

УДК 004

Шаманський Р.В.¹, Зайко Т.А.²

¹ студ. гр. КНТ-124м НУ «Запорізька політехніка»

² доц. НУ «Запорізька політехніка»

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ВДОСКОНАЛЕННЯ ВИХІДНОГО КОДУ

Об'єкт дослідження – процес вдосконалення вихідного програмного коду.

Предмет дослідження – програмні засоби для вдосконалення вихідного коду.

Мета роботи – дослідження та розробка програмного забезпечення для вдосконалення вихідного коду.

Матеріали, методи та технічні засоби: мова програмування JavaScript, бібліотека для створення інтерфейсів React.

В ході виконання роботи було проаналізовано та досліджено процес вдосконалення вихідного програмного коду.

Визначено, що вдосконалення вихідного коду являє собою систематичний процес покращення його якості, структури та ефективності без зміни основної функціональності. Такий підхід дозволяє зробити

програмний продукт більш зрозумілим, надійним і стійким до помилок, а також полегшує подальшу підтримку та розвиток. Методи вдосконалення вихідного коду охоплюють кілька напрямів. Рефакторинг дозволяє оптимізувати внутрішню структуру, роблячи її більш зрозумілою та логічною. Використання автоматизованих інструментів аналізу сприяє виявленню синтаксичних і логічних помилок, а також підвищує безпеку. Сукупність таких методів формує основу для створення надійного, оптимізованого та зручного у подальшому розвитку програмного забезпечення.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для вдосконалення вихідного коду. Проте деякі програмні засоби для вдосконалення вихідного коду є орієнтованими на певні мови програмування чи конкретні середовища, що зменшує універсальність і робить їх менш ефективними у випадку багатоплатформних рішень. Іншим недоліком є ймовірність появи нових помилок у процесі рефакторингу або оптимізації, особливо якщо інструменти використовуються без належного тестування. Крім того, застосування таких програмних засобів може вимагати значних ресурсів, як обчислювальних, так і часових, що знижує ефективність при роботі з великими проектами. Також важливо враховувати, що деякі програмні засоби вдосконалення вихідного коду не завжди забезпечують гнучкість, оскільки їхні алгоритми ґрунтуються на загальних правилах і не враховують унікальні вимоги конкретного програмного проєкту. Тому актуальною є розробка програмного забезпечення для вдосконалення вихідного коду.

Сформульовано функціональні вимоги до програмного забезпечення для вдосконалення вихідного коду.

Для реалізації програмного забезпечення для вдосконалення вихідного коду обрано мову програмування JavaScript, що обґрунтовується універсальністю та підтримкою у більшості сучасних середовищ. Гнучкість синтаксису та підтримка асинхронних операцій роблять JavaScript ефективним інструментом для роботи з потоками даних і зовнішніми ресурсами. Екосистема мови стрімко розвивається, а велика кількість додаткових бібліотек і фреймворків значно розширює спектр можливостей і скорочує час реалізації складних проєктів. Активна спільнота забезпечує постійне вдосконалення стандартів і появу нових рішень, що дозволяє швидко адаптувати мову до сучасних потреб розробників.

Для створення програмного забезпечення для вдосконалення вихідного коду обрано бібліотеку React, що поєднує продуктивність, гнучкість і зручність у використанні. Компонентна архітектура React дозволяє створювати інтерфейси, що легко масштабуються й підтримуються навіть у складних проєктах. Крім того, забезпечується ефективне оновлення

елементів, що дає можливість працювати з великими обсягами даних без зниження швидкодії. Розвинена екосистема бібліотек і численні готові рішення дозволяють швидко додавати необхідний функціонал, наприклад інтеграцію з редакторами коду, підсвічування синтаксису чи інструментами візуалізації.

Новизна роботи полягає у тому, що запропоновано метод динамічного маркування коду, який полягає у поєднанні коментарів з метаданими та автоматизованого оновлення цих даних під час виконання програм, що дозволяє підвищити прозорість програмного коду, спростити його оптимізацію та забезпечити гнучку підтримку на всіх етапах життєвого циклу.

Практична цінність роботи полягає у тому, що розроблено програмне забезпечення для вдосконалення вихідного коду.

Запропоновано структуру програмного забезпечення для вдосконалення вихідного коду. Визначено, що структура програмного забезпечення для вдосконалення вихідного коду організована таким чином, щоб забезпечити чіткий поділ обов'язків між серверною логікою, клієнтським інтерфейсом і спільними інструментами, а також надати можливість здійснювати автоматизований аналіз тексту програм високого рівня з метою пошуку і подальшого усунення проблемних рішень, що негативно впливають на продуктивність або стійкість роботи коду.

Описано функціонування програмного забезпечення для вдосконалення вихідного коду. Функціонування програмного забезпечення для вдосконалення вихідного коду подано за допомогою схеми як послідовний процес перетворення та інтерпретації, у якому задіяні кілька рівнів обробки. Показано, що при функціонуванні програмного забезпечення для вдосконалення вихідного коду послідовні етапи трансформації JSX-описів у JavaScript-файли поєднуються з роботою інтерпретатора та нативного механізму браузера, що забезпечує як оптимізовану обробку, так і мінімальні оновлення у візуальному середовищі.

Розроблено базу даних програмного забезпечення для вдосконалення вихідного коду, яка є основою для збереження всієї інформації, що виникає в процесі роботи із вдосконаленням вихідного коду. Вона виконує роль центрального сховища, де поєднуються як початкові дані у вигляді вихідних текстів програм, так і вторинні артефакти, отримані після запуску алгоритмів автоматичного аналізу. У базі даних фіксується стан коду, відображаються знайдені недоліки, рекомендації щодо покращення та зміни, які були зроблені розробниками після внесення виправлень. Схема бази даних програмного забезпечення для вдосконалення вихідного коду відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Основними сутностями бази даних обрано такі: проєкт (Project),

Файл вихідного коду (SourceFile), Фрагмент коду (CodeSnippet), Аналіз (AnalysisRun), Виявлена проблема (Issue), Рекомендація (Recommendation), Зміна коду (CodeChange) та інші.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для вдосконалення вихідного коду.

Виконано тестування розробленого програмного забезпечення для вдосконалення вихідного коду. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з вдосконаленням програмного коду.

Галузь використання – вдосконалення програмного коду.