

- [3] N. Schärli, S. Ducasse, O. Nierstrasz та A. P. Black, “Traits: Composable Units of Behaviour”, у *ECOOP 2003 – Object-Oriented Programming*. Berlin, Heidelberg: Springer Berl. Heidelb., 2003, с. 248–274.
- [4] P. Tesone, S. Ducasse, G. Polito, L. Fabresse та N. Bouraqadi, “A new modular implementation for stateful traits”, *Sci. Comput. Program.*, т. 195, с. 102470, верес. 2020.
- [5] D. Cassou, S. Ducasse та R. Wuyts, “Traits at work: The design of a new trait-based stream library”, *Comput. Lang., Syst. & Struct.*, т. 35, № 1, с. 2–20, квіт. 2009.
- [6] J. Owens, “Dotnet Source Generators in 2024”, *Medium*, 1 жовт. 2024. [Онлайн]. Доступно: <https://posts.specterops.io/dotnet-source-generators-in-2024-part-1-getting-started-76d619b633f5>

УДК: 004.4

ПРОЄКТУВАННЯ АЛГОРИТМІВ РОБОТИ З ТЕКСТОВИМИ ДАНИМИ НА МОВІ ПРОГРАМУВАННЯ C#

Поліщук Д.С., Собко О.В., Залуцька О.О.

(dimvpolisuk@gmail.com, olenasobko.ua@gmail.com, zalutska.olha@gmail.com)

Хмельницький національний університет (Україна)

У роботі розглянуто процес проєктування алгоритмів для роботи з текстовими даними мовою програмування C#. Основну увагу приділено побудові алгоритмів збереження, відкриття та перевірки правопису текстових файлів у середовищі .NET. Запропоновані рішення базуються на використанні потоків введення-виведення, діалогових вікон вибору файлів та методів лексичної перевірки з використанням словникових структур.

Одним із ключових завдань сучасного програмування є ефективна організація роботи з текстовими даними [1], що включає збереження, відкриття, редагування та перевірку правопису. Проблема полягає у розробленні оптимальних алгоритмів, які забезпечують швидке виконання операцій з текстовими файлами [2] при мінімальних витратах ресурсів і збереженні цілісності даних.

Метою дослідження є проєктування та реалізація алгоритмів для роботи з текстовими файлами у програмному застосунку мовою C#.

Для досягнення мети поставлено такі завдання: розробити алгоритми збереження текстового файлу з контролем шляху запису, відкриття текстового файлу з перевіркою доступності ресурсу, перевірки правопису з використанням лексичного словника, а також виконати програмну реалізацію запропонованих алгоритмів.

Поставлені у дослідженні завдання передбачається реалізувати на прикладі розробленого текстового редактора, який виступає об'єктом практичної апробації проєктованих алгоритмів. Алгоритм збереження файлу реалізується через послідовність логічних дій, що починається з ініціації користувачем команди «Зберегти» або «Зберегти як» (рисунки 1).

Програма перевіряє, чи існує файл із вказаною назвою, і залежно від результату виконує або первинне збереження через діалогове вікно вибору шляху, або перезапис за раніше визначеною адресою. У разі відсутності імені файлу система генерує виняткову ситуацію, сповіщаючи користувача про необхідність заповнення обов'язкових полів. Такий алгоритм забезпечує цілісність даних і запобігає втраті інформації під час запису.

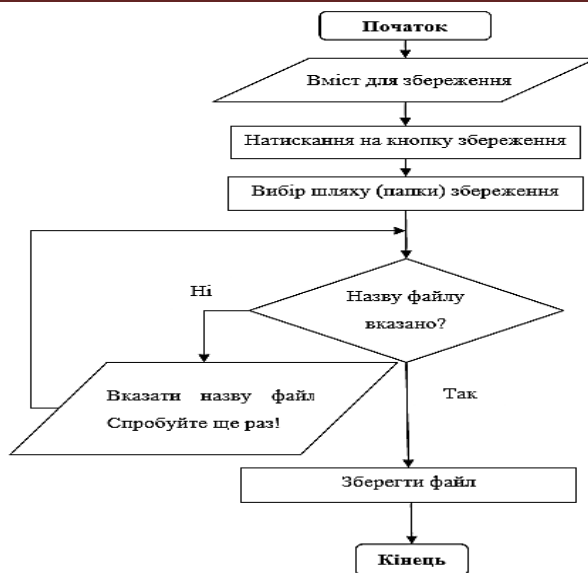


Рисунок 1 – Алгоритм збереження текстового файлу

Аналогічні принципи застосовуються при відкриванні файлу: після натискання на кнопку «Відкрити» програма за допомогою діалогового вікна зчитує вибраний користувачем файл, визначає його тип і сумісність із системою. Алгоритм подано на рисунку 2.

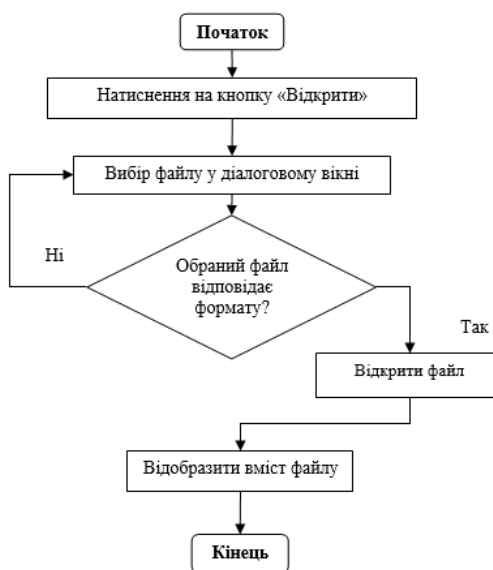


Рисунок 2 – Алгоритм відкривання текстового файлу

У випадку з текстовими файлами з розширенням .txt здійснюється пряме відображення вмісту, тоді як формати .doc, .docx або .rtf відкриваються з урахуванням структурованих параметрів шрифту, кольору та форматування. Це дозволяє зберігати цілісність документів і підтримувати можливість їх подальшого редагування.

Важливою складовою функціональності редактора є перевірка правопису, що виконується в реальному часі під час набору тексту (рисунок 3). Для цього текстовий редактор використовує вбудований словник, з яким порівнює кожне введене слово. У разі невідповідності слово підсвічується червоною лінією або іншим маркером, а користувачеві може бути запропоновано варіант виправлення чи синонім. Така система не лише підвищує якість текстового матеріалу, але й автоматизує процес редагування, роблячи його зручним та інтуїтивно зрозумілим.

Усі наведені алгоритми, реалізовані мовою програмування C# у середовищі .NET, формують основу роботи текстового редактора.

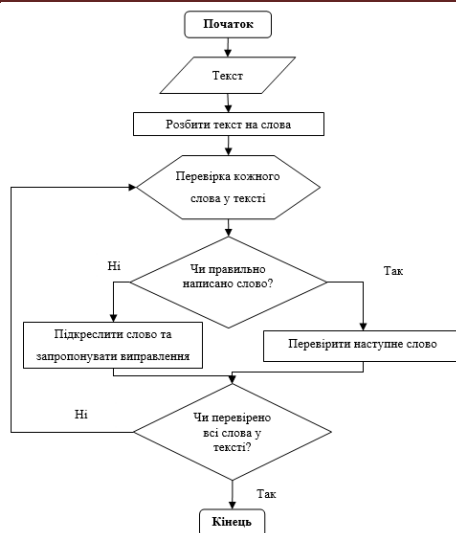


Рисунок 3 – Алгоритм перевірки правопису у текстовому файлі

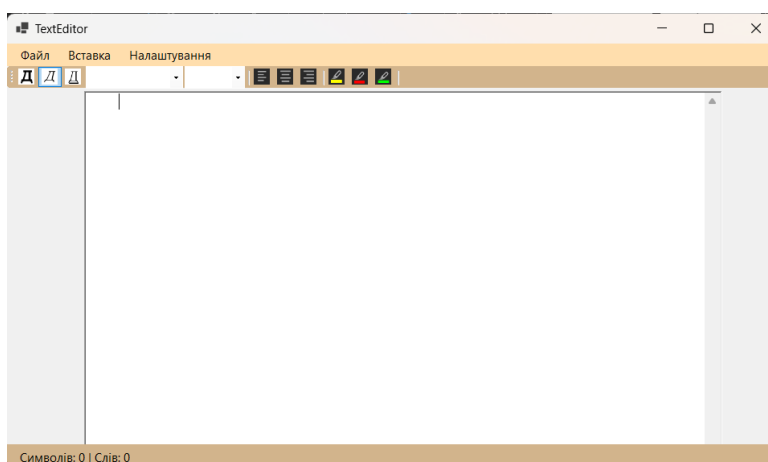


Рисунок 4 – Програмна реалізація текстового редактора

У результаті проведеного дослідження було спроектовано та описано основні алгоритми роботи з текстовими даними мовою програмування C#, що охоплюють процеси збереження, відкриття та перевірки правопису текстових файлів. Реалізація цих алгоритмів на прикладі текстового редактора підтвердила їх ефективність і коректність функціонування.

Список використаних джерел

- [1] N. Ostapchenko, O. Tyschenko, B. Denysenko, and O. Mazurets, “Semantic search of relevant images using vector databases,” Khmelnytskyi National University Institutional Repository, 2025. [Online]. Available: <https://elar.khmnmu.edu.ua/items/deafefe1-b21c-41df-9668-a3476a39e0ae> [Accessed: Oct. 22, 2025].
- [2] О. В. Мазурець, О. В. Ковальчук, В. О. Слободзян, «Використання спеціалізованих програмних розширень для автоматизації роботи з цифровими документами навчальних матеріалів», Інституційний репозитарій Хмельницького національного університету, 2018. [Online]. Available: <https://elar.khmnmu.edu.ua/items/5ebf8fb3-495e-4302-a15d-127634697766> [Accessed: Oct. 22, 2025].