



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

Лабораторна робота №3
Технології розроблення програмного забезпечення
ДІАГРАМА РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ.
ДІАГРАМА ВЗАЄМОДІЙ ТА ПОСЛІДОВНОСТЕЙ.
FTP-server

Виконала

студентка групи ІА–22

Степанюк-Б. Анастасія

Перевірив:

Мягкий Михайло Юрійович

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

Мета: Проаналізувати тему, розробити діаграму розгортання, діаграму компонентів, діаграму взаємодій та послідовностей.

Хід роботи

..26 Download manager (iterator, command, observer, template method, composite, p2p)

Інструмент для скачування файлів з інтернету по протоколах http або https з можливістю продовження завантаження в зупиненому місці, розподілу швидкостей активним завантаженням, ведення статистики завантажень, інтеграції в основні браузери (firefox, opera, internet explorer, chrome).

Я створив усі класи та інтерфейси відповідно до діаграми. Ось короткий опис кожного компонента:

- 1 PriorityLevel: визначає рівні пріоритету (HIGH, MEDIUM, LOW)
2. Class File: представляє файл з його властивостями та методом отримання прогресу завантаження
3. Class Priority: керує пріоритетом завантаження та обмеженням швидкості
4. Class Download: представляє процес завантаження з відповідними властивостями та зв'язками
5. Class User: представляє користувача системи
6. Interface IDownloadRepository: визначає методи для роботи з сховищем завантажень
7. Interface IDownloadService: визначає основні операції з завантаженнями
8. Class DownloadRepository: реалізація репозиторію завантажень
9. Class DownloadService: реалізація сервісу завантажень

Код містить всі необхідні зв'язки між класами, як показано на діаграмі:

- Зв'язок "uses" між User та IDownloadService
- Зв'язок "uses" між DownloadService та IDownloadRepository

- Зв'язок "has" між Download та Priority
- Зв'язок "contains" між Download та File

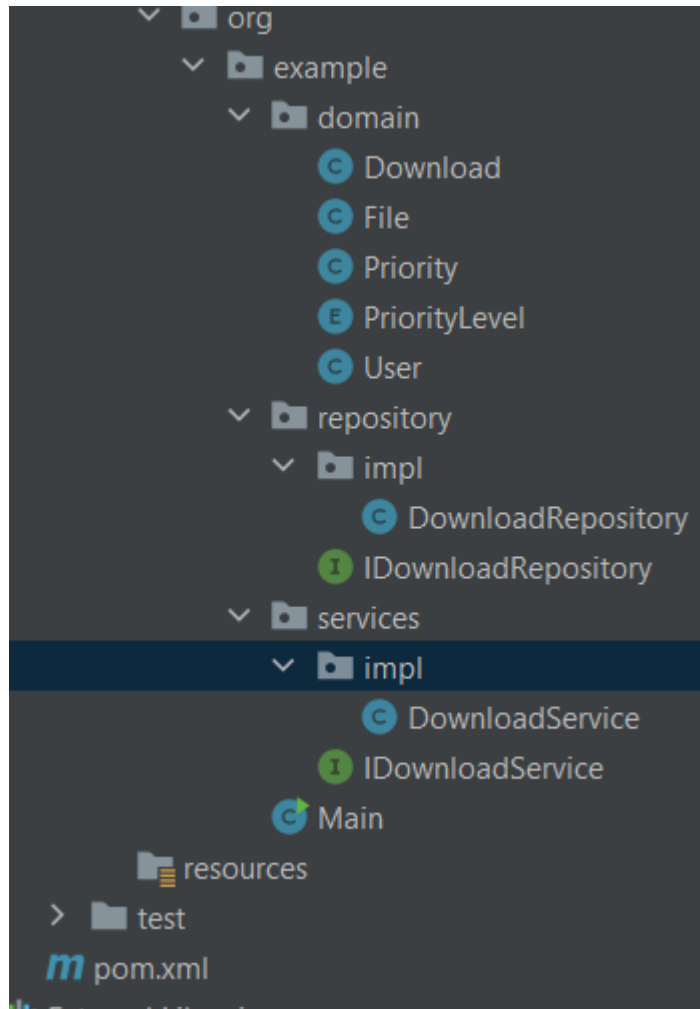


Рисунок 1 – Структура проекту

1. Спроектувати діаграму послідовностей для одного із процесів розроблюваної системи

Діаграма послідовностей для процесу завантаження файлу на сервер зображена на рисунку 2.

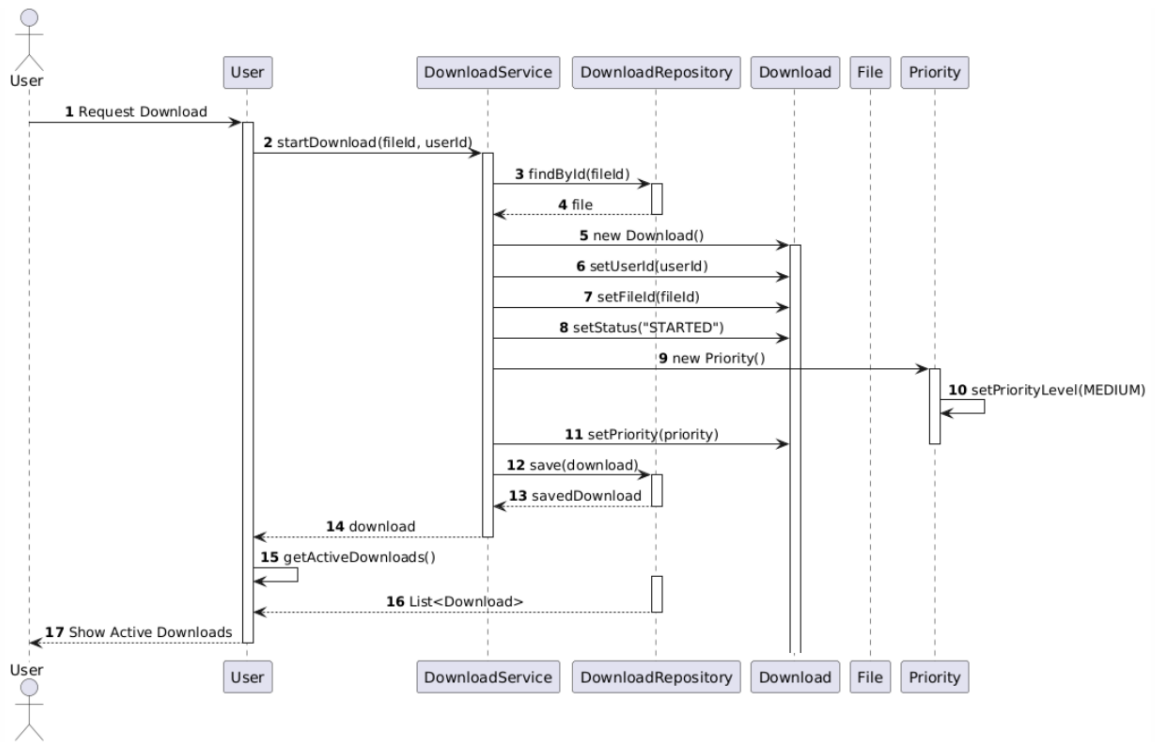


Рисунок 2 – Діаграма послідовностей

Діаграма послідовностей — це тип UML-діаграм, що ілюструє взаємодію між об'єктами в системі через послідовність повідомлень. Вона складається з таких основних компонентів:

Теорія:

1. Об'єкти: Відображаються у верхній частині діаграми.
2. Лінії життя: Вертикальні пунктирні лінії, що показують життєвий цикл об'єкта.
3. Повідомлення: Стрілки між об'єктами, які демонструють обмін даними.

Основні принципи:

- Часова послідовність: Показує порядок взаємодій.
- Взаємодії між об'єктами: Допомогає виявити проблеми в архітектурі системи.

Використання:

- Проектування: Допомогає розробникам зрозуміти, як компоненти взаємодіють.

- Документація: Служить для пояснення поведінки системи.

Діаграми послідовностей є корисним інструментом для візуалізації динаміки системи в розробці програмного забезпечення.

2. Спроекувати діаграму розгортання для розроблюваної системи.

Діаграма розгортання системи зображена на рисунку 3.

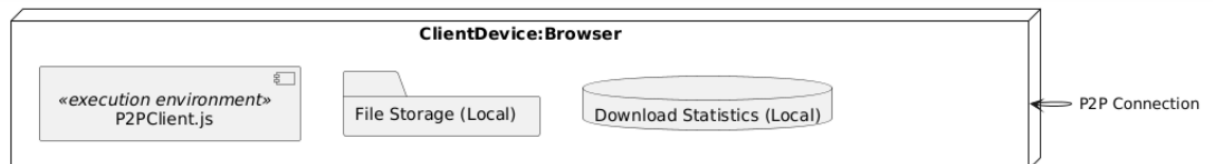


Рисунок 3 – Діаграма розгортання

Теорія:

Діаграма розгортання (Deployment Diagram) — це тип UML-діаграм, який відображає фізичну архітектуру системи, включаючи її апаратні та програмні компоненти, а також їх взаємозв'язки. Основна мета діаграми — показати, як програмне забезпечення буде розгорнуто на апаратних засобах.

Основні компоненти діаграми розгортання:

1. Ноди (Nodes): Відображають фізичні пристрої або компоненти системи, такі як сервери, комп'ютери, мережеве обладнання.
2. Артефакти (Artifacts): Представляють програмні компоненти, які розгортаються на нодах, наприклад, виконувані файли, бібліотеки або бази даних.
3. Зв'язки (Connections): Лінії, що показують зв'язки між нодами, наприклад, мережеві з'єднання.

Основні принципи:

- Фізична архітектура: Відображає, як програмне забезпечення буде працювати в реальному світі.
- Розгортання системи: Показує, де і як компоненти системи будуть розміщені.

Використання:

- Проектування: Допомогає архітекторам і розробникам планувати, як розгорнути систему.
- Документація: Використовується для документування архітектури системи для команди та замовників.

Діаграми розгортання є важливими для зрозуміння фізичних аспектів системи і забезпечують чітке уявлення про її структуру.

3. Спроекувати діаграму компонентів для розроблюваної системи.

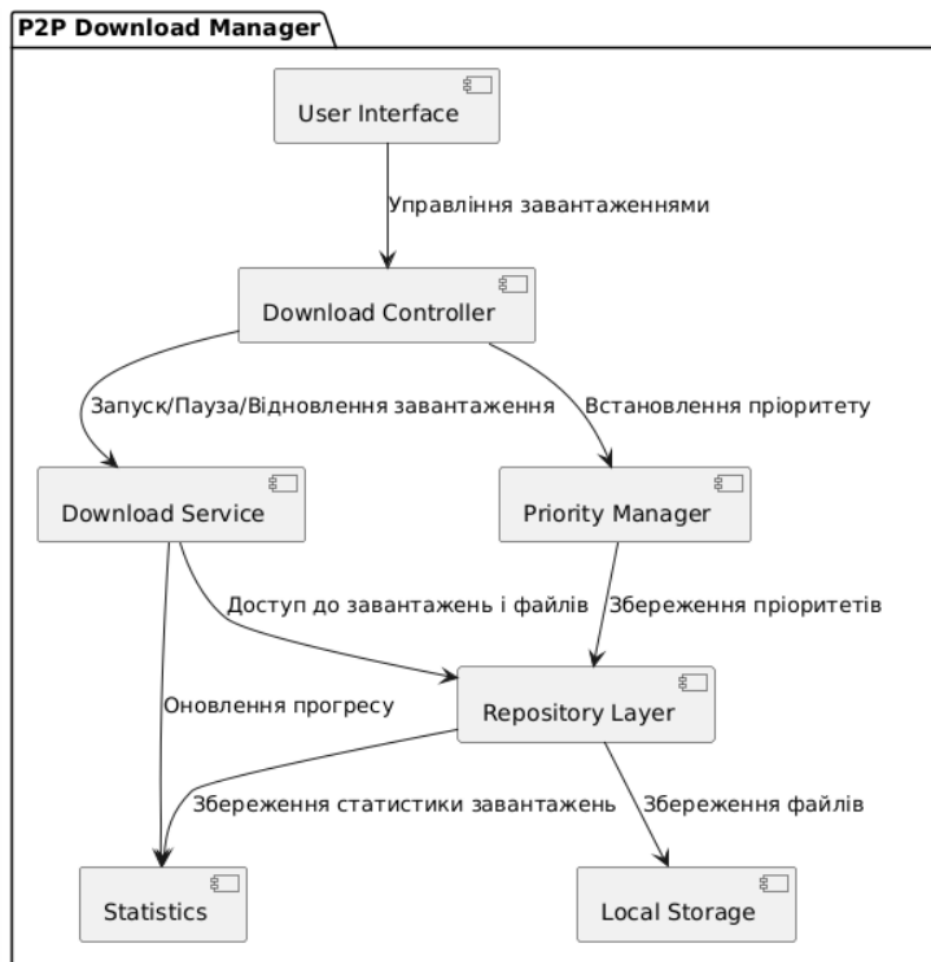


Рисунок 4 – Діаграма компонентів

Теорія:

Діаграма компонентів (Component Diagram) — це тип UML-діаграм, який відображає архітектуру програмного забезпечення, зокрема, компоненти

системи та їхні взаємозв'язки. Вона показує, як різні частини системи інтегруються один з одним.

Основні компоненти діаграми компонентів:

1. Компоненти: Основні елементи, які представляють частини програмного забезпечення, такі як модулі, бібліотеки або сервіси.
2. Інтерфейси: Визначають способи взаємодії між компонентами. Інтерфейси можуть бути представлені як "порти" на компонентах.
3. Зв'язки: Стрілки, які показують, як компоненти взаємодіють один з одним.

Основні принципи:

- Модульність: Діаграма демонструє, як система поділена на модулі, що полегшує її управління та розширення.
- Взаємозв'язки: Показує, які компоненти залежать один від одного, що допомагає в ідентифікації точок інтеграції.

Використання:

- Проектування: Допомагає в архітектурному плануванні та розробці системи.
- Документація: Служить для опису структури програми, полегшуючи спілкування між розробниками.

Діаграми компонентів є корисним інструментом для візуалізації структури програмного забезпечення і спрощення процесу його розробки.

Висновок: під час виконання лабораторної роботи я проаналізувала тему, розробив діаграму розгортання, діаграму компонентів, діаграму взаємодій та послідовностей.