



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

**Лабораторна робота № 3**  
із дисципліни «Технології розробки програмного забезпечення»  
Тема: «Основи проектування розгортання.»

Виконав  
Студенти групи ІА-31:  
Корнійчук М.Р

Перевірив:  
Мягкий М.Ю

Київ 2025

**Тема:** Основи проектування розгортання.

**Мета:** Навчитися проектувати діаграми розгортання та компонентів для системи що проектується, а також розробляти діаграми взаємодії, а саме діаграми послідовностей, на основі сценаріїв зроблених в попередній лабораторній роботі.

**Посилання на гіт:** [https://github.com/MkEger/Lab3\\_trpz](https://github.com/MkEger/Lab3_trpz)

**Варіант :**

### **3. Текстовий редактор (strategy, command, observer, template method, flyweight, SOA)**

Текстовий редактор повинен вміти розпізнавати текстові файли в будь-якій кодуванні, мати розширені функції редагування: макроси, сніппети, підказки, закладки, перехід на рядок / сторінку, підсвічування синтаксису (для однієї мови програмування або розмітки на розсуд студента).

#### **Завдання:**

- Ознайомитись з короткими теоретичними відомостями.
- Проаналізувати діаграми створені в попередній лабораторній роботі а також тему системи та спроектувати діаграму розгортання використання відповідно до обраної теми лабораторного циклу.
- Розробити діаграму компонентів для проєктованої системи.
- Розробити діаграму розгортання для проєктованої системи.
- Розробити як мінімум дві діаграми послідовностей для сценаріїв прописаних в попередній лабораторній роботі.
- На основі спроектованих діаграм розгортання та компонентів доопрацювати програмну частину системи. Реалізація системи, додатково до попередньої реалізації, повинна містити як мінімум дві візуальні форми. В системі вже повинен бути повністю реалізована архітектура (повний цикл роботи з даними від вводу на формі до збереження їх в БД і подальшій виборці з БД та відображенням на UI).
- Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму розгортання з описом, діаграму компонентів системи

з описом, діаграми послідовностей, а також вихідний код системи, який було додано в цій лабораторній роботі.

**Хід роботи:**

**Тема проекту: Text Editor**

- 1) Ознайомитись з короткими теоретичними відомостями.
- 2) Проаналізувати діаграми створені в попередній лабораторній роботі а також тему системи та спроектувати діаграму розгортання використання відповідно до обраної теми лабораторного циклу

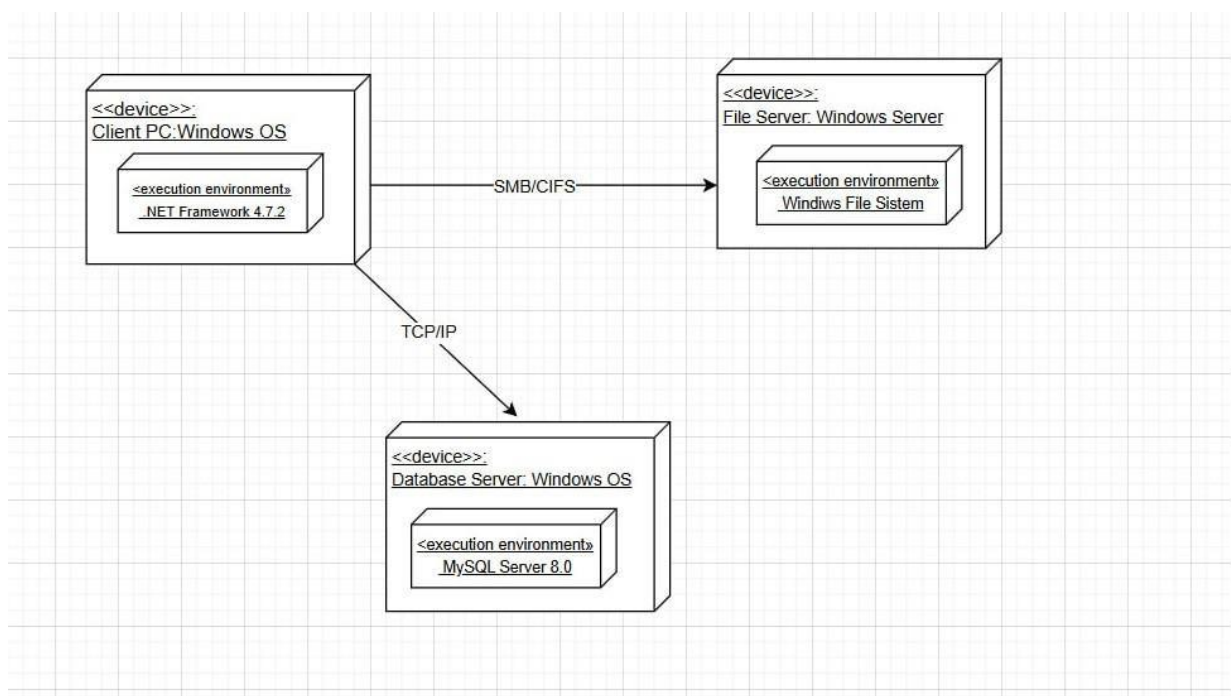


Рис.1 – Діаграма розгортання

Ця діаграма розгортання показує компоненти системи для текстового редактора і те, як вони розгорнуті в різних середовищах.

#### 1. Client PC Windows OS:

а) Вузол Client PC містить додаток TextEditor, який виконується в середовищі .NET Framework 4.7.2, де користувач взаємодіє з текстовим редактором.

б) TextEditor відповідає за створення, редагування та збереження текстових документів, управління налаштуваннями редактора та відстеження нещодавно відкритих файлів, а також за взаємодію з файловою системою та базою даних.

#### 2. File Server Windows Server:

а) Вузол File Server містить середовище виконання Windows File System - компонент для централізованого зберігання файлів.

б) Windows File System виступає як файлове сховище, де зберігаються текстові документи користувачів, доступ до яких здійснюється через протокол

SMB/CIFS. с) Цей сервер забезпечує спільний доступ до файлів для кількох користувачів та централізоване управління документами.

### 3. Database Server Windows OS:

а) Вузол Database Server представляє сервер бази даних з MySQL Server 8.0 як середовищем виконання для зберігання метаданих системи.

б) MySQL Server зберігає інформацію про документи (назви файлів, шляхи, дати створення/модифікації), налаштування редактора (тема, шрифт, розмір), історію нещодавно відкритих файлів та налаштування кодувань тексту.

с) База даних забезпечує швидкий пошук та отримання метаданих про документи, не зберігаючи сам контент файлів.

### 4. Протоколи зв'язку:

а) TCP/IP використовується для з'єднання клієнтського додатка з сервером бази даних для операцій читання/запису метаданих.

б) SMB/CIFS забезпечує доступ до файлового сервера для операцій з текстовими документами (відкриття, збереження, створення нових файлів).

## 2. Розробити діаграму компонентів для проєктованої системи

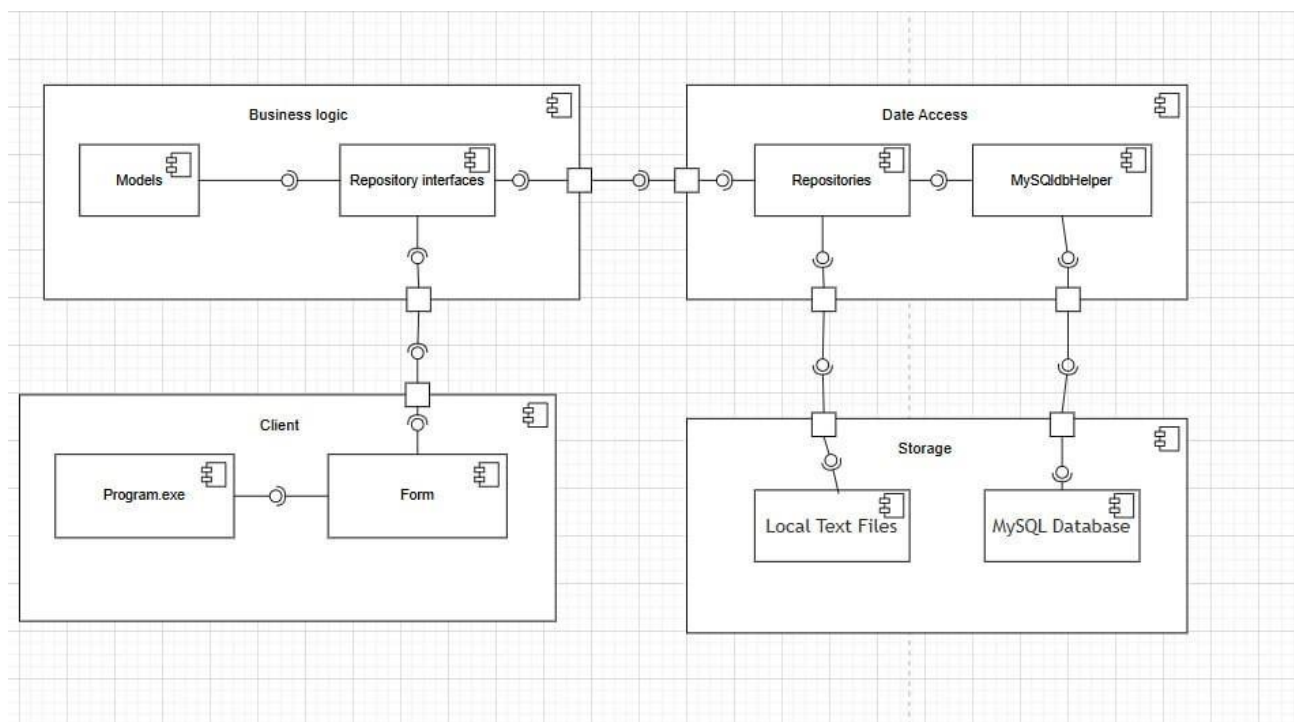


Рис.2 – Діаграма компонентів

Опис діаграми:

1. Client:

a) Містить два основні компоненти: Program.exe (точка входу в додаток) та Form(візуальний інтерфейс для користувача)

b) Program.exe ініціалізує інтерфейс користувача, який забезпечує всю взаємодію з системою для створення, редагування та збереження документів.

2. Business Logic:

a) Містить два ключові компоненти: Models - сутності даних (Document, TextEncoding, RecentFile, EditorSettings), що представляють основні об'єкти предметної області. Repository Interfaces - набір інтерфейсів, які визначають контракти для операцій з даними та забезпечують абстракцію від способів зберігання.

3. Data Access:

a) Містить два компоненти для роботи з даними: Repositories - реалізації репозиторіїв для роботи з базою, що забезпечують усі операції даними. MySqlDatabaseHelper - допоміжний компонент для управління підключенням до MySQL, створення таблиць та виконання SQL-операцій.

4. Storage:

a) Містить два компоненти зберігання: MySQL Database - централізована база даних для зберігання всіх даних системи: метаінформації про документи, налаштування редактора, історії файлів та кодувань. Local Text Files - локальна файлова система для зберігання фактичного текстового контенту документів.

4. Розробити як мінімум дві діаграми послідовностей для сценаріїв прописаних в попередній лабораторній роботі

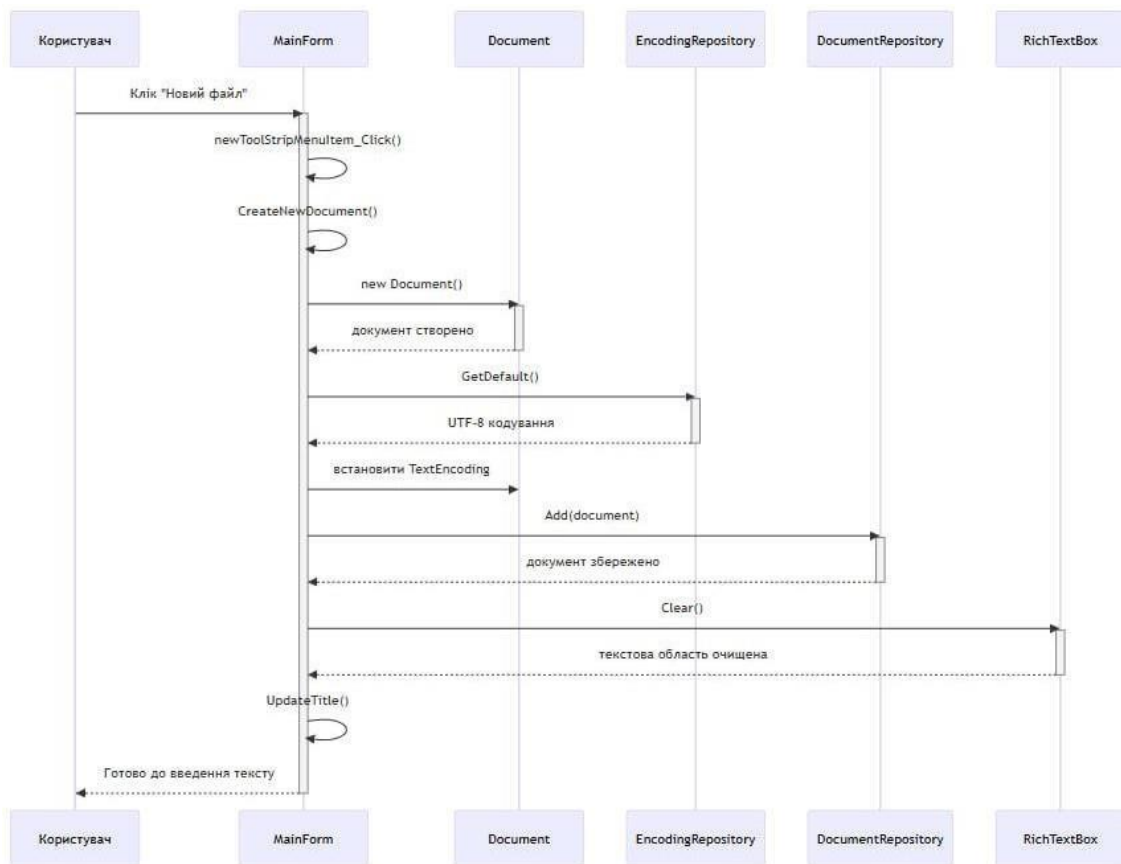


Рисунок 3.1. Діаграма послідовності «Створення нового документа»

### Основний потік подій:

1. Ініціація користувачем: Користувач натискає кнопку "Новий файл" в головному меню, що активує відповідний обробник подій у MainForm.
2. Створення об'єкта документа: MainForm викликає метод `CreateNewDocument()`, який створює новий екземпляр класу `Document` з базовими параметрами (назва "Untitled.txt", порожній вміст, ID кодування).
3. Налаштування кодування: Система звертається до `EncodingRepository` для отримання кодування за замовчуванням (UTF-8) та встановлює його для нового документа.
4. Збереження в репозиторій: Новостворений документ додається до `DocumentRepository` для подальшого відстеження та управління життєвим циклом документа.

5. Підготовка інтерфейсу: MainForm очищає текстову область (RichTextBox) від попереднього вмісту та оновлює заголовок вікна для відображення назви нового документа.

6. Завершення: Система повідомляє користувача про готовність до введення тексту, після чого користувач може починати роботу з новим документом.

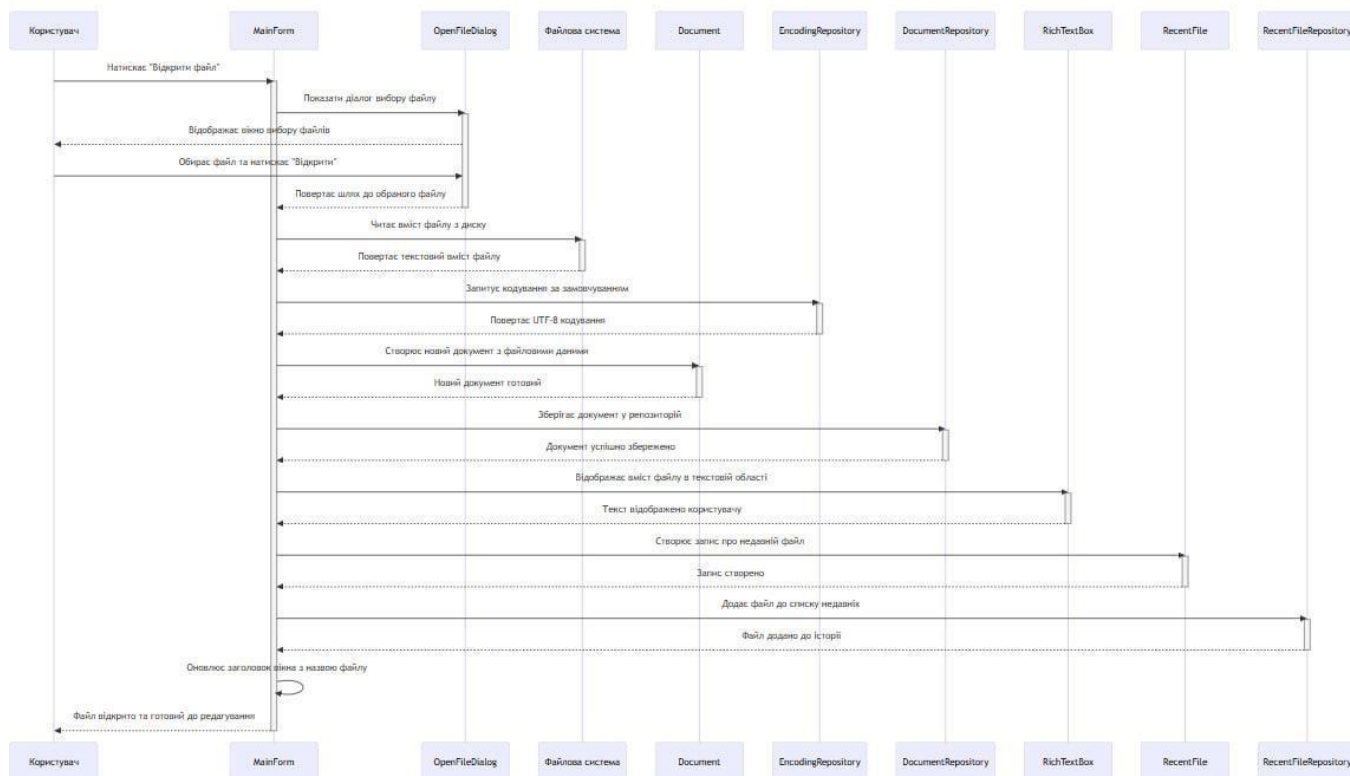


Рисунок 3.2. Діаграма послідовності «Відкриття існуючого файлу»

1. Ініціація процесу: Користувач натискає кнопку "Відкрити файл" в головному меню програми.
2. Діалог вибору файлу: Система відкриває стандартне вікно вибору файлів, де користувач може знайти та обрати потрібний текстовий файл.
3. Читання файлу: Після підтвердження вибору система звертається до файлової системи для читання вмісту обраного файлу.
4. Налаштування кодування: Система запитує кодування за замовчуванням (UTF-8) для правильного відображення тексту.



5. Створення документа: Система створює новий об'єкт документа з отриманим вмістом та метаданими файлу.
6. Збереження в репозиторій: Новий документ зберігається у внутрішньому репозиторії для подальшого відстеження змін.
7. Відображення тексту: Вміст файлу завантажується в текстову область для перегляду та редагування користувачем.
8. Оновлення історії: Система створює запис про недавно відкритий файл та додає його до списку нещодавніх файлів.
9. Завершення: Система оновлює заголовок вікна з назвою файлу та повідомляє користувача про готовність до роботи з документом.

**Висновок:** У процесі виконання лабораторної роботи було спроектовано архітектуру системи Text Editor. Було створено діаграму розгортання, яка показує взаємодію клієнтського додатка з файловим сервером і сервером бази даних. Система використовує MySQL Server для збереження метаданих про документи та Windows File System для зберігання текстових файлів. Розроблена діаграма компонентів відобразила структуру програми, поділену на рівні клієнта, бізнес-логіки, доступу до даних і сховища. Визначено основні компоненти, такі як Program.exe, MainForm, Repositories та MySqlDatabaseHelper. Також було створено діаграми послідовностей для сценаріїв “Створення нового документа” та “Відкриття існуючого файлу”. Вони допомогли зрозуміти логіку взаємодії об'єктів і послідовність виконання операцій у системі.

## Відповіді на контрольні питання:

### 1. Що собою становить діаграма розгортання?

Діаграма розгортання (Deployment Diagram) — це UML-діаграма, яка показує фізичне розміщення програмних компонентів на апаратних пристроях. Вона описує, як система реалізується на реальному обладнанні.

### 2. Які бувають види вузлів на діаграмі розгортання?

Види вузлів:

- Апаратні вузли (device) — фізичні пристрої, наприклад сервер, комп'ютер, мобільний телефон.
- Виконувані вузли (execution environment) — середовища виконання програм, наприклад JVM, веб-сервер, контейнер додатків.

### 3. Які бувають зв'язки на діаграмі розгортання?

- Комунікаційні зв'язки (communication paths) — показують обмін інформацією між вузлами.
- Залежності (dependencies) — відображають залежність одного компонента або вузла від іншого.

### 4. Які елементи присутні на діаграмі компонентів?

Основні елементи: компоненти, інтерфейси та зв'язки (залежності) між ними.

### 5. Що становлять собою зв'язки на діаграмі компонентів?

Зв'язки показують залежність одного компонента від іншого або реалізацію певного інтерфейсу. Вони вказують, які частини системи використовують інші.

### 6. Які бувають види діаграм взаємодії?

Основні види:

- Діаграма послідовностей (Sequence Diagram)
- Діаграма комунікації (Communication Diagram)
- Діаграма з часовими обмеженнями (Timing Diagram)
- Діаграма огляду взаємодії (Interaction Overview Diagram)

7. Для чого призначена діаграма послідовностей?

Діаграма послідовностей показує послідовність обміну повідомленнями між об'єктами у часі. Вона описує, як реалізується певний сценарій або варіант використання.

8. Які ключові елементи можуть бути на діаграмі послідовностей?

Актори, об'єкти (lifelines), повідомлення, активності (activation bars) та фрагменти управління (alt, loop, opt).

9. Як діаграми послідовностей пов'язані з діаграмами варіантів використання?

Діаграма послідовностей деталізує поведінку сценарію, описаного у варіанті використання. Кожен варіант використання може мати свою діаграму послідовностей.

10. Як діаграми послідовностей пов'язані з діаграмами класів?

Діаграма послідовностей показує взаємодію об'єктів, які створені на основі класів, зображених на діаграмі класів. Класи визначають типи об'єктів, що взаємодіють у послідовності.