

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ЦИКЛОВА КОМІСІЯ
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи освітнього ступеня «фаховий молодший бакалавр»
за спеціальністю 122 «Комп'ютерні науки»

на тему:

«Текстовий редактор з підтримкою форматування»

Виконав здобувач освіти групи П-42
Лейченко Володимир Олександрович

Керівник роботи:
Устименко Людмила Миколаївна

Рецензент:
Устименко Ярослав Іванович

Зміст

Вступ	6
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ	8
1.1. Постановка основної задачі розробки	8
1.2. Огляд та порівняння аналогічних систем.	9
1.3. Вибір та обґрунтування технологій розробки	14
Висновки до розділу 1	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА	19
2.1. Вибір алгоритмів та їх ефективність	19
2.2. Спроектовані класи програми	22
2.3. Програмна реалізація	27
Висновки до розділу 2	33
РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	35
3.1. Мінімальні вимоги до системи	35
3.2. Інтерфейс користувача	36
3.3. Інструкція для роботи з програмою	39
Висновки до розділу 3	43
ВИСНОВКИ	45
Перелік літературних джерел	47
Додаток А.	50

					ДР.122.042.026.ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Лейченко В О			Текстовий редактор з підтримкою форматування	Літ.	Арк.
Перевірив		Устименко Л.М.					3
Рецензент		Устименко Я.І.				Аркушів	55
Н. Контр.						ЖАТФК	
Затвердив.		Лавріщев О.О.					

РЕФЕРАТ

Обсяг: 55 сторінок, 9 рисунків, 1 додаток, 24 джерела.

Метою роботи є розробка текстового редактора на основі Windows Forms і .NET 8, який поєднує форматування тексту, синтаксичну підсвітку для мов програмування (C++, C#, Java, Python, JavaScript) і підтримку української мови через UTF-8. Проведено аналіз аналогів (Notepad++, Visual Studio Code, Sublime Text, Microsoft Word), що виявив потребу в універсальному редакторі. Обґрунтовано вибір .NET 8 і Windows Forms через їх продуктивність і простоту створення інтерфейсу.

Розроблено алгоритми синтаксичної підсвітки (пошук підрядків, регулярні вирази), пошуку, заміни та форматування, оптимізовані асинхронною обробкою та дебонсингом. Програма базується на класі TextEditorForm, що інкапсулює графічний інтерфейс і логіку. Інтерфейс включає панель інструментів, текстову область і форму пошуку. Реалізація підтримує UTF-8, обробку великих файлів і стабільність.

Описано системні вимоги, інтерфейс і інструкцію користування. Редактор є зручним, конкурентоспроможним і придатним для навчальних та професійних цілей. Перспективи розвитку включають автодоповнення та кросплатформність. Робота демонструє ефективність .NET 8 і Windows Forms для створення настільних додатків.

Ключові слова: текстовий редактор, форматування, синтаксична підсвітка, Windows Forms, .NET 8, UTF-8.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

.NET — платформа розробки від Microsoft, яка використовується для створення додатків, зокрема текстового редактора, у середовищі .NET 8.

Windows Forms — технологія для створення графічного інтерфейсу настільних додатків у середовищі .NET.

UTF-8 — кодування символів Unicode, що забезпечує коректне відображення українських та інших символів у текстових файлах і інтерфейсі програми.

RTF — формат файлу (Rich Text Format), який підтримує форматування тексту, включаючи шрифти, стилі та вирівнювання.

RichTextBox — компонент Windows Forms для введення, редагування та відображення тексту з підтримкою форматування та синтаксичної підсвітки.

ToolStrip — компонент Windows Forms, що реалізує панель інструментів із кнопками та іншими елементами керування.

ToolStripButton — кнопка на панелі інструментів (ToolStrip), яка відповідає за виконання певної дії (наприклад, відкриття файлу, форматування).

ToolStripComboBox — випадаючий список на панелі інструментів, що використовується для вибору мови програмування.

ToolStripProgressBar — індикатор прогресу на панелі інструментів, який відображає стан обробки великих файлів.

ContextMenuStrip — компонент Windows Forms, що реалізує контекстне меню для текстової області.

Regex — бібліотека регулярних виразів у .NET, яка використовується для обробки шаблонів тексту під час синтаксичної підсвітки.

C++, C#, Java, Python, JavaScript — мови програмування, для яких підтримується синтаксична підсвітка в текстовому редакторі.

O(n) — позначення часової складності алгоритму в нотації "О велике", де (n) — розмір вхідних даних (наприклад, довжина тексту).

BOM — позначка порядку байтів (Byte Order Mark) у кодуванні UTF-8, що використовується для коректного збереження файлів.

					ДР.122.042.026.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному світі інформаційних технологій текстовий редактор є одним із найпоширеніших інструментів, що використовується для створення, редагування та форматування текстових документів. Від простих текстових файлів до складних документів із форматуванням, синтаксичною підсвіткою та підтримкою різних мов програмування, текстові редактори відіграють ключову роль у професійній діяльності розробників програмного забезпечення, студентів, науковців та офісних працівників. Розвиток технологій, зокрема платформи .NET та Windows Forms, відкриває нові можливості для створення ефективних, зручних та функціональних текстових редакторів, які відповідають сучасним вимогам користувачів.

Метою даної дипломної роботи є розробка текстового редактора з підтримкою форматування на основі технології Windows Forms у середовищі .NET 8. Запропонований редактор забезпечує базові функції роботи з текстом, такі як створення, збереження та відкриття файлів у різних форматах (зокрема .txt, .rtf та файли мов програмування), а також розширені можливості, включаючи форматування тексту (вибір шрифту, кольору, стилю), вирівнювання, пошук і заміну тексту, а також синтаксичну підсвітку для популярних мов програмування (C++, C#, Java, Python, JavaScript). Особливу увагу приділено підтримці української мови, включаючи коректне відображення символів UTF-8, а також оптимізації продуктивності за допомогою асинхронної обробки великих файлів і дебонсингу для підсвітки синтаксису.

Актуальність роботи зумовлена зростанням попиту на універсальні текстові редактори, які поєднують простоту використання з потужними функціями форматування та аналізу коду. Розробка такого редактора на платформі .NET 8 дозволяє не лише реалізувати сучасний інтерфейс і стабільну роботу, але й продемонструвати можливості Windows Forms для створення настільних додатків. Використання .NET 8 забезпечує високу

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

продуктивність, підтримку сучасних стандартів програмування та гнучкість у реалізації складних функцій, таких як асинхронна обробка і робота з Unicode.

У процесі розробки використано об'єктно-орієнтований підхід, що дозволило створити модульну структуру програми з чітким розподілом функціональних компонентів. Основними завданнями роботи є:

- Реалізація інтуїтивно зрозумілого графічного інтерфейсу з використанням Windows Forms.
- Забезпечення підтримки форматування тексту та синтаксичної підсвітки.
- Оптимізація продуктивності обробки великих текстових файлів.
- Впровадження підтримки UTF-8 для коректної роботи з українськими символами.
- Інтеграція асинхронних механізмів для підвищення швидкодії.

Розроблений текстовий редактор може бути використаний як у навчальних цілях, так і в професійній діяльності, зокрема для написання та аналізу коду, створення документів із форматуванням, а також як основа для подальшого розвитку функціоналу, наприклад, додавання автодоповнення коду чи інтеграції з хмарними сервісами.

Дана дипломна робота має як теоретичне, так і практичне значення, демонструючи можливості сучасних технологій .NET 8 і Windows Forms для створення функціональних настільних додатків, а також вирішуючи актуальні задачі у сфері розробки текстових редакторів із підтримкою форматування та синтаксичної підсвітки.

					ДР.122.042.026.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ

1.1. Постановка основної задачі розробки

Розробка текстового редактора з підтримкою форматування є актуальним завданням у контексті сучасних інформаційних технологій, оскільки такі програми широко застосовуються в освіті, професійній діяльності розробників програмного забезпечення та створенні документів різного призначення. Основна задача полягає в створенні настільного додатка на основі технології Windows Forms у середовищі .NET 8, який забезпечує базові функції роботи з текстом, розширені можливості форматування, синтаксичну підсвітку для популярних мов програмування та підтримку української мови через коректне відображення символів UTF-8.

Текстовий редактор повинен надавати користувачу інтуїтивно зрозумілий графічний інтерфейс для створення, редагування, збереження та відкриття файлів у різних форматах, таких як .txt, .rtf, а також файли кодів мов програмування, зокрема C++, C#, Java, Python і JavaScript. Важливою вимогою є реалізація форматування тексту, що включає вибір шрифту, стилю (жирний, курсив, підкреслений), кольору, а також вирівнювання тексту за лівим, правим або центральним краєм. Окрім цього, редактор має підтримувати пошук і заміну тексту, що полегшує роботу з великими документами.

Ключовою особливістю програми є синтаксична підсвітка, яка дозволяє виділяти ключові слова, рядки, числа та коментарі у коді відповідними кольорами та стилями, підвищуючи зручність роботи з програмним кодом. Для забезпечення високої продуктивності при обробці великих файлів необхідно впровадити асинхронну обробку та механізм дебонсингу, що мінімізує затримки під час підсвітки синтаксису. Підтримка UTF-8 гарантує коректне відображення українських символів, що є важливим для локалізації програми.

Для реалізації поставлених задач обрано платформу .NET 8, яка забезпечує сучасні інструменти розробки, високу продуктивність і стабільність роботи настільних додатків. Технологія Windows Forms обрана

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

завдяки її простоті у створенні графічних інтерфейсів, широкій підтримці компонентів для роботи з текстом і форматуванням, а також сумісності з .NET 8. Програма має бути модульною, з чітким розподілом функціональних компонентів, що полегшить її підтримку та подальше розширення.

Основна задача розробки полягає в створенні текстового редактора, який поєднує простоту використання, розширені функції форматування, синтаксичну підсвітку, підтримку української мови та високу продуктивність. Реалізація такого редактора на базі Windows Forms і .NET 8 дозволить створити універсальний інструмент, придатний як для навчальних, так і для професійних цілей, з можливістю подальшого вдосконалення.

1.2. Огляд та порівняння аналогічних систем.

Розробка текстового редактора з підтримкою форматування та синтаксичної підсвітки вимагає аналізу існуючих аналогічних систем, щоб визначити їхні сильні сторони, недоліки та сформулювати вимоги до власного рішення. У цьому розділі розглядаються популярні текстові редактори, які використовуються для роботи з текстом і кодом, а також проводиться їх порівняння за ключовими характеристиками, такими як функціональність, інтерфейс, продуктивність і підтримка локалізації.

Одним із найвідоміших текстових редакторів є **Notepad++**, відкрите програмне забезпечення, призначене для редагування текстових файлів і коду. Notepad++ підтримує синтаксичну підсвітку для десятків мов програмування, включаючи C++, C#, Java, Python і JavaScript. Програма дозволяє працювати з кількома вкладками, має функції пошуку та заміни, підтримує плагіни для розширення функціоналу, наприклад, автодоповнення або порівняння файлів. Інтерфейс Notepad++ простий, але дещо застарілий, що може впливати на зручність для нових користувачів. Програма оптимізована для роботи з великими файлами, однак не підтримує розширене форматування тексту, таке як вибір шрифтів чи вирівнювання, що обмежує її використання для створення

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

документів. Локалізація, включаючи українську мову, реалізована коректно завдяки підтримці UTF-8.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <meta name="description" content="This page is experimental">
6 <meta name="keyword" content="html, css, learning, experimental">
7 <meta name="author" content="Sharoonas">
8 <meta name="viewport" content="width=device-width, initial-scale=1.0">
9 <title>This page is experimental and is used only for learning purpose</title>
10 <style>
11 body{background-image:url(https://thumbs.gfycat.com/DistinctScrawnyCicada.webp)}
12 h1{ color:white; font-size:100px;text-transform:uppercase;}
13 s{color:red; font-size:64px;text-transform:lowercase;}
14 </style>
15 </head>
16 <body style=" background-color: black">
17 <h1>Eikit <s> nah </s> su dievu</h1>
18 </body>
19 </html>

```

length : 770 lines : 19 Ln : 15 Col : 1 Sel : 7 | 1 Windows (CR LF) UTF-8 INS

Рисунок 1.2.1 – Notepad++

Ще одним популярним рішенням є **Visual Studio Code (VS Code)**, кросплатформовий редактор від Microsoft, який здобув широку популярність серед розробників. VS Code підтримує синтаксичну підсвітку, автодоповнення, інтеграцію з системами контролю версій, такими як Git, і розширення через маркетплейс. Інтерфейс сучасний і налаштовуваний, що робить його зручним для різних категорій користувачів. Програма ефективно обробляє великі файли завдяки асинхронній архітектурі, але потребує значних ресурсів при використанні багатьох розширень. VS Code підтримує форматування коду, але не має вбудованих інструментів для роботи з текстом у стилі документів, таких як вибір стилів чи вирівнювання. Українська локалізація доступна через розширення, однак не завжди повна.

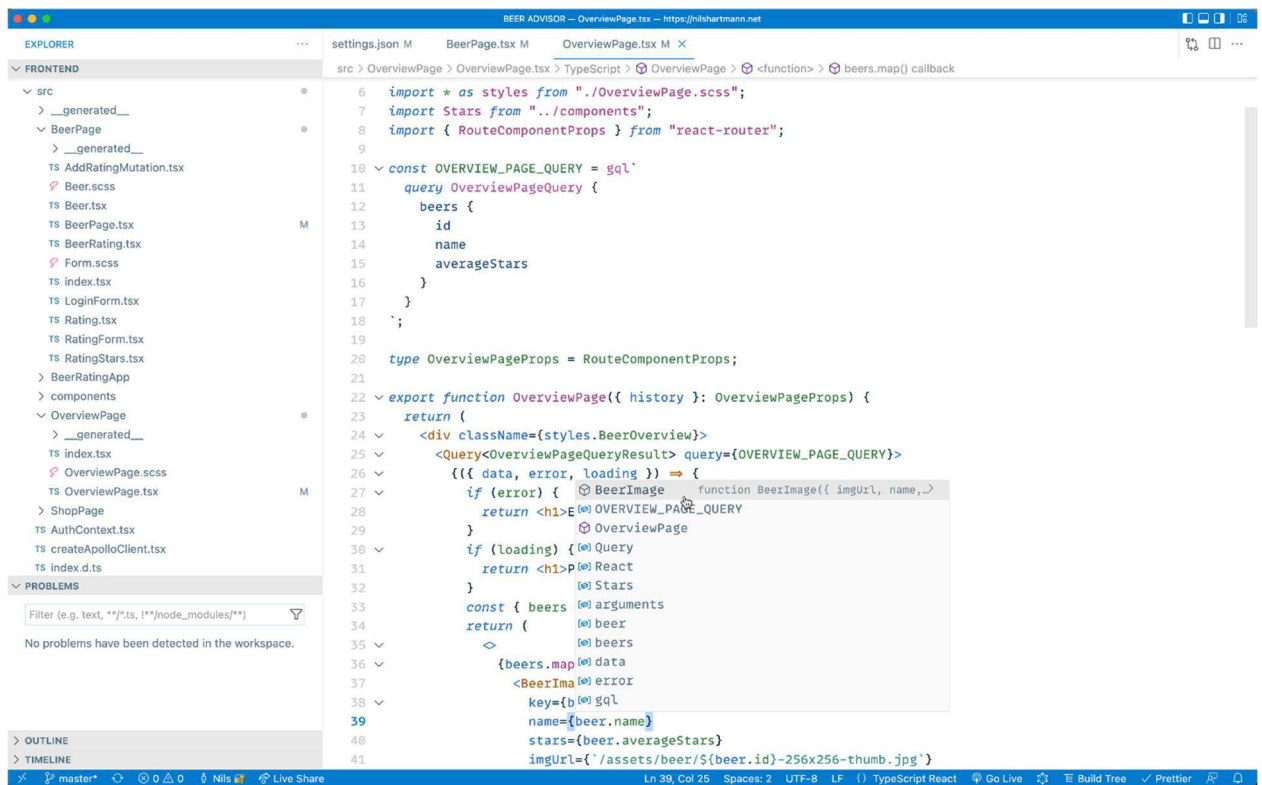


Рисунок 1.2.2 – Visual Studio Code

Sublime Text — це комерційний текстовий редактор із мінімалістичним дизайном і високою швидкістю. Він підтримує синтаксичну підсвітку, множинне редагування, пошук і заміну, а також розширення через плагіни. Sublime Text оптимізований для роботи з великими файлами та має гнучкий інтерфейс, який можна налаштувати під потреби користувача. Однак програма не пропонує інструментів для форматування тексту, таких як зміна шрифтів чи кольорів у стилі текстових процесорів, і її функціонал для документів обмежений. Підтримка UTF-8 забезпечує коректне відображення українських символів, але локалізація інтерфейсу потребує додаткових налаштувань.

```

75     def center_view(self, view):
76         sel = view.sel()
77         region = sel[0] if len(sel) == 1 else None
78         if region != None:
79             global offset
80             offset = sublime.load_settings('Typewriter.sublime-settings').get(
81                 'typewriter_mode_scrolling_offset', 0.0) * view.line_height()
82             if offset != 0:
83                 if offset > (view.viewport_extent()[1] / 2):
84                     print(
85                         "Typewriter Error: The cursor is outside the viewport! U
86                     else:
87                         region = sublime.Region(
88                             self.offset_point(view, region.a), self.offset_point(vie
89                             view.show_at_center(region)
90
91     def offset_point(self, view, point):
92         vector = list(view.text_to_layout(point))
93         vector[1] = vector[1] + offset
94         return view.layout_to_text(tuple(vector))
95

```

172 Words, Unix, Line 77, Column 50 Spaces: 4 Python

Рисунок 1.2.3 – Sublime Text

Microsoft Word, хоча і є текстовим процесором, також розглядається як аналог через підтримку розширеного форматування. Word дозволяє змінювати шрифти, стилі, кольори, вирівнювання, а також працювати з таблицями, зображеннями та іншими елементами документів. Інтерфейс програми зручний для створення документів, але не призначений для редагування коду, оскільки синтаксична підсвітка відсутня. Word ефективно працює з UTF-8 і має повну українську локалізацію, однак його продуктивність знижується при обробці великих текстових файлів, а функціонал для програмування відсутній.

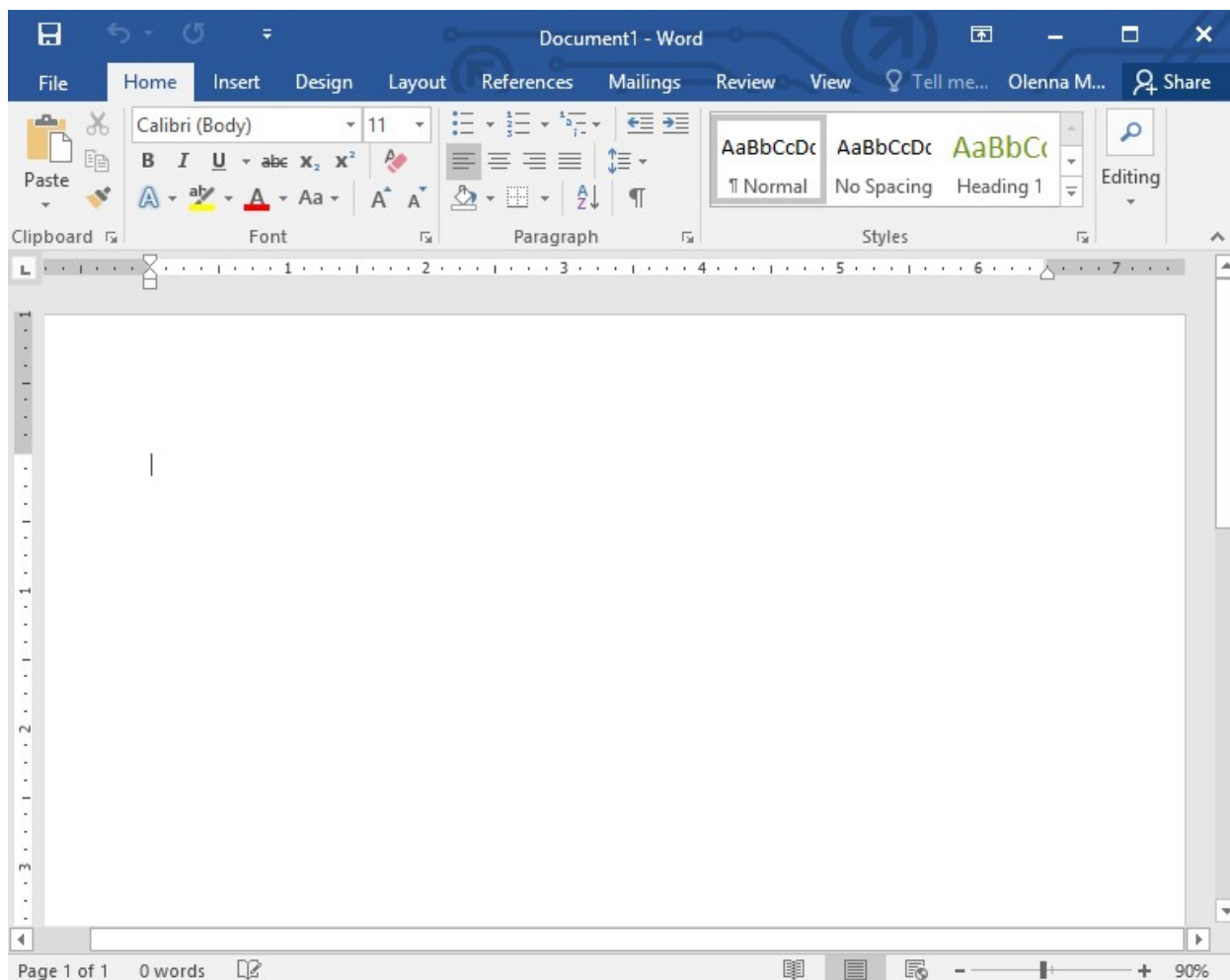


Рисунок 1.2.4 – Microsoft Word

Для порівняння аналогів використано такі критерії: підтримка форматування тексту, синтаксична підсвітка, продуктивність при роботі з великими файлами, зручність інтерфейсу, підтримка UTF-8 і української локалізації, а також кросплатформність. Notepad++ вирізняється простотою і швидкодією, але не підтримує форматування тексту. VS Code пропонує потужний функціонал для програмування, однак потребує додаткових ресурсів і налаштувань для локалізації. Sublime Text забезпечує високу продуктивність і гнучкість, але обмежений у форматуванні документів. Microsoft Word ідеально підходить для роботи з документами, але не придатний для редагування коду. Жоден із аналогів не поєднує в повній мірі розширене форматування тексту, синтаксичну підсвітку та оптимізацію для великих файлів у рамках настільного додатка з повною українською локалізацією.

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Розроблюваний текстовий редактор на базі Windows Forms і .NET 8 має на меті усунути ці недоліки, поєднуючи можливості форматування тексту, подібні до Microsoft Word, із синтаксичною підсвіткою, як у Notepad++ чи VS Code, у легкому та швидкому настільному додатку. Програма фокусується на підтримці UTF-8 для коректного відображення українських символів, асинхронній обробці для роботи з великими файлами та інтуїтивному інтерфейсі, оптимізованому для користувачів із різним рівнем підготовки. Windows Forms забезпечує простоту реалізації графічного інтерфейсу, а .NET 8 гарантує продуктивність і сучасні можливості, такі як асинхронне програмування, що робить розробку конкурентоспроможною порівняно з аналогами.

Аналіз аналогічних систем показав, що існує потреба в текстовому редакторі, який би поєднував розширене форматування, синтаксичну підсвітку, високу продуктивність і повну підтримку української мови. Розробка такого редактора на основі Windows Forms і .NET 8 є виправданою, оскільки дозволяє створити універсальний інструмент, який відповідає сучасним вимогам користувачів і заповнює прогалини, виявлені в існуючих рішеннях.

1.3. Вибір та обґрунтування технологій розробки

Вибір технологій для розробки текстового редактора з підтримкою форматування та синтаксичної підсвітки є ключовим етапом, що визначає ефективність, продуктивність і зручність кінцевого продукту. Для реалізації поставлених задач необхідно обрати платформу, яка забезпечує створення стабільного настільного додатка з інтуїтивним графічним інтерфейсом, підтримкою розширеного форматування тексту, асинхронною обробкою великих файлів і коректною роботою з українськими символами. Після аналізу доступних технологій обрано платформу .NET 8 у поєднанні з технологією Windows Forms, що пояснюється їхніми можливостями та відповідністю вимогам проєкту.

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Платформа .NET 8 була обрана як основа для розробки завдяки її сучасній архітектурі, високій продуктивності та широкому набору інструментів для створення настільних додатків. .NET 8 є останньою версією платформи від Microsoft, яка забезпечує покращену швидкодію порівняно з попередніми версіями, оптимізовану роботу з пам'яттю та підтримку асинхронного програмування через ключові слова `async/await`. Ці можливості є критично важливими для реалізації синтаксичної підсвітки великих файлів, де обробка тексту виконується у фоновому режимі, не блокуючи основний інтерфейс користувача. Крім того, .NET 8 підтримує кросплатформність, що відкриває перспективи для подальшого портування редактора на інші операційні системи, хоча на даному етапі проєкт фокусується на Windows. Вбудована підтримка кодування UTF-8 у .NET 8 гарантує коректне відображення українських символів, що є однією з основних вимог до локалізації програми.

Технологія Windows Forms обрана для створення графічного інтерфейсу завдяки своїй простоті, зрілості та широкій підтримці компонентів для роботи з текстом. Windows Forms є частиною екосистеми .NET і забезпечує швидке створення настільних додатків із використанням готових елементів керування, таких як `RichTextBox`, `ToolStrip`, `ContextMenuStrip` та діалогові вікна для вибору шрифтів і кольорів. Компонент `RichTextBox` ідеально підходить для реалізації форматування тексту (вибір шрифту, стилю, кольору, вирівнювання) та синтаксичної підсвітки, оскільки дозволяє маніпулювати текстом на рівні окремих символів і ділянок. Windows Forms також пропонує простий механізм прив'язки подій, що полегшує обробку дій користувача, таких як натискання кнопок чи зміна тексту. Порівняно з альтернативами, такими як WPF чи Avalonia, Windows Forms має нижчий поріг входження, що важливо для швидкої реалізації проєкту, і менші вимоги до ресурсів, що сприяє продуктивності на слабших комп'ютерах.

Альтернативні технології, такі як WPF (Windows Presentation Foundation), розглядалися, але не були обрані через їхню складність і вищі

					ДР.122.042.026.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

вимоги до ресурсів. WPF забезпечує сучасніший інтерфейс і підтримку векторної графіки, але потребує більше часу на освоєння та налаштування, що не виправдано для текстового редактора з класичним інтерфейсом. Avalonia, як кросплатформовий фреймворк, могла б бути корисною для майбутнього портування, але на момент розробки її екосистема менш розвинена порівняно з Windows Forms, а підтримка компонентів для роботи з текстом обмежена. Qt і JavaFX також розглядалися, але їх інтеграція з екосистемою Windows і .NET є менш природною, а розробка вимагала б додаткових зусиль для локалізації та оптимізації.

Для реалізації синтаксичної підсвітки використано об'єктно-орієнтований підхід у поєднанні з регулярними виразами для аналізу тексту. Регулярні вирази дозволяють ефективно виділяти ключові слова, рядки, числа та коментарі, що є стандартною практикою в текстових редакторах. Асинхронне програмування, підтримуване .NET 8, використано для обробки великих файлів, щоб уникнути блокування інтерфейсу. Механізм дебонсінгу, реалізований через таймер, забезпечує оптимізацію підсвітки при частій зміні тексту, зменшуючи кількість викликів обробки.

Локалізація програми, зокрема підтримка української мови, реалізована завдяки вбудованій підтримці UTF-8 у .NET 8 і Windows Forms. Для роботи з файлами використано стандартні класи .NET, такі як File і Encoding, що дозволяють коректно обробляти текстові файли у форматах .txt, .rtf та інших, з резервною підтримкою кодування Windows-1251 для старих файлів. Використання Windows Forms також полегшує інтеграцію діалогових вікон, таких як OpenFileDialog і SaveFileDialog, що підвищує зручність роботи з файлами.

Обґрунтування вибору .NET 8 і Windows Forms базується на їхній здатності забезпечити швидку розробку стабільного настільного додатка з усіма необхідними функціями: форматуванням тексту, синтаксичною підсвіткою, підтримкою UTF-8 і високою продуктивністю. .NET 8 надає сучасні інструменти для асинхронного програмування та оптимізації, а

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Windows Forms спрощує створення графічного інтерфейсу з готовими компонентами, що ідеально відповідає цілям проєкту. Цей вибір дозволяє створити текстовий редактор, який є конкурентоспроможним порівняно з аналогами, зручним для користувачів і придатним для використання в навчальних і професійних цілях, а також має потенціал для подальшого розширення функціоналу.

Висновки до розділу 1

Проведений огляд і аналіз технологій розробки дозволив сформулювати чітке уявлення про задачі створення текстового редактора з підтримкою форматування та синтаксичної підсвітки, а також визначити оптимальний набір інструментів для їх реалізації. Основна задача полягає в розробці настільного додатка, який поєднує функції форматування тексту, синтаксичну підсвітку для популярних мов програмування, підтримку української мови через UTF-8 і високу продуктивність при роботі з великими файлами.

Аналіз аналогічних систем, таких як Notepad++, Visual Studio Code, Sublime Text і Microsoft Word, показав, що жоден із них не поєднує в повній мірі розширене форматування тексту, синтаксичну підсвітку та оптимізацію для великих файлів у рамках легкого настільного додатка з повною українською локалізацією. Цей аналіз підкреслив необхідність створення універсального текстового редактора, який усуває виявлені недоліки та відповідає потребам користувачів.

Для реалізації проєкту обрано платформу .NET 8 і технологію Windows Forms. .NET 8 забезпечує високу продуктивність, підтримку асинхронного програмування та коректну роботу з UTF-8, що є критично важливим для локалізації та обробки великих файлів. Windows Forms вибрано завдяки простоті створення графічного інтерфейсу, наявності компонентів для роботи з текстом, таких як RichTextBox, і низьким вимогам до ресурсів. Альтернативні технології, такі як WPF чи Avalonia, відхилено через їхню

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

складність і меншу відповідність завданням проєкту. Використання регулярних виразів, асинхронної обробки та дебонсінгу додатково оптимізує функціонал синтаксичної підсвітки.

Обрані технології .NET 8 і Windows Forms є обґрунтованим вибором для створення текстового редактора, який відповідає сучасним вимогам, забезпечує зручність, продуктивність і можливості для подальшого розвитку. Ці висновки створюють основу для практичної реалізації проєкту, описаної в наступних розділах.

					ДР.122.042.026.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА

2.1. Вибір алгоритмів та їх ефективність

Розробка текстового редактора з підтримкою форматування та синтаксичної підсвітки вимагає ретельного вибору алгоритмів, які забезпечують високу продуктивність, коректність роботи та зручність для користувача. Основними задачами, що потребують алгоритмічного вирішення, є синтаксична підсвітка тексту, обробка великих файлів, пошук і заміна тексту, а також підтримка форматування. У цьому підрозділі розглядаються обрані алгоритми, їх обґрунтування та аналіз ефективності з точки зору часової та просторової складності.

Синтаксична підсвітка

Синтаксична підсвітка є ключовою функцією редактора, яка полягає у виділенні ключових слів, рядків, чисел і коментарів у коді різними кольорами та стилями. Для реалізації цього функціоналу використано комбінацію алгоритмів пошуку підрядків і регулярних виразів.

1. Пошук ключових слів.

Для виділення ключових слів (наприклад, `int`, `class`, `def`) застосовується алгоритм прямого пошуку підрядків із перевіркою меж слова. Кожне ключове слово зі списку, завантаженого з файлу для відповідної мови програмування, шукається в тексті за допомогою методу `IndexOf` із врахуванням регістру. Щоб уникнути помилкового виділення підрядків усередині слів (наприклад, `print` у `printf`), перевіряються межі слова: символи перед і після ключового слова не повинні бути буквами чи цифрами. Часова складність цього алгоритму залежить від довжини тексту (n), кількості ключових слів (k) і середньої довжини ключового слова (m), що дає приблизну оцінку ($O(n \cdot k \cdot m)$) у найгіршому випадку. Для зменшення накладних витрат список ключових слів зберігається в пам'яті як масив, що забезпечує швидкий доступ із часовою складністю ($O(1)$).

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

2. Регулярні вирази для рядків, чисел і коментарів.

Для виділення рядкових літералів, числових значень і коментарів використано регулярні вирази, які дозволяють ефективно знаходити складні шаблони в тексті. Наприклад, рядки обробляються за допомогою шаблону `@"(?:\\.[^"]*)"|'(?:\\.[^']*')*"`, який розпізнає як одинарні, так і подвійні лапки з урахуванням екранування. Числові літерали обробляються шаблоном `@"\b\d+(\.\d+)?([eE][+-]?\d+)?\b"`, що враховує цілі та дробові числа, а також експоненціальну форму. Для коментарів використано окремі шаблони залежно від мови: `@"#.*?(?=\n|$)"` для Python і `@"//.*?(?=\n|$)"` та `@"/.*?*/"`, для C-подібних мов. Часова складність обробки регулярних виразів залежить від реалізації бібліотеки .NET, але в середньому становить ($O(n)$) для лінійного пошуку, де (n) — довжина тексту. Використання регулярних виразів забезпечує гнучкість і дозволяє легко додавати нові шаблони для інших мов програмування.

3. Оптимізація за допомогою дебонсингу та асинхронності.

Для підвищення продуктивності при частій зміні тексту (наприклад, під час набору) застосовано техніку дебонсингу, яка обмежує частоту викликів підсвітки. Алгоритм працює через таймер із затримкою 300 мс: кожен виклик підсвітки перезавантажує таймер, і обробка виконується лише після паузи у введенні тексту. Це знижує кількість викликів із ($O(n)$) до одного на групу змін, значно зменшуючи навантаження. Асинхронна обробка, реалізована через Task.Run, дозволяє виконувати підсвітку у фоновому потоці, не блокуючи інтерфейс користувача. Часова складність асинхронної обробки залишається ($O(n \cdot k \cdot m)$) для повного тексту, але користувач не відчуває затримок.

4. Обробка великих файлів.

Для великих файлів використано часткову підсвітку, яка обмежується поточним рядком і сусідніми рядками. Це знижує часову складність до ($O(1 \cdot k \cdot m)$), де (1) — довжина одного рядка, що значно менше за (n).

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Просторова складність залежить від розміру тексту та тимчасового RichTextBox, який зберігає RTF-представлення, і становить ($O(n)$).

Пошук і заміна тексту

Функція пошуку та заміни тексту реалізована двома алгоритмами:

1. Пошук підрядка.

Для пошуку тексту використано метод IndexOf із підтримкою ігнорування регістру, що має часову складність ($O(n + m)$), де (n) — довжина тексту, а (m) — довжина шуканого підрядка. Пошук виконується послідовно від поточної позиції курсора, що дозволяє користувачу знаходити наступне входження. Просторова складність становить ($O(1)$), оскільки додаткової пам'яті не потрібно.

2. Заміна тексту.

Для заміни використано регулярні вирази через метод Regex.Replace, що дозволяє замінити всі входження або виконати поодинокі заміни. Часова складність заміни становить ($O(n)$) для лінійного проходу тексту, а просторова — ($O(n)$) через створення нового рядка з результатом. Регулярні вирази забезпечують гнучкість, дозволяючи враховувати регістр або використовувати шаблони.

Форматування тексту

Форматування тексту (вибір шрифту, стилю, кольору, вирівнювання) реалізовано через вбудовані можливості компонента RichTextBox у Windows Forms. Алгоритми форматування мають часову складність ($O(1)$), оскільки змінюють лише властивості виділеного тексту, такі як SelectionFont, SelectionColor чи SelectionAlignment. Просторова складність також становить ($O(1)$), оскільки не потребує додаткової пам'яті. Для збереження форматування у файлах .rtf використано метод SaveFile, який автоматично обробляє RTF-структуру, забезпечуючи сумісність із іншими редакторами.

Робота з файлами

Для читання та запису файлів використано класи .NET File і Encoding. Читання текстових файлів із підтримкою UTF-8 і резервним кодуванням

					ДР.122.042.026.ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Windows-1251 має часову складність ($O(n)$), де (n) — розмір файлу. Запис файлів також виконується за ($O(n)$). Використання UTF8Encoding із BOM (Byte Order Mark) забезпечує коректне збереження українських символів. Просторова складність залежить від розміру файлу і становить ($O(n)$).

Оцінка ефективності

Обрані алгоритми забезпечують баланс між продуктивністю та функціональністю. Синтаксична підсвітка, яка є найбільш ресурсоемною операцією, оптимізована за допомогою дебонсингу, асинхронності та часткової обробки, що робить її ефективною навіть для великих файлів. Пошук і заміна тексту працюють швидко завдяки лінійним алгоритмам, а форматування та робота з файлами мають мінімальну обчислювальну складність. Просторова складність усіх алгоритмів пропорційна розміру тексту, що є прийнятним для настільного додатка.

Обрані алгоритми відповідають вимогам щодо продуктивності, дозволяють ефективно реалізовувати синтаксичну підсвітку, форматування, пошук і роботу з файлами, забезпечуючи швидкодію та зручність для користувача. Використання асинхронного програмування та дебонсингу додатково підвищує ефективність, роблячи редактор придатним для роботи з текстами різного обсягу та складності.

2.2. Спроектовані класи програми

Розробка текстового редактора з підтримкою форматування та синтаксичної підсвітки на основі Windows Forms і .NET 8 передбачає створення модульної структури програми з чітким розподілом функціональних компонентів. Для забезпечення зрозумілості, розширюваності та легкості підтримки використано об'єктно-орієнтований підхід. У цьому підрозділі описуються спроектовані класи програми, їх призначення, основні методи та взаємодія, що дозволяє реалізувати всі

					ДР.122.042.026.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідні функції редактора, включаючи форматування тексту, синтаксичну підсвітку, роботу з файлами та підтримку української мови через UTF-8.

Програма складається з одного основного класу, який відповідає за весь функціонал редактора, та допоміжних структур, що забезпечують обробку даних. Основна логіка зосереджена в класі форми, створеному за допомогою Windows Forms, що дозволяє ефективно використовувати вбудовані компоненти для графічного інтерфейсу та обробки тексту. Нижче наведено детальний опис спроектованих класів і їх ролі в програмі.

Клас `TextEditorForm`

Клас `TextEditorForm`, успадкований від `System.Windows.Forms.Form`, є центральним елементом програми, відповідаючи за створення графічного інтерфейсу, обробку подій користувача та реалізацію всіх функціональних можливостей редактора. Цей клас інкапсулює логіку роботи з текстом, форматуванням, синтаксичною підсвіткою, роботою з файлами та локалізацією.

Поля класу:

- `languageKeywords`: `Dictionary<string, string[]>` — словник, що зберігає списки ключових слів для кожної підтримуваної мови програмування (C++, C#, Java, Python, JavaScript). Ключ — назва мови, значення — масив ключових слів.
- `currentLanguage`: `string` — поточна мова програмування, обрана користувачем для синтаксичної підсвітки.
- `extensionToLanguage`: `Dictionary<string, string>` — словник для зіставлення розширень файлів (наприклад, `.cpp`, `.cs`) із відповідними мовами програмування.
- `isHighlighting`: `bool` — прапорець, що запобігає одночасному виконанню кількох операцій підсвітки.
- `debounceTimer`: `System.Windows.Forms.Timer` — таймер для реалізації дебонсингу при підсвітці синтаксису, із затримкою 300 мс.

- `toolStrip`: `ToolStrip` — панель інструментів із кнопками для форматування, роботи з файлами та вибору мови.
- `richTextBox`: `RichTextBox` — основний компонент для введення, редагування та відображення тексту з підтримкою форматування та підсвітки.
- `languageCombo`: `ToolStripComboBox` — випадаючий список для вибору мови програмування.
- `progressBar`: `ToolStripProgressBar` — індикатор прогресу для відображення стану обробки великих файлів.

Основні методи:

- `TextEditorForm()` — конструктор, який ініціалізує словник `extensionToLanguage` і налаштовує таймер дебонсингу.
- `TextEditorForm_Load(object, EventArgs)` — обробник події завантаження форми, який ініціалізує графічний інтерфейс, завантажує ключові слова, налаштовує контекстне меню та таймер дебонсингу.
- `LoadKeywords()` — завантажує ключові слова з файлів у папці `Keywords`, створюючи словник `languageKeywords`. Метод підтримує UTF-8 для коректного читання українських символів.
- `HighlightSyntaxAsync(bool)` — асинхронний метод для підсвітки синтаксису, який очищає `RichTextBox`, виконує обробку у фоновому потоці і відновлює текст із підсвіткою. Використовує тимчасовий `RichTextBox` для уникнення мерехтіння.
- `HighlightFullText(RichTextBox, string)` — виконує підсвітку всього тексту, обробляючи кожен рядок і оновлюючи прогрес-бар.
- `HighlightLine(RichTextBox, string, int)` — підсвічує окремий рядок, оптимізуючи обробку при редагуванні.
- `HighlightTextRange(RichTextBox, string, int, int)` — застосовує підсвітку до діапазону тексту, використовуючи регулярні вирази для рядків, чисел і коментарів.

- `Bold_Click(object, EventArgs)`, `Italic_Click(object, EventArgs)`, `Underline_Click(object, EventArgs)` — обробники для зміни стилів тексту (жирний, курсив, підкреслений).
- `AlignLeft_Click(object, EventArgs)`, `AlignCenter_Click(object, EventArgs)`, `AlignRight_Click(object, EventArgs)` — методи для вирівнювання тексту.
- `Font_Click(object, EventArgs)`, `Color_Click(object, EventArgs)` — відкривають діалогові вікна для вибору шрифту та кольору.
- `Save_Click(object, EventArgs)`, `Open_Click(object, EventArgs)` — обробляють збереження та відкриття файлів у форматах `.txt`, `.rtf` та інших, із підтримкою UTF-8 і резервним кодуванням Windows-1251.
- `Find_Click(object, EventArgs)` — відкриває форму для пошуку та заміни тексту, використовуючи `IndexOf` і регулярні вирази.
- `LanguageCombo_SelectedIndexChanged(object, EventArgs)` — оновлює мову підсвітки при зміні вибору в `languageCombo`.
- `RichTextBox_TextChanged(object, EventArgs)` — запускає таймер дебонсингу при зміні тексту.

Клас `TextEditorForm` є самодостатнім, оскільки інкапсулює весь функціонал редактора. Він взаємодіє з компонентами Windows Forms (`RichTextBox`, `ToolStrip`, `ToolStripComboBox`, `ToolStripProgressBar`) через обробники подій, що забезпечують реакцію на дії користувача. Для роботи з файлами використовуються класи .NET (`File`, `Encoding`), а для підсвітки — регулярні вирази (`Regex`). Зовнішні залежності обмежені стандартною бібліотекою .NET 8 і Windows Forms, що спрощує розгортання програми.

Хоча програма не використовує окремих допоміжних класів, у ній застосовуються структури даних для організації роботи:

Словники:

- `languageKeywords` (тип `Dictionary<string, string[]>`) для зберігання ключових слів.
- `extensionToLanguage` (тип `Dictionary<string, string>`) для зіставлення розширень файлів із мовами.

					ДР.122.042.026.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Тимчасовий RichTextBox використовується в методі HighlightSyntaxAsync для обробки підсвітки у фоновому потоці, що дозволяє очищати основний RichTextBox під час обробки та уникати мерехтіння.

Структура програми спроектована з урахуванням принципів модульності та єдиної відповідальності. Клас TextEditorForm об'єднує всі функції, але чітко розподіляє їх за методами, кожен із яких відповідає за окрему задачу (форматування, підсвітка, робота з файлами). Використання Windows Forms дозволяє делегувати управління графічним інтерфейсом вбудованим компонентам, що зменшує складність коду. Відсутність окремих класів для логіки підсвітки чи роботи з файлами обґрунтована компактністю проєкту: усі операції тісно пов'язані з інтерфейсом і компонентом RichTextBox, тому їх винесення в окремі класи не підвищило б зрозумілості коду.

Для підвищення розширюваності передбачено можливість додавання нових мов програмування шляхом розміщення файлів ключових слів у папці Keywords. Підтримка UTF-8 забезпечується використанням стандартних класів .NET, що гарантує коректну роботу з українськими символами. Асинхронна обробка та дебонсинг, реалізовані в HighlightSyntaxAsync, підвищують продуктивність, особливо для великих файлів.

Оскільки програма містить лише один основний клас, діаграма класів спрощена. Клас TextEditorForm успадковується від Form і містить поля для компонентів Windows Forms та допоміжних структур даних. Взаємодія з користувачем відбувається через обробники подій, а з файлами — через класи .NET.

Спроектована структура програми базується на єдиному класі TextEditorForm, який ефективно інтегрує всі функції редактора завдяки компонентам Windows Forms і можливостям .NET 8. Використання вбудованих компонентів, таких як RichTextBox і ToolStrip, спрощує реалізацію графічного інтерфейсу та обробки тексту, тоді як асинхронна обробка та дебонсинг забезпечують продуктивність. Структура є модульною

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

та розширюваною, дозволяючи додавати нові мови програмування чи функції без значних змін у коді. Такий підхід відповідає вимогам проєкту, забезпечуючи зручність, стабільність і підтримку української мови.

2.3. Програмна реалізація

Програмна реалізація текстового редактора з підтримкою форматування та синтаксичної підсвітки виконана на основі технології Windows Forms у середовищі .NET 8. Проєкт спроектовано як настільний додаток, який забезпечує зручний графічний інтерфейс, функціонал для роботи з текстом, форматування, синтаксичну підсвітку для популярних мов програмування та підтримку української мови через кодування UTF-8. У цьому підрозділі описано склад проєкту, структуру файлів, форми, їх компоненти та особливості реалізації.

Склад проєкту

Проєкт є рішенням Visual Studio, створеним для платформи .NET 8, і має модульну організацію, що забезпечує чіткий розподіл функціональних компонентів. Основна логіка зосереджена в одному основному класі форми, а допоміжні файли забезпечують ресурси, конфігурацію та дані для синтаксичної підсвітки.

					ДР.122.042.026.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

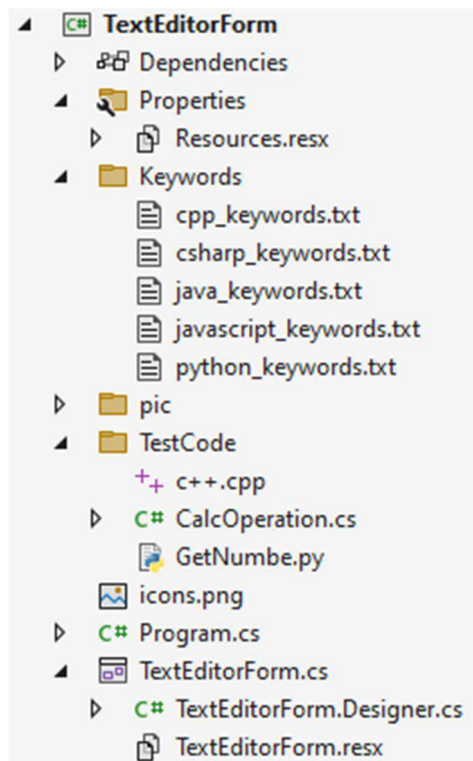


Рисунок 2.3.1 – Склад проєкту

Склад проєкту включає наступні елементи:

- TextEditorForm.cs — основний файл із кодом класу TextEditorForm, який успадковується від System.Windows.Forms.Form. Містить логіку обробки подій, форматування тексту, синтаксичної підсвітки, роботи з файлами та локалізації.
- TextEditorForm.Designer.cs — згенерований Visual Studio файл, який визначає компоненти графічного інтерфейсу форми, такі як панель інструментів (ToolStrip), текстовий редактор (RichTextBox), випадаючий список мов (ToolStripComboBox) і прогрес-бар (ToolStripProgressBar).
- TextEditorForm.resx — файл ресурсів, що містить зображення для кнопок панелі інструментів (наприклад, toolStripOpen.Image, toolStripBold.Image) розміром 90x90, які масштабуються до 50x50.
- Program.cs — точка входу програми, яка ініціалізує додаток і запускає форму TextEditorForm.
- Папка Keywords — містить текстові файли з ключовими словами для підтримуваних мов програмування (наприклад, cpp_keywords.txt,

csharp_keywords.txt, java_keywords.txt, python_keywords.txt, javascript_keywords.txt). Файли зберігаються в UTF-8 і завантажуються під час запуску програми для синтаксичної підсвітки.

Опис форм

Програма містить дві форми: основну форму текстового редактора (TextEditorForm) та допоміжну форму для пошуку і заміни тексту. Обидві форми створено з використанням Windows Forms, що забезпечує швидке проектування графічного інтерфейсу та інтеграцію з компонентами .NET.

1. Основна форма: TextEditorForm

Форма TextEditorForm є центральним елементом програми, відповідаючи за весь функціонал редактора. Вона має розмір 1024x500 пікселів і містить графічні компоненти для введення тексту, форматування, вибору мови та відображення прогресу обробки.

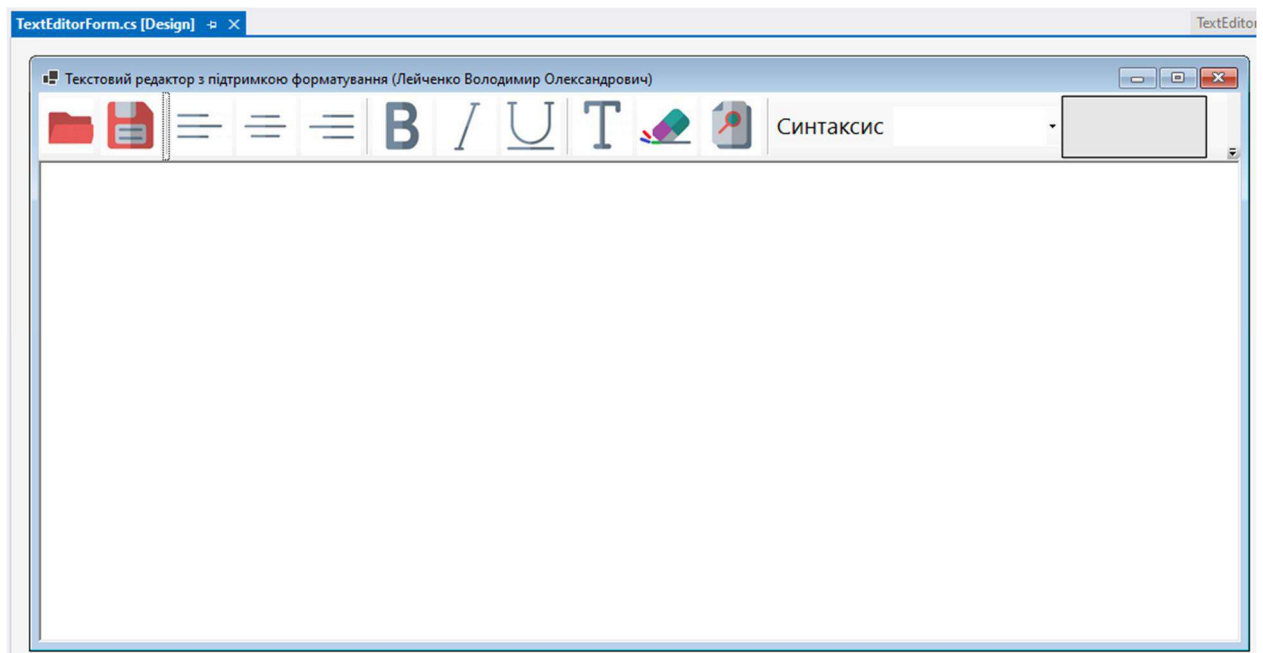


Рисунок 2.3.2 – Основна форма

Панель інструментів розташована у верхній частині форми, має висоту 57 пікселів і містить кнопки розміром 50x50 пікселів із масштабованими зображеннями 90x90. Кнопки включають:

- toolStripOpen — відкриває файл (.txt, .rtf, .cpp, .cs, .java, .py, .js).
- toolStripSave — зберігає файл.

- `toolStripLeft`, `toolStripCenter`, `toolStripRight` — вирівнювання тексту (ліворуч, по центру, праворуч).
- `toolStripBold`, `toolStripItalic`, `toolStripUnderline` — форматування тексту (жирний, курсив, підкреслений).
- `toolStripFont`, `toolStripColor` — вибір шрифту та кольору.
- `toolStripFind` — відкриває форму пошуку і заміни.
- `toolStripLabel1` — текстова мітка "Синтаксис".
- `languageCombo`: `ToolStripComboBox` — випадаючий список для вибору мови програмування ("None", "C++", "C#", "Java", "Python", "JavaScript").
- `progressBar`: `ToolStripProgressBar` — індикатор прогресу для обробки великих файлів, прихований за замовчуванням.

Текстовий редактор (`richTextBox`: `RichTextBox`) розташований нижче панелі інструментів, займає всю решту простору форми завдяки властивості `Dock = Fill`. Використовує шрифт `Consolas` розміром 12 для чіткого відображення коду. Підтримує форматування (шрифт, стиль, колір, вирівнювання), синтаксичну підсвітку та контекстне меню з командами `Undo`, `Redo`, `Cut`, `Copy`, `Paste`.

Контекстне меню реалізовано через `ContextMenuStrip`, прив'язане до `richTextBox`. Містить стандартні команди для роботи з текстом.

Форматування тексту реалізовано через методи `richTextBox.SelectionFont`, `SelectionColor` і `SelectionAlignment`, які змінюють властивості виділеного тексту. Обробники подій кнопок (`Bold_Click`, `Italic_Click` тощо) викликають відповідні методи.

Синтаксична підсвітка виконується асинхронно в методі `HighlightSyntaxAsync`, який очищає `richTextBox`, обробляє текст у тимчасовому `RichTextBox` і відновлює підсвічений текст у форматі `RTF`. Використовує регулярні вирази для виділення ключових слів, рядків, чисел і коментарів.

Методи `Open_Click` і `Save_Click` використовують `OpenFileDialog` і `SaveFileDialog` для роботи з файлами. Підтримується UTF-8 із резервним кодуванням Windows-1251 для старих файлів.

Для дебонсингу використовується Таймер `debounceTimer` із затримкою 300 мс обмежує частоту викликів підсвітки при зміні тексту.

Прогрес-бар відображається під час повної підсвітки тексту, оновлюється через `progressBar.Value` пропорційно кількості оброблених рядків.

Підтримка UTF-8 забезпечує коректне відображення українських символів у тексті та файлах ключових слів.

2. Форма «Пошук і заміна»

Ця форма створюється динамічно в методі `Find_Click` і не має окремого файлу в проєкті, оскільки є частиною логіки `TextEditorForm`. Вона призначена для пошуку та заміни тексту в `richTextBox`.

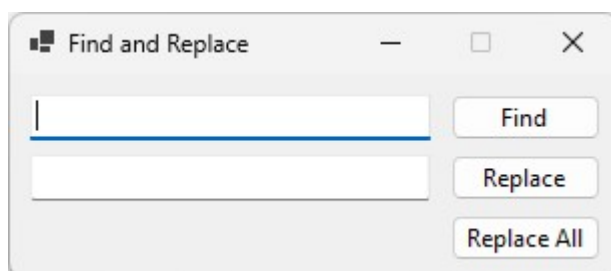


Рисунок 2.3.3 – Пошук і заміна

Форма (`findForm: Form`): Має розмір 320x140 пікселів, фіксовані межі (`FormBorderStyle = FixedSingle`) і розташовується в центрі батьківської форми (`StartPosition = CenterParent`).

Поле для пошуку (`findBox: TextBox`): Розташоване за координатами (10, 10), має ширину 200 пікселів. Використовується для введення шуканого тексту.

Поле для заміни (`replaceBox: TextBox`): Розташоване за координатами (10, 40), також шириною 200 пікселів. Призначене для введення тексту заміни.

Кнопка `findButton` ("Find", координати 220, 10) — виконує пошук наступного входження за допомогою `richTextBox.Text.IndexOf`.

Кнопка `replaceButton` ("Replace", координати 220, 40) — замінює поточне входження та шукає наступне.

Кнопка `replaceAllButton` ("Replace All", координати 220, 70) — замінює всі входження за допомогою `Regex.Replace`.

Пошук використовує `IndexOf` із підтримкою ігнорування регістру для швидкого пошуку. Якщо текст не знайдено, відображається повідомлення через `MessageBox`.

Поодинокі заміни перевіряє відповідність виділеного тексту і замінює його через `richTextBox.SelectedText`. Масова заміна використовує регулярні вирази для ефективної обробки.

Форма створюється під час виклику `Find_Click` і знищується після закриття, що економить ресурси.

Логіка розподілена за методами в `TextEditorForm.cs`, кожен із яких відповідає за окрему функцію (форматування, підсвітка, робота з файлами). Це полегшує підтримку та розширення.

Асинхронна підсвітка через `Task.Run` запобігає блокуванню інтерфейсу. Дебонсинг обмежує частоту підсвітки при редагуванні.

Часткова підсвітка (лише поточний і сусідній рядки) оптимізує обробку великих файлів.

Використання `UTF8Encoding` із BOM для файлів і `Encoding.GetEncoding(1251)` як резервне кодування забезпечує коректне відображення українських символів.

Файли ключових слів у UTF-8 дозволяють додавати українські коментарі чи ідентифікатори.

Помилки підсвітки перехоплюються в `HighlightSyntaxAsync` із виведенням повідомлення через `MessageBox` і відновленням оригінального тексту.

					ДР.122.042.026.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Перевірка `richTextBox.IsHandleCreated` усуває помилки `Invoke or BeginInvoke`.

Зображення кнопок зберігаються в `TextEditorForm.resx` і масштабуються до 50x50 через `toolStrip.ImageScalingSize` і `ImageScaling = SizeToFit`.

Програмна реалізація текстового редактора складається з основної форми `TextEditorForm`, допоміжної форми пошуку і заміни та набору файлів, що включають код, ресурси та дані для підсвітки. Структура проєкту є компактною, з чітким розподілом функціоналу між файлами `TextEditorForm.cs` (логіка), `TextEditorForm.Designer.cs` (дизайн) і `TextEditorForm.resx` (зображення). Використання Windows Forms забезпечило швидке створення графічного інтерфейсу, а .NET 8 — підтримку асинхронної обробки, UTF-8 і регулярних виразів. Реалізація є модульною, продуктивною та розширюваною, відповідаючи вимогам до текстового редактора з форматуванням і синтаксичною підсвіткою.

Висновки до розділу 2

Розділ присвячений проєктуванню та розробці текстового редактора з підтримкою форматування та синтаксичної підсвітки на основі Windows Forms і .NET 8. Проведено детальний аналіз алгоритмів, спроектовано структуру програми та здійснено її програмну реалізацію, що дозволило досягти поставлених цілей.

Вибір алгоритмів для синтаксичної підсвітки, пошуку, заміни тексту та форматування базується на їхній ефективності та відповідності вимогам. Використання прямого пошуку підрядків і регулярних виразів для підсвітки забезпечує точність і гнучкість, а асинхронна обробка та дебонсинг оптимізують продуктивність при роботі з великими файлами. Алгоритми форматування та роботи з файлами мають мінімальну обчислювальну складність, що гарантує швидкодію. Застосування UTF-8 як основного кодування забезпечує коректну підтримку української мови.

					ДР.122.042.026.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура програми побудована навколо єдиного класу `TextEditorForm`, який інкапсулює весь функціонал, включаючи графічний інтерфейс, форматування, підсвітку та роботу з файлами. Використання компонентів `Windows Forms`, таких як `RichTextBox` і `ToolStrip`, дозволило спростити розробку інтерфейсу, а модульна організація методів забезпечує зрозумілість і розширюваність коду. Додавання нових мов програмування можливе через файли ключових слів, що підвищує гнучкість програми.

Програмна реалізація включає основну форму `TextEditorForm` для редагування тексту та допоміжну форму для пошуку і заміни. Проект складається з файлів коду (`TextEditorForm.cs`, `TextEditorForm.Designer.cs`, `Program.cs`), ресурсів (`TextEditorForm.resx`) і папки `Keywords` із ключовими словами. Реалізація враховує продуктивність (асинхронність, дебонсинг), локалізацію (UTF-8) та обробку помилок, що робить редактор стабільним і зручним.

Розроблений текстовий редактор відповідає вимогам щодо функціональності, продуктивності та підтримки української мови. Використання `.NET 8` і `Windows Forms` забезпечило ефективну реалізацію, створивши основу для подальшого вдосконалення, наприклад, додавання автодоповнення чи інтеграції з хмарними сервісами.

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1. Мінімальні вимоги до системи

Для коректної роботи текстового редактора з підтримкою форматування та синтаксичної підсвітки, розробленого на основі Windows Forms і .NET 8, необхідно забезпечити відповідність системи мінімальним технічним вимогам. Програма призначена для запуску на операційних системах Windows, що забезпечує широку сумісність із сучасними комп'ютерами. Операційна система має бути Windows 10 версії 1809 або новішої, або Windows 11, оскільки ці версії підтримують .NET 8 і забезпечують стабільну роботу Windows Forms. Для встановлення та роботи програми потрібен .NET 8 Runtime, який можна завантажити з офіційного сайту Microsoft, якщо він ще не встановлений.

Щодо апаратного забезпечення, програма є легкою і не потребує значних ресурсів. Процесор із тактовою частотою від 1 ГГц забезпечить комфортну швидкість навіть на бюджетних пристроях. Оперативна пам'ять обсягом 2 ГБ достатня для виконання всіх функцій редактора, включаючи обробку великих текстових файлів. Для зберігання програми та її даних, таких як файли ключових слів і тимчасові журнали, потрібно близько 50 МБ вільного місця на диску. Роздільна здатність екрана від 1024x768 пікселів забезпечує коректне відображення графічного інтерфейсу, хоча для зручності рекомендуються вищі значення, наприклад, 1920x1080.

Додаткові вимоги включають наявність миші або іншого вказівного пристрою для взаємодії з графічним інтерфейсом, а також клавіатуру для введення тексту. Програма не потребує підключення до Інтернету, оскільки всі функції, включаючи синтаксичну підсвітку, виконуються локально. Для коректного відображення українських символів і роботи з файлами в кодуванні UTF-8 система має підтримувати відповідні шрифти, такі як Consolas, які встановлені в Windows за замовчуванням.

					ДР.122.042.026.ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці вимоги дозволяють запускати текстовий редактор на більшості сучасних комп'ютерів, включаючи офісні ПК, ноутбуки та навчальні станції, забезпечуючи доступність програми для широкого кола користувачів.

3.2. Інтерфейс користувача

Інтерфейс текстового редактора, розробленого на основі Windows Forms і .NET 8, спроектовано з урахуванням простоти, інтуїтивності та функціональності, щоб забезпечити зручну роботу з текстом, форматуванням і синтаксичною підсвіткою. Графічний інтерфейс програми є компактним, з чітким розподілом елементів керування, що робить його зрозумілим як для початківців, так і для досвідчених користувачів. Основна форма редактора поєднує панель інструментів, область для редагування тексту та контекстне меню, забезпечуючи швидкий доступ до всіх функцій. Додатково передбачено допоміжну форму для пошуку і заміни тексту. Нижче описано основні компоненти інтерфейсу та їх призначення.

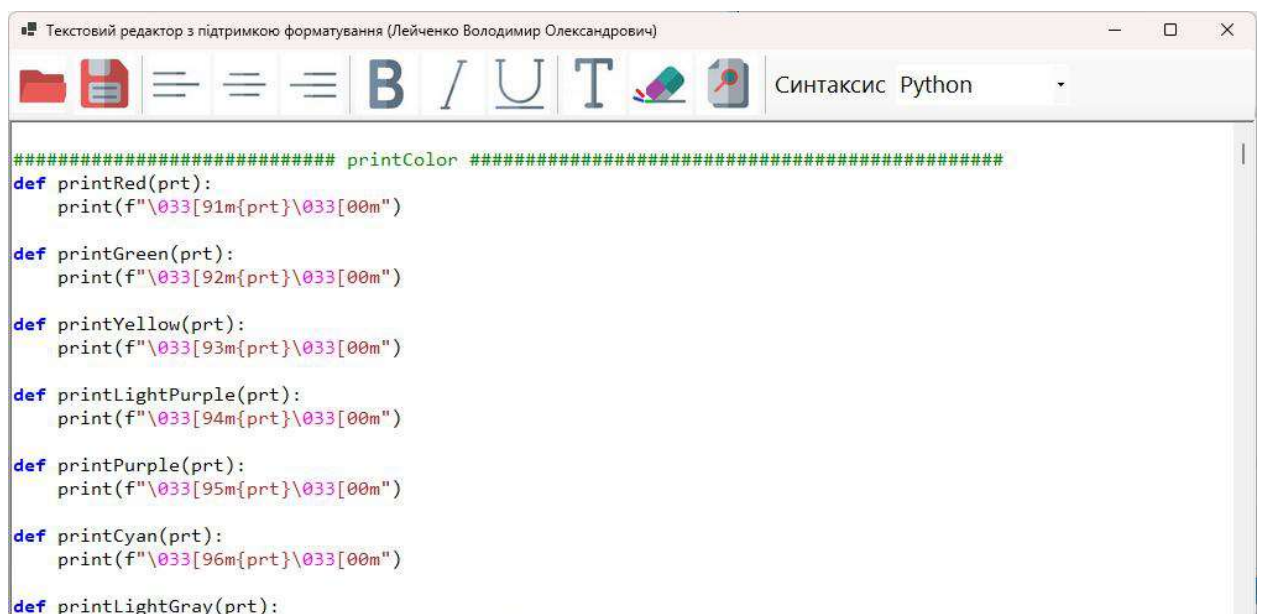


Рисунок 3.2.1 – Інтерфейс програми

Основна форма програми, яка має назву "Текстовий редактор з підтримкою форматування (Лейченко Володимир Олександрович)", відкривається з розміром 1024x450 пікселів. Вона містить два ключові

елементи: панель інструментів у верхній частині та текстову область, що займає решту простору. Панель інструментів (ToolStrip) розташована горизонтально і має висоту 57 пікселів. Вона включає набір кнопок розміром 50x50 пікселів із масштабованими зображеннями (оригінальний розмір 90x90), які відповідають за основні функції редактора. Кожна кнопка має іконку з прозорим фоном і відображається лише у вигляді зображення, що забезпечує чистий і сучасний вигляд. Кнопки організовано в логічні групи, розділені вертикальними сепараторами, для зручної навігації.

Перша група кнопок відповідає за роботу з файлами: кнопка "Відкрити" дозволяє вибрати файл (.txt, .rtf, .cpp, .cs, .java, .py, .js) через діалогове вікно, а кнопка "Зберегти" зберігає текст у вибраному форматі. Друга група керує вирівнюванням тексту: кнопки для вирівнювання ліворуч, по центру та праворуч змінюють розташування виділеного тексту. Третя група призначена для форматування: кнопки "Жирний", "Курсив" і "Підкреслений" перемикають відповідні стилі тексту, а кнопки "Шрифт" і "Колір" відкривають діалогові вікна для вибору шрифту та кольору. Кнопка "Знайти" відкриває допоміжну форму для пошуку і заміни тексту. Окремо розташована мітка "Синтаксис" поруч із випадаючим списком (ToolStripComboBox), який дозволяє обрати мову програмування для синтаксичної підсвітки ("None", "C++", "C#", "Java", "Python", "JavaScript"). На панелі також присутній прогрес-бар (ToolStripProgressBar), який відображається під час обробки великих файлів, наприклад, при повній підсвітці синтаксису, і прихований у звичайному режимі.

Основна область редагування тексту реалізована через компонент RichTextBox, який займає весь простір форми нижче панелі інструментів (1089x393 пікселів) завдяки властивості Dock = Fill. Текстовий редактор використовує шрифт Consolas розміром 12, що забезпечує чітке відображення коду та тексту. Він підтримує введення, редагування, форматування (шрифт, стиль, колір, вирівнювання) і синтаксичну підсвітку, де ключові слова виділяються синім жирним шрифтом, рядки — коричневим, числа —

					ДР.122.042.026.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

пурпуровим, а коментарі — зеленим. RichTextBox дозволяє вводити та відображати українські символи завдяки підтримці UTF-8. До текстової області прив'язане контекстне меню (ContextMenuStrip), яке викликається правою кнопкою миші. Меню містить стандартні команди: "Скасувати" (Undo), "Повернути" (Redo), "Вирізати" (Cut), "Копіювати" (Copy) і "Вставити" (Paste), що спрощують роботу з текстом.

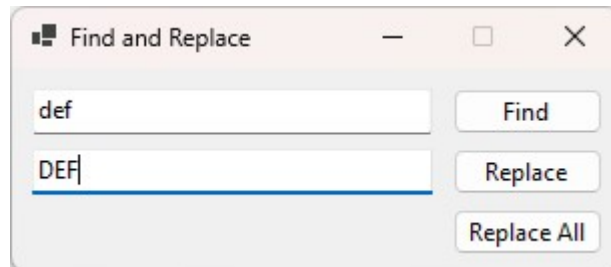


Рисунок 3.2.2 – Пошук і заміна

Допоміжна форма для пошуку і заміни відкривається при натисканні кнопки "Знайти" і має розмір 320x140 пікселів. Вона розташовується в центрі основної форми і має фіксовані межі, що запобігає зміні розміру. Форма містить два текстові поля: одне для введення шуканого тексту, друге — для тексту заміни, обидва шириною 200 пікселів. Поруч із полями розташовані три кнопки: "Знайти" виконує пошук наступного входження, "Замінити" замінює поточне входження і шукає наступне, а "Замінити всі" виконує заміну всіх входжень у тексті. Інтерфейс форми є мінімалістичним, з чітким розташуванням елементів, що забезпечує легкість використання.

Особливості інтерфейсу включають його адаптивність до потреб користувача. Панель інструментів забезпечує швидкий доступ до всіх основних функцій, а контекстне меню дублює стандартні операції для зручності. Прогрес-бар інформує про стан обробки великих файлів, а очищення текстової області під час підсвітки (з подальшим відновленням) запобігає візуальним артефактам. Підтримка UTF-8 гарантує коректне відображення українських символів у тексті, файлах і коментарях. Інтерфейс є компактним, але функціональним, що робить редактор придатним для

навчальних і професійних цілей, наприклад, редагування коду чи створення форматуваних документів.

Інтерфейс текстового редактора поєднує простоту та зручність із широкими можливостями форматування і синтаксичної підсвітки. Основна форма з панеллю інструментів і текстовою областю, доповнена контекстним меню та формою пошуку, забезпечує інтуїтивну взаємодію, відповідаючи потребам користувачів із різним рівнем підготовки.

3.3. Інструкція для роботи з програмою

Текстовий редактор, розроблений на основі Windows Forms і .NET 8, призначений для створення, редагування та форматування текстових документів із підтримкою синтаксичної підсвітки для популярних мов програмування. Програма забезпечує зручний графічний інтерфейс, коректну роботу з українськими символами через UTF-8 і високу продуктивність. Ця інструкція описує основні кроки для роботи з програмою, включаючи запуск, редагування тексту, форматування, роботу з файлами, синтаксичну підсвітку та пошук із заміною тексту.

Встановлення та запуск програми

Для початку роботи переконайтеся, що на комп'ютері встановлено операційну систему Windows 10 (версія 1809 або новіша) або Windows 11, а також .NET 8 Runtime, який можна завантажити з офіційного сайту Microsoft. Програма поставляється у вигляді виконуваного файлу TextEditor.exe, розташованого в папці bin/Debug або bin/Release проєкту. Разом із виконуваним файлом у цій же папці має бути підпапка Keywords, що містить файли ключових слів для мов програмування (cpp_keywords.txt, csharp_keywords.txt, java_keywords.txt, python_keywords.txt, javascript_keywords.txt) у форматі UTF-8.

Щоб запустити програму, виконайте подвійний клік на файлі TextEditor.exe. Після запуску відкриється основна форма з назвою "Текстовий редактор з підтримкою форматування (Лейченко Володимир

					ДР.122.042.026.ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Олександрович)". Форма містить панель інструментів у верхній частині та текстову область, готову до введення тексту. Якщо програма не запускається, перевірте наявність .NET 8 Runtime і папки Keywords.

Робота з текстом

Текст вводиться безпосередньо в текстову область (RichTextBox), яка займає більшу частину вікна. Для введення використовуйте клавіатуру, а для навігації — мишу або клавіші зі стрілками. Текстовий редактор підтримує українські символи завдяки кодуванню UTF-8, що дозволяє вводити, наприклад, коментарі чи ідентифікатори українською мовою. Для стандартних операцій із текстом, таких як копіювання, вирізання чи вставка, використовуйте контекстне меню, яке викликається правою кнопкою миші в текстовій області. Контекстне меню містить команди:

- "Скасувати" — скасовує останню дію.
- "Повернути" — відновлює скасовану дію.
- "Вирізати" — видаляє виділений текст і копіює його в буфер обміну.
- "Копіювати" — копіює виділений текст у буфер обміну.
- "Вставити" — вставляє текст із буфера обміну.

Форматування тексту

Для форматування тексту використовуйте кнопки на панелі інструментів, розташовані у верхній частині форми. Щоб відформатувати текст, спочатку виділіть потрібний фрагмент у текстовій області за допомогою миші або клавіатури (утримуйте Shift і використовуйте стрілки). Доступні функції форматування:

- Кнопка "Жирний" (іконка із літерою B) робить виділений текст жирним або скасовує цей стиль.
- Кнопка "Курсив" (іконка із похиленою літерою I) застосовує курсив до виділеного тексту.
- Кнопка "Підкреслений" (іконка із підкресленою літерою U) додає або прибирає підкреслення.

					ДР.122.042.026.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

- Кнопка "Шрифт" (іконка із літерою А та пензлем) відкриває діалогове вікно для вибору шрифту, його розміру та стилю.
- Кнопка "Колір" (іконка із палітрою) відкриває діалогове вікно для вибору кольору тексту.

Для вирівнювання тексту використовуйте кнопки:

- Вирівнювання ліворуч (іконка з рядками, вирівняними ліворуч).
- Вирівнювання по центру (іконка з центрованими рядками).
- Вирівнювання праворуч (іконка з рядками, вирівняними праворуч).

Ці функції дозволяють створювати форматовані документи, наприклад, із різними шрифтами чи кольорами, подібно до текстових процесорів.

Робота з файлами

Для відкриття існуючого файлу натисніть кнопку "Відкрити" (іконка із папкою) на панелі інструментів. У діалоговому вікні виберіть файл із підтримуваного формату (.txt, .rtf, .cpp, .cs, .java, .py, .js або будь-який інший). Програма автоматично розпізнає розширення файлу і, якщо це файл коду, встановлює відповідну мову програмування для синтаксичної підсвітки (наприклад, ".cpp" для C++). Файли читаються в кодуванні UTF-8, а якщо це неможливо, використовується Windows-1251 для старих файлів.

Щоб зберегти текст, натисніть кнопку "Зберегти" (іконка із дискетою). У діалоговому вікні виберіть формат (.rtf для форматованого тексту або .txt, .cpp, .cs, .java, .py, .js для звичайного тексту) і вкажіть ім'я файлу. Формат .rtf зберігає форматування (шрифти, кольори, вирівнювання), тоді як інші формати зберігають лише текст. Усі файли зберігаються в UTF-8 із BOM для коректного відображення українських символів.

Синтаксична підсвітка

Для активації синтаксичної підсвітки виберіть мову програмування у випадіючому списку поруч із міткою "Синтаксис" на панелі інструментів. Доступні варіанти: "None" (без підсвітки), "C++", "C#", "Java", "Python", "JavaScript". Після вибору мови програма автоматично підсвічує текст: ключові слова (наприклад, int, class, def) відображаються синім жирним

					ДР.122.042.026.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

шрифтом, рядки — коричневим, числа — пурпуровим, коментарі — зеленим. Підсвітка оновлюється при кожній зміні тексту, але з затримкою 300 мс, щоб уникнути надмірного навантаження. При відкритті великих файлів або зміні мови з'являється прогрес-бар, який показує стан обробки, а текстова область тимчасово очищається, після чого текст відновлюється з підсвіткою.

Щоб вимкнути підсвітку, виберіть "None" у списку мов. Програма підтримує підсвітку для файлів із українськими ідентифікаторами чи коментарями, оскільки ключові слова та текст обробляються в UTF-8.

Пошук і заміна тексту

Для пошуку тексту натисніть кнопку "Знайти" (іконка із лупою) на панелі інструментів. Відкриється допоміжна форма розміром 320x140 пікселів із двома текстовими полями та трьома кнопками. У верхнє поле введіть текст для пошуку, а в нижнє — текст для заміни (якщо потрібна заміна). Доступні дії:

- Кнопка "Знайти" шукає наступне входження введеного тексту, починаючи з поточної позиції курсора, і виділяє його в текстовій області. Пошук ігнорує регістр. Якщо текст не знайдено, з'явиться повідомлення "Текст не знайдено!".
- Кнопка "Замінити" замінює поточне виділене входження на текст із нижнього поля і автоматично шукає наступне.
- Кнопка "Замінити всі" замінює всі входження шуканого тексту в усьому документі.

Після завершення роботи закрийте форму, натиснувши хрестик у верхньому правому куті.

Рекомендації та особливості

Для роботи з великими файлами (тисячі рядків) дочекайтеся завершення обробки, коли прогрес-бар зникне. Щоб уникнути помилок, не змінюйте текст під час підсвітки. Якщо зображення кнопок не відображаються, перевірте наявність файлу TextEditorForm.resx із ресурсами. Для створення власних файлів ключових слів додайте текстовий файл у папку Keywords із назвою

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

{мова}_keywords.txt (наприклад, ruby_keywords.txt), де кожне ключове слово записано в новому рядку, і перезапустіть програму.

Програма є автономною і не потребує підключення до Інтернету. У разі виникнення помилок, таких як "Помилка підсвітки", оригінальний текст автоматично відновлюється. Для діагностики проблем увімкніть журналювання, додавши запис до файлу debug.log у папці програми.

Ця інструкція охоплює всі основні функції текстового редактора, дозволяючи користувачам ефективно працювати з текстом, кодом і форматованими документами, використовуючи зручний інтерфейс і підтримку української мови.

Висновки до розділу 3

Розділ присвячений опису керівництва користування текстовим редактором, розробленим на основі Windows Forms і .NET 8. Описано мінімальні системні вимоги, інтерфейс користувача та детальну інструкцію з використання програми, що забезпечує повне розуміння її функціональних можливостей.

Мінімальні вимоги до системи включають операційну систему Windows 10 (версія 1809 або новіша) або Windows 11, .NET 8 Runtime, процесор із частотою від 1 ГГц, 2 ГБ оперативної пам'яті, 50 МБ вільного місця на диску та роздільну здатність екрана від 1024x768 пікселів. Ці вимоги роблять програму доступною для більшості сучасних комп'ютерів, включаючи бюджетні пристрої, що сприяє її широкому застосуванню.

Інтерфейс користувача є інтуїтивним і компактним, поєднуючи панель інструментів із кнопками для форматування, роботи з файлами та вибору мови, текстову область із підтримкою синтаксичної підсвітки та контекстне меню для стандартних операцій. Допоміжна форма для пошуку і заміни тексту забезпечує додаткову зручність. Підтримка UTF-8 гарантує коректне

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

відображення українських символів, а прогрес-бар і очищення текстової області під час обробки підвищують візуальну комфортність.

Інструкція для роботи з програмою охоплює всі аспекти використання: запуск, редагування тексту, форматування, роботу з файлами, синтаксичну підсвітку та пошук із заміною. Вона детально описує дії користувача, включаючи обробку великих файлів і особливості локалізації, що робить програму доступною для користувачів із різним рівнем підготовки.

Розділ підтверджує, що текстовий редактор є зручним, функціональним і придатним для навчальних і професійних цілей. Інтерфейс і документація забезпечують легкість освоєння, а підтримка української мови та продуктивність роблять програму конкурентоспроможною.

					ДР.122.042.026.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Дипломна робота присвячена розробці текстового редактора з підтримкою форматування та синтаксичної підсвітки на основі технології Windows Forms у середовищі .NET 8. У процесі виконання роботи досягнуто поставленої мети — створено універсальний настільний додаток, який поєднує можливості форматування тексту, синтаксичну підсвітку для популярних мов програмування, підтримку української мови через UTF-8 і високу продуктивність при роботі з великими файлами.

У першому розділі визначено основну задачу розробки, проведено аналіз аналогічних систем (Notepad++, Visual Studio Code, Sublime Text, Microsoft Word) і обґрунтовано вибір технологій. Аналіз показав потребу в редакторі, який поєднує форматування та підсвітку коду, а .NET 8 і Windows Forms обрано завдяки їхній продуктивності, простоті створення інтерфейсу та підтримці сучасних стандартів програмування.

Другий розділ охоплює проектування та реалізацію програми. Обрано ефективні алгоритми для синтаксичної підсвітки (пошук підрядків, регулярні вирази), пошуку, заміни та форматування, оптимізовано продуктивність за допомогою асинхронної обробки та дебонсингу. Програма спроектована навколо єдиного класу TextEditorForm, який інкапсулює весь функціонал, забезпечуючи модульність і розширюваність. Реалізація включає основну форму для редагування тексту та допоміжну форму для пошуку і заміни, із підтримкою UTF-8 і обробкою помилок.

Третій розділ містить керівництво користування, описуючи системні вимоги, інтерфейс і детальну інструкцію. Програма є легкою, працює на більшості сучасних комп'ютерів і має інтуїтивний інтерфейс із панеллю інструментів, текстовою областю та контекстним меню. Інструкція охоплює всі аспекти роботи, від форматування до підсвітки, забезпечуючи зручність для користувачів.

Розроблений текстовий редактор успішно виконує поставлені завдання: підтримує форматування (шрифти, стилі, кольори, вирівнювання),

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

синтаксичну підсвітку для C++, C#, Java, Python і JavaScript, коректно обробляє українські символи та ефективно працює з великими файлами. Програма є конкурентоспроможною порівняно з аналогами, поєднуючи простоту Notepad++ із можливостями форматування, подібними до Microsoft Word. Вона придатна для навчальних цілей, професійного редагування коду та створення документів.

Перспективи подальшого розвитку включають додавання автодоповнення коду, інтеграцію з системами контролю версій, підтримку додаткових мов програмування та портування на інші платформи за допомогою .NET MAUI. Робота має практичне значення, демонструючи можливості Windows Forms і .NET 8 для створення функціональних настільних додатків, і може бути основою для вдосконалення текстових редакторів.

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Албахарі Дж., Албахарі Б. С# 9.0 у двох словах: Довідник / пер. з англ. О. П. Кондуфор. Київ: Фабула, 2022. 1088 с.
2. Батлер Н., Вілсон Д. Програмування на С# для початківців: Основи та практика. Харків: Фоліо, 2020. 512 с.
3. Вірт Н. Алгоритми та структури даних / пер. з англ. В. О. Григор'єв. Львів: Світ, 2019. 320 с.
4. Гаврилюк О. В. Основи програмування на С# у середовищі .NET. Київ: Видавнича група BHV, 2021. 432 с.
5. Гама Е., Хелм Р., Джонсон Р., Влісідес Дж. Прийоми об'єктно-орієнтованого проєктування. Патерни проєктування / пер. з англ. І. М. Кравець. Київ: Видавництво "Наш Формат", 2020. 448 с.
6. Гудрич М. Т., Тамассія Р., Голдвассер М. Х. Структури даних та алгоритми на Java / пер. з англ. О. В. Скрипник. Львів: Видавництво "Літопис", 2021. 704 с.
7. Джонс Т., Мак-Дональд А. .NET 8: Нові можливості та оптимізація продуктивності. Лондон: Apress, 2023. 512 р.
8. Кормен Т. Х., Лейзерсон Ч. І., Рівест Р. Л., Стайн К. Алгоритми: побудова й аналіз / пер. з англ. С. О. Семеріков. Київ: Видавнича група BHV, 2020. 1328 с.
9. Кук Д. Регулярні вирази: Практичний посібник / пер. з англ. Ю. М. Ковальчук. Харків: Ранок, 2022. 304 с.
10. Ліберті Дж. Програмування на С# для професіоналів. Київ: Діалектика, 2021. 960 с.
11. МакКоннелл С. Досконалий код: Практичний посібник з розробки програмного забезпечення / пер. з англ. О. В. Матвієнко. Київ: Фабула, 2020. 896 с.
12. Мейєрс С. Ефективне програмування: 55 способів покращити код / пер. з англ. В. І. Лозинський. Львів: Світ, 2019. 352 с.

					ДР.122.042.026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

13. Нейгл К. Windows Forms Programming with C#. Нью-Йорк: Manning Publications, 2020. 672 р.
14. Петцольд Ч. Програмування для Windows: .NET і Windows Forms / пер. з англ. О. Л. Шевчук. Київ: Видавництво "Книга", 2021. 784 с.
15. Ріхтер Дж. CLR via C#: Програмування на платформі .NET / пер. з англ. І. П. Сидоренко. Київ: Діалектика, 2020. 912 с.
16. Скіна Д. Алгоритми для програмістів: Практичний підхід / пер. з англ. М. С. Коваленко. Харків: Фоліо, 2021. 640 с.
17. Стівенс Р. Т. Патерни проектування: Практичний посібник для розробників / пер. з англ. О. В. Кравець. Київ: Видавництво "Наш Формат", 2022. 416 с.
18. Троелсен Е., Джепиксе Ф. C# 8.0 і платформа .NET Core: Посібник для розробників / пер. з англ. В. О. Гриненко. Київ: Видавнича група BHV, 2020. 1248 с.
19. Фаулер М. Рефакторинг: Поліпшення існуючого коду / пер. з англ. Ю. В. Литвиненко. Київ: Фабула, 2021. 432 с.
20. Офіційна документація Microsoft .NET 8 [Електронний ресурс]. Режим доступу: <https://docs.microsoft.com/dotnet/core/whats-new/dotnet-8>. Дата звернення: 20.04.2025.
21. Документація Windows Forms [Електронний ресурс]. Режим доступу: <https://docs.microsoft.com/dotnet/desktop/winforms/>. Дата звернення: 20.04.2025.
22. Репозиторій Notepad++ на GitHub [Електронний ресурс]. Режим доступу: <https://github.com/notepad-plus-plus/notepad-plus-plus>. Дата звернення: 20.04.2025.
23. Visual Studio Code: Офіційний сайт [Електронний ресурс]. Режим доступу: <https://code.visualstudio.com/docs>. Дата звернення: 20.04.2025.
24. Регулярні вирази у .NET: Документація [Електронний ресурс]. Режим доступу: <https://docs.microsoft.com/dotnet/standard/base-types/regular-expressions>. Дата звернення: 20.04.2025.

ДОДАТКИ

Програмний код модулів

```

namespace TextEditor
{
    partial class TextEditorForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
            System.ComponentModel.ComponentResourceManager(typeof(TextEditorForm));
            toolStrip = new ToolStrip();
            toolStripOpen = new ToolStripButton();
            toolStripSave = new ToolStripButton();
            toolStripSeparator1 = new ToolStripSeparator();
            toolStripLeft = new ToolStripButton();
            toolStripCenter = new ToolStripButton();
            toolStripRight = new ToolStripButton();
            toolStripSeparator2 = new ToolStripSeparator();
            toolStripBold = new ToolStripButton();
            toolStripItalic = new ToolStripButton();
            toolStripUnderline = new ToolStripButton();
            toolStripSeparator3 = new ToolStripSeparator();
            toolStripFont = new ToolStripButton();
            toolStripColor = new ToolStripButton();
            toolStripFind = new ToolStripButton();

```