результати дослідження можуть допомогти досягти потрібного результату з меншою витратою сил на самостійне вивчення можливостей відповідного інструментального засобу.

З суб'єктивної точки зору авторів, найбільш функціональним та зручним у користуванні є плагін CodeMR. Далі приблизно однакове місце займають CK, Metrics Reloaded та Metrics Tree. За ними йдуть ckjm, ckjm-extended, які потребують деякої підготовки для роботи з ними.

Також маємо за необхідне рекомендувати перед застосуванням будь-якого інструменту ознайомитися з інтерпретацією метрик в офіційній документації. Однозначно виділити той чи інший інструмент – достатньо важке завдання. Все залежить від поставленої задачі та досвіду користувача.

Список використаної літератури

1. ISO, ISO/IEC TR 9126-2: Software Engineering – Product Quality – Part 2: External Metrics. Geneva, Switzerland: International Organization for Standardization, 2003.
2. ISO, ISO/IEC TR 9126-3: Software Engineering – Product Quality – Part 3: Internal Metrics. Geneva, Switzerland: International Organization for Standardization, 2003.
3. J.D. van Leusen. Tool support for software metrics calculation of Java projects. 2014. URL: https://fse.studenttheses.ub.rug.nl/11928/1/INF-BA-2014-J.D.van.Leusen.pdf (дата звернення: 20.05.2025).
4. Sandeep Kaur, Navjot Kaur. Software Metrics and Metric Tools – A Review. *International Journal on Recent and Innovation Trends in Computing and Communication*. Vol. 3, Issue: 4. 2015. URL: https://www.academia.edu/28255167/Software_Metrics_and_Metric_Tools_A_Review (дата звернення: 20.05.2025).

5. Kayarvizhy N. Systematic Review of Object Oriented Metric Tools. *International Journal of Computer Applications*. 135, no. 2 (2016): PP. 8–13. URL: https://www.ijcaonline.org/research/volume135/number2/n-2016-ijca-908269.pdf (дата звернення: 20.05.2025).

6. Shyam R. Chidamber, Chris F. Kemerer. Object-Oriented Metrics Suite. 2021. URL: https://www.aivosto.com/project/help/pm-oo-ck.html (дата звернення: 23.06.2023).

7. Shyam R. Chidamber, Chris F. Kemerer. A metrics suite for object oriented design. 1993. URL: https://www.eso.org/~tcsmgr/oowg-forum/TechMeetings/Articles/OOMetrics.pdf (дата звернення: 25.06.2023).

8. Chidamber S. R., Kemerer C. F. Towards a Metrics Suite for Object Oriented Design. *Conference Proceedings on Object-Oriented Programming Systems, Languages, and Applications*. Phoenix Arizona USA: ACM. PP. 197–211. 1991. URL: https://doi.org/10.1145/117954.117970. (дата звернення: 19.07.2024).

9. Chidamber and Kemerer object-oriented metrics suite. URL: https://www.aivosto.com/project/help/pm-oo-ck.html (дата звернення: 23.06.2023).

10. TIOBE Index for May 2025. The TIOBE Programming Community index is an indicator of the popularity of programming languages. 2025. URL: https://www.tiobe.com/tiobe-index/ (дата звернення: 24.05.2025).

11. Maurício Aniche. Ck – Code metrics for Java code by means of static analysis. 2024. URL: https://github.com/mauricioaniche/ck (дата звернення: 01.07.2023).

12. Spinellis, D. Ckjm – A Tool for Calculating Chidamber and Kemerer Java Metrics. *IEEE Software*. URL: https://www.spinellis.gr/sw/ckjm/doc/indexw.html (дата звернення: 01.07.2023).

13. Marian Jureczko. CKJM extended – An extended version of Tool for Calculating Chidamber and Kemerer Java Metrics (and many other metrics). Home page. 2024. URL: https://gromit.iiar.pwr.wroc.pl/p_inf/ckjm (дата звернення: 11.06.2024).

14. Marian Jureczko. CKJM extended – An extended version of Tool for Calculating Chidamber and Kemerer Java Metrics (and many other metrics). Introduction. 2024. URL: https://gromit.iiar.pwr.wroc.pl/p_inf/ckjm/intro.html (дата звернення: 11.06.2024).

15. Marian Jureczko. CKJM extended – An extended version of Tool for Calculating Chidamber and Kemerer Java Metrics (and many other metrics). 2024. URL: https://github.com/mjureczko/CKJM-extended (дата звернення: 11.06.2024).

16. CodeMR is a software quality and static code analysis tool for Java, Kotlin and Scala projects. JetBrains Marketplace. 2025. URL: https://plugins.jetbrains.com/plugin/10811-codemr (дата звернення: 23.05.2025).

17. CodeMR Static Code Analyser. Eclipse Marketplace. 2025. URL: https://marketplace.eclipse.org/content/codemr-static-code-analyser (дата звернення: 04.07.2024).

18. CodeMR – Measure, visualise and improve your code quality. Home page. 2025. URL: https://www.codemr.co.uk/ (дата звернення: 23.05.2025).

19. MetricsReloaded – Automated source code metrics plugin for IntelliJ IDEA. 2025. URL: https://github.com/BasLeijdekkers/MetricsReloaded (дата звернення: 23.05.2025).

20. MetricsReloaded – Automated Source Code Metrics for IntelliJ IDEA and IntelliJ Platform IDEs Plugin. 2025. URL: https://plugins.jetbrains.com/plugin/93-metricsreloaded (дата звернення: 23.05.2025).

21. Robert Cecil Martin. Agile Software Development: Principles, Patterns, and Practices. 2003. Prentice Hall PTR, USA.

22. MetricsTree is an IDE extension that helps to evaluate quantitative properties of Java code. GitHub repository. 2025. URL: https://github.com/b333vv/metricstree (дата звернення: 23.05.2025).

23. MetricsTree is an IDE extension that helps to evaluate quantitative properties of java code. JetBrains Marketplace. 2025. URL: https://plugins.jetbrains.com/plugin/13959-metricstree (дата звернення: 23.05.2025).

References

1. ISO, ISO/IEC TR 9126-2. (2003). Software Engineering – Product Quality – Part 2: External Metrics. Geneva, Switzerland: International Organization for Standardization.

2. ISO, ISO/IEC TR 9126-3. (2003). Software Engineering – Product Quality – Part 3: Internal Metrics. Geneva, Switzerland: International Organization for Standardization.

3. J. D. van Leusen. (2014). Tool support for software metrics calculation of Java projects. https://fse.studenttheses.ub.rug.nl/11928/1/INF-BA-2014-J.D.van.Leusen.pdf

4. Sandeep Kaur & Navjot Kaur. (2015). Software Metrics and Metric Tools – A Review. *International Journal on Recent and Innovation Trends in Computing and Communication*. Vol. 3 Issue 4. https://www.academia.edu/28255167/Software_Metrics_and_Metric_Tools_A_Review

5. Kayarvizhy N. (2016). Systematic Review of Object Oriented Metric Tools. *International Journal of Computer Applications*. 135, no. 2. pp. 8–13. https://www.ijcaonline.org/research/volume135/number2/n-2016-ijca-908269.pdf

6. Shyam R. Chidamber & Chris F. Kemerer. (2021). Object-Oriented Metrics Suite. https://www.aivosto.com/project/help/pm-oo-ck.html

7. Shyam R. Chidamber & Chris F. Kemerer. (1993). A metrics suite for object oriented design. https://www.eso.org/~tcsmgr/oowg-forum/TechMeetings/Articles/OOMetrics.pdf

8. Chidamber S. R. & Kemerer C. F. (1991). Towards a Metrics Suite for Object Oriented Design. *Conference Proceedings on Object-Oriented Programming Systems, Languages, and Applications*. Phoenix Arizona USA: ACM. pp. 197–211. https://doi.org/10.1145/117954.117970

9. Chidamber and Kemerer object-oriented metrics suite. https://www.aivosto.com/project/help/pm-oo-ck.html

10. TIOBE Index for May 2025. The TIOBE Programming Community index is an indicator of the popularity of programming languages. (2025) https://www.tiobe.com/tiobe-index/

11. Maurício Aniche. (2024). Ck – Code metrics for Java code by means of static analysis. https://github.com/mauricioaniche/ck

12. Spinellis, D. (2011). Ckjm – A Tool for Calculating Chidamber and Kemerer Java Metrics. IEEE Software. https://www.spinellis.gr/sw/ckjm/doc/indexw.html

13. Marian Jureczko. (2024). CKJM extended – An extended version of Tool for Calculating Chidamber and Kemerer Java Metrics (and many other metrics). Home page. https://gromit.iiar.pwr.wroc.pl/p_inf/ckjm

14. Marian Jureczko. (2024). CKJM extended – An extended version of Tool for Calculating Chidamber and Kemerer Java Metrics (and many other metrics). Introduction. https://gromit.iiar.pwr.wroc.pl/p_inf/ckjm/intro.html

15. Marian Jureczko. (2024). CKJM extended – An extended version of Tool for Calculating Chidamber and Kemerer Java Metrics (and many other metrics). https://github.com/mjureczko/CKJM-extended

16. CodeMR is a software quality and static code analysis tool for Java, Kotlin and Scala projects. JetBrains Marketplace. (2025). https://plugins.jetbrains.com/plugin/10811-codemr

17. CodeMR Static Code Analyser. Eclipse Marketplace. (2025). https://marketplace.eclipse.org/content/codemr-static-code-analyser

18. CodeMR – Measure, visualise and improve your code quality. Home page. (2025). https://www.codemr.co.uk/

19. MetricsReloaded – Automated source code metrics plugin for IntelliJ IDEA. (2025). https://github.com/BasLeijdekkers/MetricsReloaded

20. MetricsReloaded – Automated Source Code Metrics for IntelliJ IDEA and IntelliJ Platform IDEs Plugin. (2025). https://plugins.jetbrains.com/plugin/93-metricsreloaded

21. Robert Cecil Martin. (2003). Agile Software Development: Principles, Patterns, and Practices. Prentice Hall PTR, USA.

22. MetricsTree is an IDE extension that helps to evaluate quantitative properties of java code. GitHub repository. (2025). https://github.com/b333vv/metricstree

23. MetricsTree is an IDE extension that helps to evaluate quantitative properties of java code. JetBrains Marketplace. (2025). https://plugins.jetbrains.com/plugin/13959-metricstree