

## ПРОГРАМНИЙ ЗАСТОСУНОК ДЛЯ ОЦІНЮВАННЯ СКЛАДНОСТІ ТЕКСТІВ

Вінницький національний технічний університет

### Анотація

У роботі представлено результати розробки програмного забезпечення “Text Analyzer Expert”, призначеного для автоматизованого оцінювання складності технічних текстів. Система реалізує статистичні алгоритми розрахунку індексів Флеша-Кінкейда та Ганніга, а також визначає допоміжні показники — середню довжину речення, кількість складів, складних слів і прогнозований час читання. Для створення програмного продукту використано мову Python і бібліотеку Tkinter, що забезпечило кросплатформність, простоту інтерфейсу та автономність роботи. Проведено тестування на зразках технічної документації, яке показало точність розрахунків і стабільність функціонування програми.

**Ключові слова:** читабельність тексту; оцінювання складності; Python; індекс Флеша-Кінкейда; індекс Ганніга; Tkinter.

### Abstract

The paper presents the development of “Text Analyzer Expert” software designed for automated evaluation of technical text complexity. The system implements statistical algorithms for calculating Flesch–Kincaid and Gunning Fog indexes as well as supplementary metrics such as average sentence length, number of syllables, complex word ratio, and estimated reading time. The application was implemented in Python using the Tkinter library, ensuring cross-platform compatibility, a simple interface, and full offline functionality. Testing on samples of technical documentation confirmed the accuracy of computations and stable performance of the program.

**Keywords:** text readability; complexity evaluation; Python; Flesch–Kincaid index; Gunning Fog index; Tkinter.

### Вступ

Сучасна технічна документація містить значні обсяги спеціалізованої інформації, зрозумілість якої безпосередньо впливає на ефективність використання програмних і апаратних засобів. Автоматизація оцінювання читабельності таких текстів дозволяє підвищити якість документації та скоротити час її редакторської обробки. Особливої актуальності ця проблема набуває в умовах стрімкого розвитку інформаційних технологій та зростання обсягів технічного контенту. Ручний аналіз складності текстів є трудомістким і суб’єктивним, що обумовлює необхідність застосування формалізованих підходів. Використання математичних моделей читабельності дозволяє отримати об’єктивні кількісні показники та стандартизувати процес оцінювання.

### Основна частина

Метою даної роботи є розробка автономного програмного застосунку, призначеного для автоматизованого оцінювання складності технічних текстів на основі кількісних лінгвістичних показників. У ході дослідження було здійснено аналіз сучасних підходів до визначення читабельності текстів та існуючих програмних рішень, зокрема Hemingway Editor, Readable, ProWritingAid і LanguageTool, що дозволило визначити їх переваги та обмеження. За результатами аналізу обрано статистичний підхід, який базується на використанні формалізованих моделей оцінювання, а саме індексів Флеша-Кінкейда та Ганніга [1, 2], що забезпечують отримання об’єктивних і відтворених результатів.

У процесі реалізації застосунку розроблено алгоритмічне забезпечення для обробки текстових даних, яке включає токенізацію, визначення кількості складів, ідентифікацію складних слів, а також обчислення середньої довжини речень. Програмна реалізація виконана з використанням мови програмування Python та стандартної бібліотеки Tkinter [3, 4], що дозволило забезпечити кросплатформність, автономність функціонування та зручність взаємодії з користувачем. Інтерфейс застосунку передбачає використання візуальних засобів індикації рівня складності тексту, підтримку темного режиму, а також можливість імпорту та експорту текстових даних.

Експериментальна перевірка працездатності програмного забезпечення здійснювалася на вибірках технічних текстів різної складності українською та англійською мовами. У результаті роботи застосунок формує структурований аналітичний звіт, який містить числові показники читабельності, допоміжні лінгвістичні характеристики та узагальнений висновок щодо рівня складності тексту. Узагальнені результати експериментального дослідження наведено в таблиці 1, що підтверджує ефективність запропонованого підходу та підвищує об'єктивність оцінювання і зручність подальшого аналізу.

Таблиця 1 – Порівняльна характеристика існуючих сервісів читабельності

Критерії порівняння	Hemingway Editor	Readable	ProWritingAid	Запропоноване рішення
Підтримка укр. мови	Так	Ні	Так	Так
Індекси читабельності	Ні	Так	Ні	Так
Робота офлайн	Ні	Так	Ні	Так
Експорт звітів	Так	Частково	Частково	Так

Детальний аналіз даних, наведених у таблиці 1, дозволяє виявити специфіку кожного з програмних рішень. Hemingway Editor забезпечує простий та наочний інтерфейс, проте підтримка української мови та розширені метрики читабельності відсутні. Readable і ProWritingAid реалізують обчислення ключових індексів і додаткових метрик, проте не підтримують автономну роботу офлайн і частково обмежені українською мовою. Розроблений застосунок “Text Analyzer Expert” об'єднує всі перелічені можливості: підтримує українську та англійську мови, працює автономно, надає повний набір метрик читабельності та формує структуровані звіти, що усуває необхідність використання кількох програм для комплексного аналізу складності технічних текстів.

### Висновки

Розроблений застосунок забезпечує об'єктивне, швидке та безпечне оцінювання технічних текстів без використання зовнішніх сервісів. Завдяки поєднанню математичних метрик і зручного інтерфейсу “Text Analyzer Expert” може застосовуватися технічними письменниками, редакторами й студентами для підвищення якості професійної документації.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flesch R. A New Readability Yardstick // Journal of Applied Psychology. – 1948. – Vol. 32, № 3. – P. 221–233. – Дата звернення: 21.03.2026 р.
2. Gunning R. The Technique of Clear Writing. – New York: McGraw-Hill, 1952. – 289 p. Wood G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Yellow Paper, 2014. 39 p.
3. Tkinter – Python interface to Tcl/Tk [Електронний ресурс]. – Режим доступу: <https://python.org/>. – Дата звернення: 21.03.2026 р.
4. Войтко В. В., Черноволик Г. О. Методичні вказівки до виконання курсових проєктів з дисципліни “Проектний практик ум”. – Вінниця: ВНТУ, 2022. – 44 с.

**Майданюк Володимир Павлович** – кандидат технічних наук, доцент кафедри програмного забезпечення, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: maidaniuk2000@gmail.com.

**Плахотнюк Владислав Дмитрович** – студент групи 6ПІ-226, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: daniil.medencev01@gmail.com.

**Maidaniuk Volodymyr** – Candidate of Technical Sciences, Associate Professor of the Department of Software Engineering, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.

**Plakhotniuk Vladyslav** – student of group 6PI-22b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.