



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №3
Технологія розробки програмного забезпечення
*«Діаграма розгортання. Діаграма компонентів.
Діаграма взаємодій та послідовностей.»*

Виконала студентка
групи ІА-23:
Щрубович Н. С.

Перевірів:
Мягкий М. Ю.

Київ 2024

ЗМІСТ

<i>Теоретичні відомості.....</i>	<i>3</i>
<i>Хід Роботи.....</i>	<i>4</i>
Діаграма розгортання.....	4
Діаграма компонентів.....	6
Діаграма послідовностей.....	9
<i>Висновок</i>	<i>10</i>

Тема: Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.

Мета: Навчитися розробляти діаграму розгортання. діаграма компонентів. діаграма взаємодій та послідовностей.

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Розробити діаграму розгортання для проектованої системи.
3. Розробити діаграму компонентів для проектованої системи.
4. Розробити діаграму послідовностей для проектованої системи.
5. Скласти звіт про виконану роботу

..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).

Теоретичні відомості

- Діаграма розгортання (Deployment Diagram) – важливий інструмент у моделюванні архітектури програмних систем. Вона показує, як програмні компоненти (якщо вони є частинами системи) розгортаються на фізичних пристроях, таких як сервери, комп'ютери або інші мережеві вузли. Діаграма розгортання відображає фізичну структуру системи, вказуючи на те, як різні частини програмного забезпечення взаємодіють між собою через апаратні компоненти. Вона допомагає зрозуміти конфігурацію мережі, з'єднання між вузлами та розподіл ресурсів на різних пристроях.

- Діаграма компонентів (Component Diagram) – показує структуру системи з точки зору її компонентів та їх взаємодії. Вона використовується для візуалізації фізичних компонентів програмної системи, таких як модулі, бібліотеки, сервіси або інші програмні одиниці, та їх зв'язків. Діаграма компонентів корисна для визначення того, як різні частини системи будуть взаємодіяти між собою на етапі реалізації та для аналізу масштабованості і модульності системи.

- Діаграма послідовностей ((Sequence Diagram) – ілюструє, як об'єкти або компоненти взаємодіють один з одним протягом певного сценарію або процесу. Вона зображує обмін повідомленнями між об'єктами в часі, де час йде вертикально, а повідомлення між об'єктами — горизонтально. Діаграма послідовностей дозволяє візуалізувати порядок виконання дій і процесів, що допомагає у розумінні логіки роботи системи на етапі розробки та тестування.

Хід Роботи

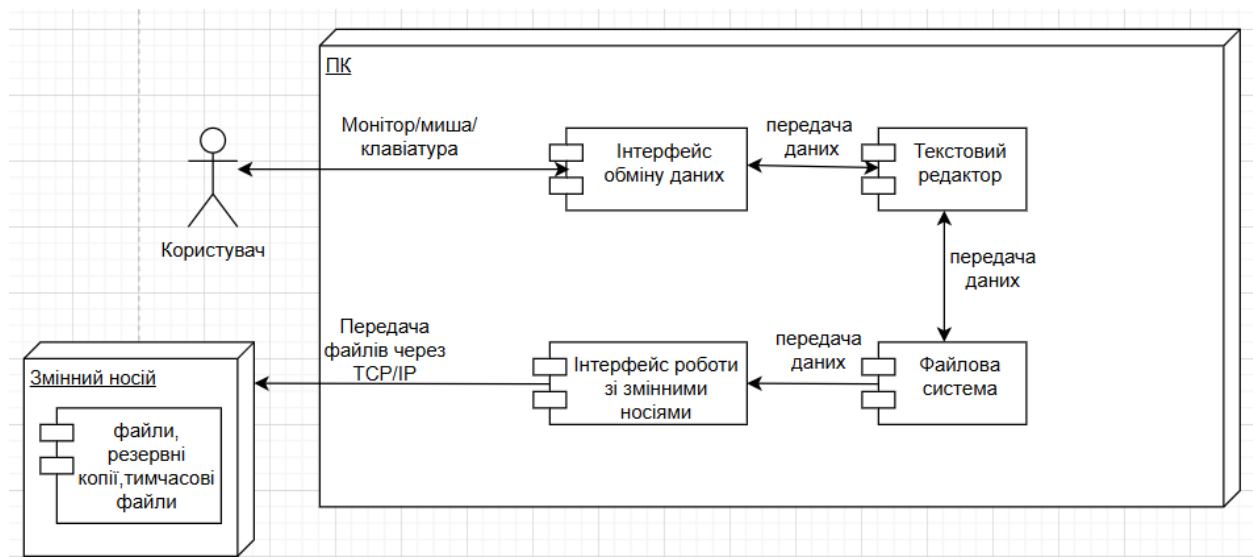


Рис 1. Діаграма розгортання

Діаграма розгортання

1. Фізичні вузли (ноди):

Особистий пристрій (ПК) - основний вузол, де розташовані всі ключові компоненти текстового редактора. Цей вузол включає такі частини:

- Інтерфейс обміну даними з користувачем:
 - Забезпечує взаємодію між користувачем (актором) та текстовим редактором.
 - Користувач вводить команди та текст через клавіатуру і мишу, які з'єднуються з цим інтерфейсом.
 - Відображає результати роботи текстового редактора на моніторі.
- Текстовий редактор:
 - Головний компонент, що виконує редагування тексту.
 - Включає функції:
 - Робота з файлами: відкриття, збереження, редагування.
 - Управління макросами та сніпшетами.
 - Підсвічування синтаксису для програмного коду чи тексту.
 - Обробляє запити користувача, передає дані до файлової системи та керує іншими компонентами.
- Файлова система:
 - Відповідає за збереження файлів текстового редактора.

- Використовується текстовим редактором для створення, читання, редагування та збереження текстових файлів.
- Інтерфейс роботи зі змінними носіями:
 - Виконує обмін даними між файловою системою ПК та зовнішнім змінним носієм.
 - Підтримує завантаження текстових файлів на зовнішній носій або їхнє завантаження з носія у систему.

Змінний носій - окремий фізичний пристрій, який містить такі дані:

- Текстові файли: Збережені документи, з якими працює текстовий редактор.
- Резервні копії: Створюються для збереження даних користувача у випадку відновлення.
- Тимчасові файли: Використовуються текстовим редактором для проміжних операцій (кешування, автозбереження). Змінний носій взаємодіє з інтерфейсом роботи зі змінними носіями через протокол TSP/IP.

2. Актор(користувач):

- Використовує пристрої введення (клавіатура, миша) для взаємодії з текстовим редактором через інтерфейс обміну даними.
- Надсилає команди для відкриття, редагування чи збереження текстових файлів.

3. Зв'язки між компонентами

- Клавіатура/Миша/Монітор → Інтерфейс обміну даними:
 - Канал взаємодії між користувачем і текстовим редактором для передачі команд.
- Передача даних:
 - Використовується для передачі даних між компонентами, зокрема:
 - Між інтерфейсом обміну даними і текстовим редактором.
 - Між текстовим редактором і файловою системою.
 - Між інтерфейсом роботи зі змінними носіями і файловою системою.
- TSP/IP:
 - Канал зв'язку між особистим пристроєм (ПК) та змінним носієм, що забезпечує обмін файлами між цими вузлами.

4. Взаємодія компонентів

1. Користувач (актор):
 - Вводить дані або команди через клавіатуру/мишу.
 - Інтерфейс обміну даними обробляє введені команди.
2. Інтерфейс обміну даними:
 - Передає команди до текстового редактора для виконання певних дій, таких як відкриття чи збереження файлу.
3. Текстовий редактор:

- Виконує команди, отримані від інтерфейсу обміну даними.
 - Взаємодіє з файловою системою для збереження, відкриття чи редагування файлів.
 - У разі потреби активує управління макросами чи сніппетами.
4. Файлова система:
- Обробляє запити на збереження, завантаження чи редагування файлів від текстового редактора.
5. Змінний носій:
- У разі потреби використовується для збереження резервних копій або тимчасових файлів.
 - Передає файли через інтерфейс роботи зі змінними носіями до файлової системи ПК.

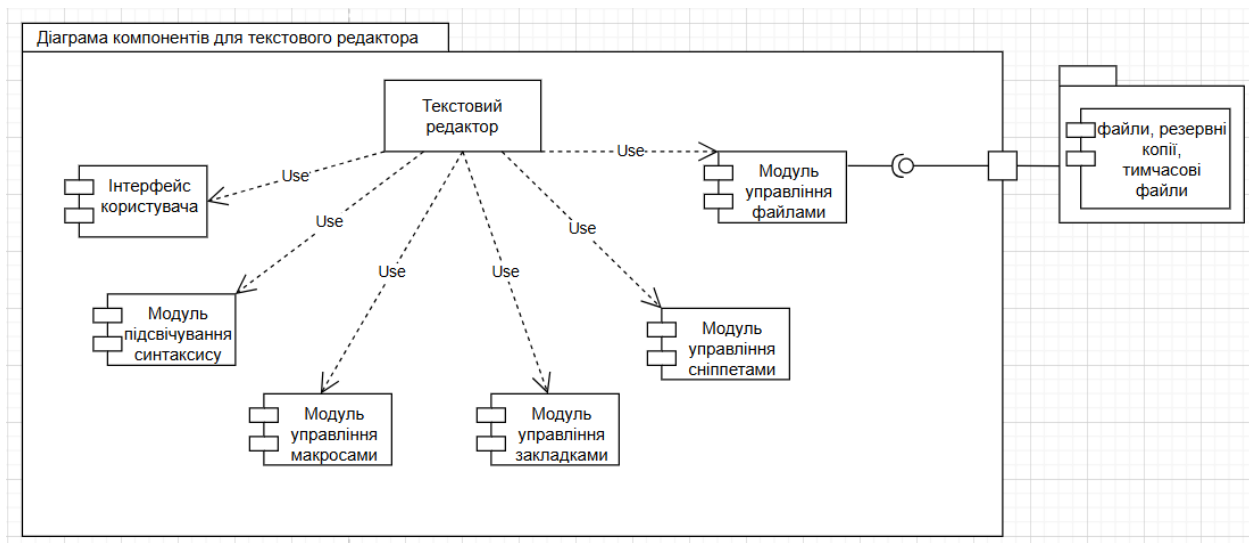


Рис 2. Діаграма компонентів

Діаграма компонентів

1. Основні компоненти:

Текстовий редактор:

- Роль: Центральний компонент, що забезпечує взаємодію з іншими модулями для виконання основних функцій.
- Функції:
 - Обробка запитів від користувача.
 - Координація роботи модулів.
 - Відображення результатів виконання операцій.

Інтерфейс користувача:

- Роль: Забезпечує користувачу можливість взаємодії з системою.
- Функції:
 - Передача запитів до текстового редактора.
 - Відображення інформації про стан системи.

Модуль підсвічування синтаксису:

- Роль: Форматує текст для зручності читання, підсвічуючи синтаксис певної мови програмування або розмітки.
- Функції:
 - Аналіз тексту.
 - Накладання стилів для підсвічування.

Модуль управління макросами:

- Роль: Автоматизує повторювані дії в редакторі.
- Функції:
 - Збереження та виконання макросів.
 - Інтеграція автоматизованих дій у процес редагування.

Модуль управління шпелетамі:

- Роль: Працює з текстовими шаблонами для їх швидкого вставлення у текст.
- Функції:
 - Створення та збереження шаблонів.
 - Інтеграція шаблонів у процес редагування.

Модуль управління закладками:

- Роль: Забезпечує можливість створення закладок для швидкого переходу до певних місць у тексті.
- Функції:
 - Додавання, видалення, редагування закладок.
 - Перехід до закладених місць у документі.

Модуль управління файлами:

- Роль: Реалізує функції роботи з файлами (відкриття, збереження, імпорт/експорт).
- Функції:
 - Робота з локальними файлами.
 - Підтримка зовнішніх носіїв.

Змінний носій:

- Роль: Зовнішній ресурс для збереження даних.
- Функції:
 - Зберігання текстових файлів, резервних копій, тимчасових документів.
 - Обмін файлами між системою та зовнішніми пристроями.

2. Взаємодії між компонентами:

- **Текстовий редактор ↔ Інтерфейс користувача:** Інтерфейс отримує запити користувача (наприклад, редагування тексту) і передає їх до текстового редактора. Текстовий редактор повертає результати операцій для відображення.
- **Текстовий редактор ↔ Модуль підсвічування синтаксису:** Передає текст для аналізу. Отримує форматовані дані для візуалізації синтаксису.
- **Текстовий редактор ↔ Модуль управління макросами:** Передає запити на виконання або запис макросів. Отримує результати виконання автоматизованих дій.
- **Текстовий редактор ↔ Модуль управління сніппетами:** Надсилає запити на додавання, редагування або вставлення шаблонів. Отримує готові сніппети для інтеграції у текст.
- **Текстовий редактор ↔ Модуль управління закладками:** Передає запити на створення, редагування чи видалення закладок. Отримує інформацію для переходу до певних місць у тексті.
- **Текстовий редактор ↔ Модуль управління файлами:** Передає запити на відкриття, збереження чи імпорт файлів. Отримує інформацію про збережені чи завантажені файли.
- **Модуль управління файлами ↔ Змінний носій:** Взаємодіє із зовнішніми джерелами для читання чи запису даних.

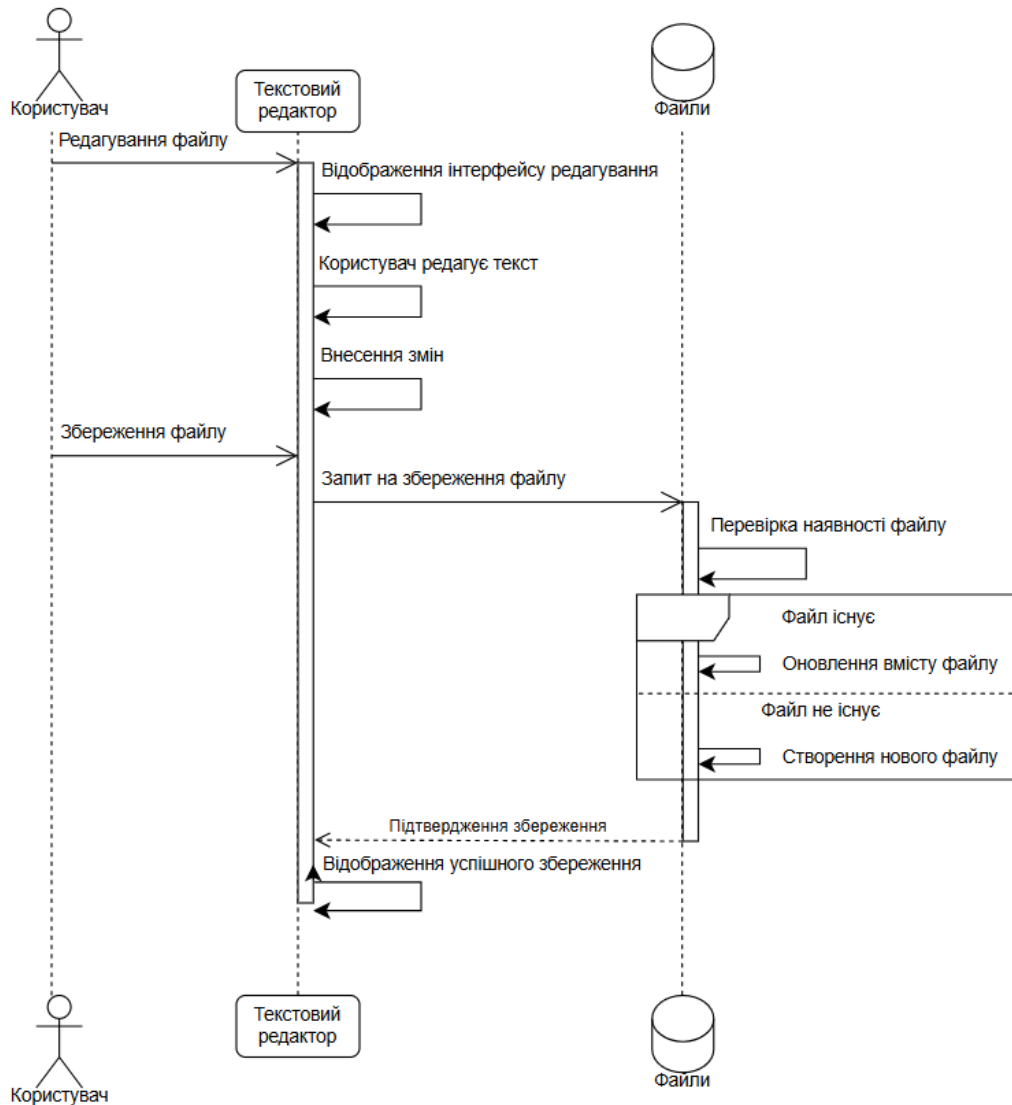


Рис 3. Діаграма послідовностей

Діаграма послідовностей

1. Користувач відкриває текстовий файл для редагування.

Користувач обирає файл для редагування через текстовий редактор. Текстовий редактор завантажує файл і відображає його в інтерфейсі редагування.

2. Відображення інтерфейсу редагування.

Текстовий редактор відображає інтерфейс редагування, у якому користувач може вносити зміни до тексту.

3. Користувач редагує текст.

Через клавіатуру користувач додає або змінює текст у документі. Текстовий редактор обробляє ці зміни та оновлює текст на екрані.

4. Користувач зберігає текстовий файл.

Користувач натискає опцію збереження у текстовому редакторі. Текстовий редактор надсилає запит на збереження файлу до файлової системи.

5. Перевірка наявності файлу у файловій системі.

Файлова система отримує запит від текстового редактора:

- Якщо файл існує, його вміст оновлюється новими даними.
- Якщо файл не існує, створюється новий файл із заданими параметрами.

6. Підтвердження успішного збереження.

Файлова система надсилає підтвердження текстовому редактору про успішне збереження.

Текстовий редактор повідомляє користувача про успішне завершення операції.

7. Завершення роботи.

Користувач може продовжити роботу з текстовим документом або закрити програму.

Додаткові особливості:

- **Перевірка файлу:**

Важливий етап, що визначає, чи слід оновлювати існуючий файл або створювати новий.

- **Умовний блок :**

Діаграма відображає логіку роботи залежно від існування файлу:

Якщо файл існує, вміст оновлюється.

Якщо файл не існує, створюється новий.

- **Інтерактивність:**

Користувач має зворотний зв'язок на кожному етапі: підтвердження успішного збереження, відображення змін тощо.

Висновок: Під час виконання лабораторної роботи я навчилася створювати діаграми розгортання, компонентів і послідовностей. Діаграма розгортання показує, як система працює на різних пристроях і як вони між собою взаємодіють. Діаграма компонентів допомагає зрозуміти, з яких частин складається програма, які завдання вони виконують і як пов'язані між собою. Діаграма послідовностей показує, у якому порядку виконуються дії користувача та як система на них реагує. Завдяки цим діаграмам стало зрозуміліше, як працює система в цілому і як її реалізувати.