

# КЛАСИФІКАЦІЯ КЛАСІВ ЗАСОБІВ РЕФАКТОРИНГУ ПРОГРАМНОГО КОДУ ТА КРИТЕРІЇ ЇХНЬОГО ВИБОРУ ДЛЯ ВИСОКОЕФЕКТИВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Вінницький національний технічний університет

## **Анотація**

У роботі систематизовано класи засобів рефакторингу програмного коду та запропоновано систему критеріїв їхнього порівняння і вибору з орієнтацією на потреби високоефективних інформаційних систем. Показано, що різні класи інструментів (IDE-рушії, платформи статичного аналізу, rule/recipe-based засоби масових перетворень, AST-бібліотеки, інструменти autofix, детектори рефакторингів, LLM-асистенти) відрізняються за рівнем семантичних гарантій, масштабованістю, інтегрованістю у CI/CD та відтворюваністю. Запропоновано практично застосовну схему вибору інструментарію залежно від типу задачі рефакторингу та обмежень процесу супроводу.

**Ключові слова:** рефакторинг, засоби рефакторингу, статичний аналіз, технічний борг, CI/CD, rule-based перетворення, AST-трансформації, LLM.

## **Abstract**

The paper systematizes classes of source-code refactoring tools and proposes a set of criteria for their comparison and selection, oriented toward the needs of high-performance information systems. It is shown that different tool classes (IDE engines, static analysis platforms, rule/recipe-based large-scale transformation tools, AST libraries, autofix tools, refactoring detectors, and LLM assistants) differ in the level of semantic guarantees, scalability, CI/CD integration, and reproducibility. A practically applicable tool-selection scheme is proposed depending on the type of refactoring task and the constraints of the maintenance process.

**Keywords:** refactoring, refactoring tools, static analysis, technical debt, CI/CD, rule-based transformations, AST transformations, LLM.

## **Актуальність**

Зі зростанням розміру кодових баз підвищується частота появи «запахів коду» (code smells) і зростають витрати на супровід. Рефакторинг є контрольованою технікою покращення внутрішньої структури без зміни зовнішньої поведінки програми [1]. У високоефективних інформаційних системах це зумовлює потребу обґрунтованого вибору засобів рефакторингу. Тому доцільним є формування класифікації класів засобів рефакторингу та критеріїв їхнього вибору для типових задач супроводу.

## **Мета дослідження**

Розробити класифікацію класів засобів рефакторингу та систему критеріїв їхнього порівняння; запропонувати схему вибору класів засобів рефакторингу для типових задач рефакторингу у високоефективних інформаційних системах з урахуванням ризиків, масштабу змін і вимог CI/CD.

## **Класи засобів рефакторингу**

Класифікацію побудовано за двома узагальненими ознаками: механізм формування перетворень (каталожний, на основі правил, пошуково-оптимізаційний, генеративний) та спосіб контролю коректності (семантичні обмеження і передумови, відтворюваність застосування, інструментальна валідація). Узагальнення методів автоматизованого рефакторингу наведено в огляді [8]. Виділено шість класів:

**С1. IDE-рушії рефакторингу.** Інтерактивні засоби каталожних перетворень (перейменування, виділення методу тощо), що спираються на семантичну інформацію та перевірки передумов, забезпечуючи вимогу збереження поведінки програми [1; 8].

**C2. Платформи статичного аналізу та контролю якості.** Засоби, що формують формалізовані підстави для рефакторингу через виявлення «запахів коду», оцінювання «технічного боргу» та застосування порогів якості. У вискоєфективних інформаційних системах ці засоби застосовують також у CI/CD. У цій роботі C2 розглядається як інструмент постановки цілей та обмежень рефакторингу, а не як виконавець перетворень.

**C3. Засоби автоматизованих перетворень коду на основі правил.** Засоби, у яких перетворення коду задаються формалізованими правилами та виконуються відтворювано. До класу належать: C3.1 — «рецептурні» системні перетворення для масштабних змін у кодовій базі; C3.2 — правила з авто виправленням для локальних шаблонних змін.

**C4. Засоби програмних трансформацій на рівні AST.** Засоби аналізу й переписування коду на рівні абстрактного синтаксичного дерева (AST) для реалізації доменно-специфічних перетворень та дослідницьких прототипів.

**C5. Пошуково-орієнтований рефакторинг (search-based).** Засоби, у яких рішенням є послідовність рефакторингових перетворень, а її добір здійснюється методами оптимізаційного пошуку за однією або кількома цільовими функціями якості [2; 3; 4; 5].

**C6. Генеративні засоби на основі великих мовних моделей (LLM).** Засоби, що формують кандидатні зміни коду на основі природномовних специфікацій або прикладів; прийнятність результатів вимагає зовнішньої інструментальної валідації (компіляція, тести, обмеження області змін) [6; 7].

### Критерії порівняння та вибору

Запропоновано критерії:

**K1. Збереження поведінки.** Наявність механізмів семантичного контролю перетворень (перевірки передумов, узгоджене оновлення зв'язків) або необхідність зовнішньої валідації (компіляція, тести) як основного засобу підтвердження коректності [1; 6; 7].

**K2. Масштаб застосування.** Характер області впливу перетворень: локальні зміни (фрагмент/метод/клас) чи системні перетворення в кодовій базі, що охоплюють множину модулів/підсистем.

**K3. Відтворюваність і керованість застосування.** Можливість повторюваного виконання перетворень як формалізованої процедури (правило, специфікація перетворення). Для практичного впровадження це узгоджується з використанням у CI/CD.

**K4. Метрико-орієнтована постановка задач.** Ступінь формалізації цілей та обмежень рефакторингу через метрики якості (показники підтримуваності, «технічного боргу», порогові значення).

**K5. Альтернативність рішення.** Чи має задача множину припустимих варіантів рефакторингу з різними значеннями цільових показників, що зумовлює необхідність пошуку (зокрема багатокритеріальної оптимізації) [2; 3; 4; 5].

### Відповідність класів типовим задачам

Вибір класу засобів доцільно обґрунтовувати через тип задачі рефакторингу та критерії K1–K5.

1. Локальні каталожні перетворення з вимогою збереження поведінки (перейменування, виділення/вбудовування, зміни сигнатур). Доцільні засоби класу C1 (K1, K2) [1].

2. Постановка та пріоритизація задач рефакторингу за показниками якості («запахи коду», «технічний борг», пороги якості). Доцільні засоби класу C2 (K4).

3. Стандартизовані автоматизовані перетворення коду (масштабні міграції та локальні шаблонні виправлення). Доцільні засоби класу C3, з уточненням C3.1 для системних перетворень і C3.2 для локальних змін (K2, K3).

4. Доменно-специфічні перетворення та нетипові правила реструктурування. Доцільні засоби класу C4 (K2, K3).

5. Оптимізація якості шляхом добору послідовності перетворень. За наявності альтернативних рішень і компромісів (K5) доцільні засоби класу C5 [2; 3; 4; 5].

6. Генерація кандидатних змін для складних випадків, що слабо покриваються каталогами або правилами. Доцільні засоби класу C6 за умови зовнішньої валідації (K1, K3) [6; 7].

## Висновки

Запропонована класифікація C1-C6 і критерії K1-K5 утворюють основу для формалізованого вибору засобів рефакторингу за вимогами до збереження поведінки, масштабу перетворень, відтворюваності застосування та метрико-орієнтованої постановки задач. Особливо виділено засоби пошуково-орієнтованого рефакторингу (C5), що забезпечують добір послідовності перетворень у задачах з альтернативними рішеннями [2; 3; 4; 5].

## Перспективи подальших досліджень

Доцільними є: (1) формалізація цільових функцій та обмежень у задачах пошуково-орієнтованого рефакторингу, зокрема на основі генетичних алгоритмів [2; 3; 4; 5]; (2) удосконалення процедур валідації для генеративних засобів на основі LLM із поєднанням компіляційного контролю, тестування та обмеження області змін [6; 7]; (3) уточнення критеріїв придатності класів засобів для високоефективних інформаційних систем на підставі оглядових узагальнень [8].

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Fowler M. Refactoring: Improving the Design of Existing Code. – 2nd ed. – Boston : Addison-Wesley, 2019. – 448 с.
2. O’Keeffe M., Ó Cinnéide M. Search-based refactoring: An empirical study // Journal of Software Maintenance and Evolution: Research and Practice. – 2008. – Vol. 20, No. 5. – С. 345–364.
3. O’Keeffe M., Ó Cinnéide M. Search-based refactoring for software maintenance // Journal of Systems and Software. – 2008. – Vol. 81, No. 4. – С. 502–516.
4. Kebir S., Borne I., Meslati D. A genetic algorithm-based approach for automated refactoring of component-based software // Information and Software Technology. – 2017. – Vol. 88. – С. 17–36.
5. Saheb Nasagh R., Shahidi M., Ashtiani M. A fuzzy genetic automatic refactoring approach to improve software maintainability and flexibility // Soft Computing. – 2021. – Vol. 25, No. 6. – С. 4295–4325.
6. Liu B., Jiang Y., Zhang Y., Niu N., Li G., Liu H. Exploring the potential of general-purpose LLMs in automated software refactoring: An empirical study // Automated Software Engineering. – 2025. – Vol. 32. – Article 26.
7. Shirafuji A., Oda Y., Suzuki J., Morishita M., Watanobe Y. Refactoring programs using large language models with few-shot examples // Proceedings of the 30th Asia-Pacific Software Engineering Conference (APSEC 2023). – 2023. – С. 151–160.
8. Abid C., Alizadeh V., Kessentini M., do Nascimento Ferreira T., Dig D. 30 Years of Software Refactoring Research: A Systematic Literature Review // IEEE Transactions on Software Engineering. – 2020.

**Кузнець Ілля Ігорович** – студент групи F2-125а, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: [illiakuznets@gmail.com](mailto:illiakuznets@gmail.com)

**Ліщинська Людмила Броніславівна** – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [lhb@vntu.edu.ua](mailto:lhb@vntu.edu.ua)

**Kuznets Illia Ihorovych** – student of group F2-125a, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [illiakuznets@gmail.com](mailto:illiakuznets@gmail.com)

**Lishchynska Lyudmyla Bronislavivna** – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [lhb@vntu.edu.ua](mailto:lhb@vntu.edu.ua)