

**РОЗРОБКА ПРОГРАМИ
ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ JAVASCRIPT-КОДУ
НА ОСНОВІ МЕТРИК СТАТИЧНОГО АНАЛІЗУ**

Куцевич А.М.

e-mail: anna.kutsevych@nure.ua

Науковий керівник – к.т.н., проф. Бондарєв В.М.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

This paper presents the development of a program for assessing the quality of students' JavaScript code using static analysis metrics. The proposed system analyzes multiple solutions of the same programming task and calculates structural characteristics of the code. The analysis is based on abstract syntax tree parsing and includes metrics such as code length, nesting depth, number of variables, functions, loops, average identifier length, and code repetition. These metrics are used to compute an integrated program quality score. The tool allows comparing different implementations of the same task and identifying more readable and structured solutions. The proposed approach can be used in educational environments to support automated assessment of assignments.

У процесі навчання програмуванню важливо оцінювати не тільки правильність роботи програми, але й якість програмного коду. Студенти часто виконують однакові завдання різними способами. При цьому програми можуть давати правильний результат, але відрізнятися за структурою, читабельністю та складністю.

Дуже важливим є формування у студентів навичок написання зрозумілого та структурованого програмного коду. Якісний код легше підтримувати, модифікувати та розширювати. Тому в навчальному процесі варто звертати увагу не лише на результат виконання програми, але й на особливості реалізації алгоритмів. JavaScript і TypeScript залишаються одними з найбільш поширених мов у сучасній розробці програмного забезпечення [1].

У сучасній розробці програмного забезпечення широко використовуються інструменти статичного аналізу коду, які дозволяють перевіряти програмний код без його виконання та виявляти потенційні проблеми або недоліки [2]. Використання таких інструментів сприяє підвищенню якості програмних систем та зменшенню кількості помилок на етапі розробки.

З огляду на зазначені проблеми, метою роботи є розробка програми для оцінки якості JavaScript-коду, яка дозволяє аналізувати рішення одного й того самого завдання, написані різними студентами, та оцінювати їх якість на основі метрик статичного аналізу.

У більшості систем автоматичної перевірки програмних завдань оцінюється лише правильність результату виконання програми. Для цього використовуються тестові дані, за допомогою яких перевіряється відповідність отриманого результату очікуваному.

Однак такий підхід не враховує якість програмного коду. Програма може працювати правильно, але містити надмірну складність, дублювання коду або нечитабельні назви змінних. Подібні проблеми можуть ускладнювати підтримку та подальший розвиток програмного забезпечення. Крім того, складний або неструктурований код може бути важким для розуміння іншими розробниками.

Дослідження показують, що структура коду, його складність та стиль написання можуть суттєво впливати на якість програмних систем [3]. Крім того, аналіз програмних проєктів показує, що складність коду та неякісна структура можуть бути пов'язані з більшою кількістю помилок у програмному забезпеченні [4].

Існуючі інструменти статичного аналізу, наприклад ESLint, орієнтовані переважно на перевірку стилю програмування та пошук типових помилок [2]. Проте вони не завжди дозволяють отримати узагальнену оцінку якості коду або порівнювати різні реалізації одного алгоритму.

Таким чином, виникає необхідність у створенні інструменту, який дозволить аналізувати програмний код студентів за набором метрик та формувати узагальнений показник його якості. Такий підхід дозволяє формалізувати процес оцінювання програмних рішень.

Для розв'язання поставленої задачі було розроблено програму для аналізу якості JavaScript-коду. Її робота базується на використанні методів статичного аналізу програм.

На першому етапі виконується попередня обробка програмного коду. З нього видаляються коментарі та порожні рядки. Це дозволяє зосередитися саме на структурі програмного коду та його логіці. Завдяки цьому зменшується обсяг неінформативних даних, а результати аналізу стають більш точними та якісними.

На наступному етапі програмний код перетворюється у абстрактне синтаксичне дерево (AST). Для цього використовується бібліотека Acorn. Такий підхід дозволяє аналізувати структуру програми незалежно від форматування тексту коду.

Після побудови синтаксичного дерева стає можливим виконати детальний аналіз структури програми. На цьому етапі обчислюється набір метрик, що дозволяють отримати кількісні характеристики якості програмного коду [5]. У розробленій програмі використовуються такі метрики:

- кількість рядків коду;
- глибина вкладеності умовних операторів і циклів;
- кількість змінних;
- кількість функцій;

- кількість циклів;
- середня довжина імен змінних та функцій;
- кількість повторюваних фрагментів коду.

Ці показники дозволяють оцінити складність програми, її структуру та читабельність. Наприклад, надмірна глибина вкладеності може ускладнювати розуміння коду, а повторювані фрагменти можуть свідчити про відсутність оптимізації або рефакторингу.

Після обчислення всіх метрик формується інтегральна оцінка якості програмного коду. Для цього кожна метрика нормалізується та отримує певну вагу. Підсумковий показник обчислюється як зважена сума окремих метрик.

Такий підхід дозволяє аналізувати рішення одного й того самого завдання, написані різними студентами, та порівнювати їх між собою. Результатом роботи системи є звіт, який містить значення метрик та підсумкову оцінку якості програмного коду.

Розроблений інструмент може використовуватися у навчальному процесі для аналізу рішень студентів та підвищення об'єктивності оцінювання програмних завдань. Застосування метрик дозволяє звернути увагу не лише на правильність результату, але й на структуру, складність та читабельність програмного коду. Подібні системи також можуть сприяти розвитку навичок самостійного аналізу програмного коду. Студенти можуть бачити результати оцінювання своїх програм та визначати можливі шляхи покращення їхньої структури.

У подальшій роботі планується розширити набір метрик та вдосконалити алгоритми аналізу програмного коду. Також можливим напрямом розвитку є інтеграція програми для оцінки якості JavaScript коду з навчальними платформами для автоматичної перевірки програмних завдань.

Список використаних джерел:

1. Stack Overflow Developer Survey 2025. Stack Overflow. 2025. URL: <https://survey.stackoverflow.co/2025/> (дата звернення: 09.03.2026).
2. ESLint Documentation. ESLint: The Pluggable JavaScript Linter. URL: <https://eslint.org/docs/latest/> (дата звернення: 09.03.2026).
3. Bogner J., Merkel M. To Type or Not to Type? A Systematic Comparison of the Software Quality of JavaScript and TypeScript Applications on GitHub. IEEE Access. 2022.
4. Wang Z., Li X., Jiang Y. An Empirical Study on Bugs in TypeScript Programming Language. IEEE Transactions on Software Engineering. 2025.
5. Gruzdo I., Kyrychenko I., Tereshchenko G., Shanidze N. Metrics Applicable for Evaluating Software at The Design Stage // CEUR Workshop Proceedings. 2021. Vol. 2870. P. 916–936.