



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 №2

із дисципліни «*ТПІЗ*»

**Тема: «ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ
ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ. КОНЦЕПТУАЛЬНА
МОДЕЛЬ СИСТЕМИ»**

Виконав:
Студент групи ІА-24
Бакалець А.І.

Перевірив:
Мякий М.Ю.

Мета: Вивчення основних принципів побудови та проектування програмних систем з використанням схем прецедентів, діаграм класів та структури бази даних.

Теоретичні відомості

Діаграма прецедентів (Use Case Diagram)

Діаграма прецедентів використовується для візуалізації функціональних вимог до системи. Вона ілюструє, як користувачі (актори) взаємодіють із системою через конкретні сценарії використання (прецеденти). Основними елементами цієї діаграми є актори, прецеденти та зв'язки між ними. Прецеденти дозволяють виявити основні функції системи та забезпечити їхнє розуміння на високому рівні.

Діаграма класів (Class Diagram)

Діаграма класів моделює статичну структуру системи. Вона демонструє класи, їх атрибути, методи (операції) та зв'язки між класами, такі як асоціації, агрегації, композиції та успадкування. Класи представляють основні компоненти системи, їхні характеристики (атрибути) та поведінку (методи), а зв'язки між ними показують, як ці класи взаємодіють.

База даних та її структура

База даних — це організований набір інформації, що зберігається у структурованому вигляді, зазвичай у формі таблиць. Таблиці складаються з рядків (записів) і стовпців (полів), де поля містять атрибути даних. Структура бази даних визначає, як дані пов'язані між собою. Основні елементи бази даних включають таблиці, ключі (первинні та зовнішні), а також зв'язки між таблицями, які можуть бути одного з типів: один-до-одного, один-до-багатьох або багато-до-багатьох.

Шаблон Репозиторію (Repository Pattern)

Шаблон Репозиторію використовується для абстрагування доступу до даних. Він дозволяє працювати з базою даних через клас-репозиторій, що інкапсулює операції зі збереження, отримання, оновлення та видалення даних. Це знижує залежність між бізнес-логікою та логікою доступу до даних, роблячи систему більш гнучкою до змін або оновлень.

Завдання.

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.
3. Намалюйте діаграму класів для реалізованої частини системи.
4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.
5. Розробити основні класи і структуру системи баз даних.
6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.
7. Підготувати звіт про хід виконання лабораторних робіт. Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

Тема: MP3Player

Схема прецеденту:



Прецеденти використання:

Додати новий плейліст

Назва	Додати новий плейліст
Передумови	Користувач ввів відповідну команду та має доступ до сервера
Постумови	Новий плейліст додано до бази даних
Актори	Користувач, сервер
Опис	Користувач вводить відповідну команду у командному рядку клієнта.

Основний хід подій	Сервер отримує команду. Створює новий плейлист і зберігає його в базі даних
Винятки та примітки	Якщо плейлист з таким ім'ям вже існує, сервер повідомляє користувача, що плейлист із вказаною назвою вже існує.

Додати новий трек в плейліст

Назва	Додавання треку до плейлиста
Передумови	Користувач має доступ до серверу та вибрав плейлист, та ввів відповідну команду.
Післяумови	Новий трек додано до обраного плейлисту в базі даних.
Сторони, що взаємодіють	Користувач, сервер
Опис	Користувач вводить команду для додавання треку.
Основний потік подій	Сервер отримує команду, перевіряє наявність плейлисту та треку, додає трек до плейлисту та зберігає в базі даних.
Винятки та примітки	Якщо трек з такою назвою вже існує в обраному плейлісті, сервер повідомляє користувача, що трек вже існує та ігнорує команду

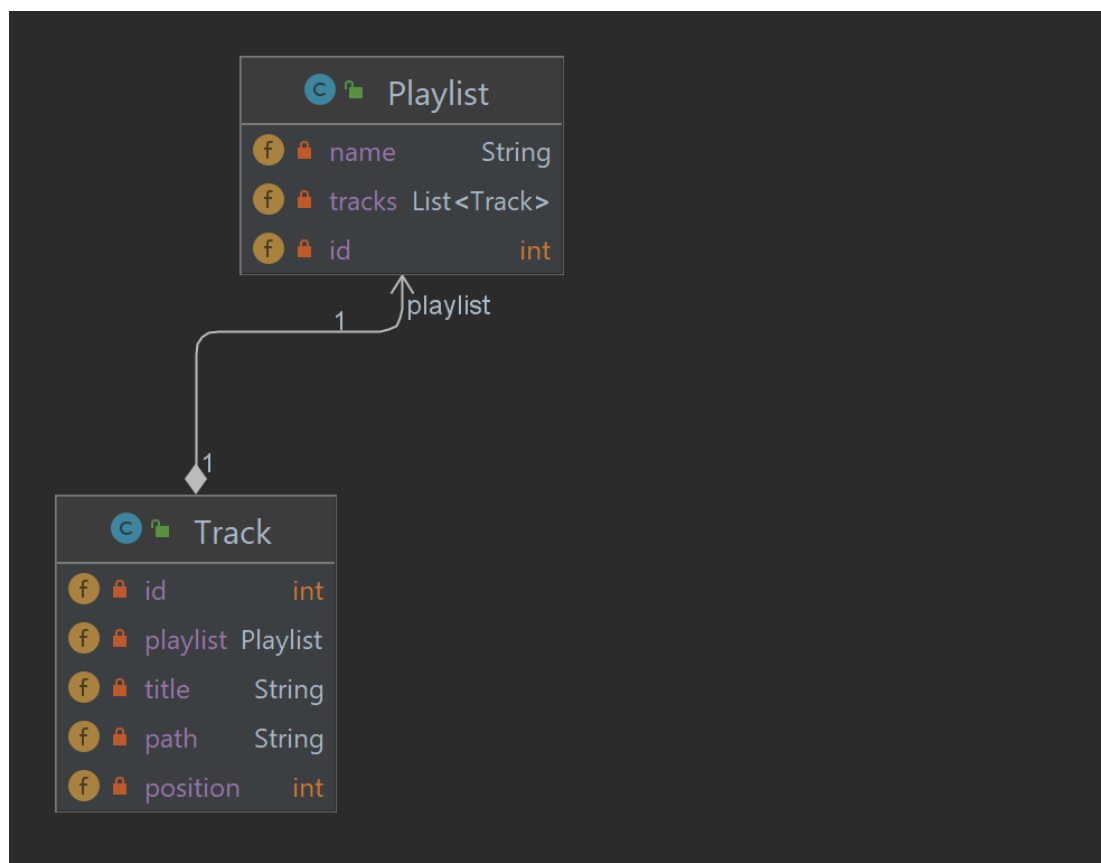
Відтворення плейлисту з використанням моменту

Назва	Відтворення плейлисту з використанням моменту
-------	---

Передумови	Користувач увійшов до системи, має доступ до серверу та вибрав активний плейлист.
Післяумови	Відтворення плейлисту розпочато, або відновлено попередній стан.

Сторони, що взаємодіють	Користувач, сервер
Опис	Користувач вводить відповідну команду
Основний потік подій	Сервер отримує команду розпочинає відтворення треків у потоці та надсилає відповідь користувачеві.
Винятки та примітки	Якщо обраний плейлист порожній, сервер повідомляє користувача, що плейлист порожній та ігнорує команду.

Діаграма класів:



Track:

Id – Індивідуальний номер кожного треку, **PK**

Playlist – Плейлист до якого належить, **FK**

Title – Назва композиції

Path – Шлях до композиції

Position – Розташування треку в плейлисті

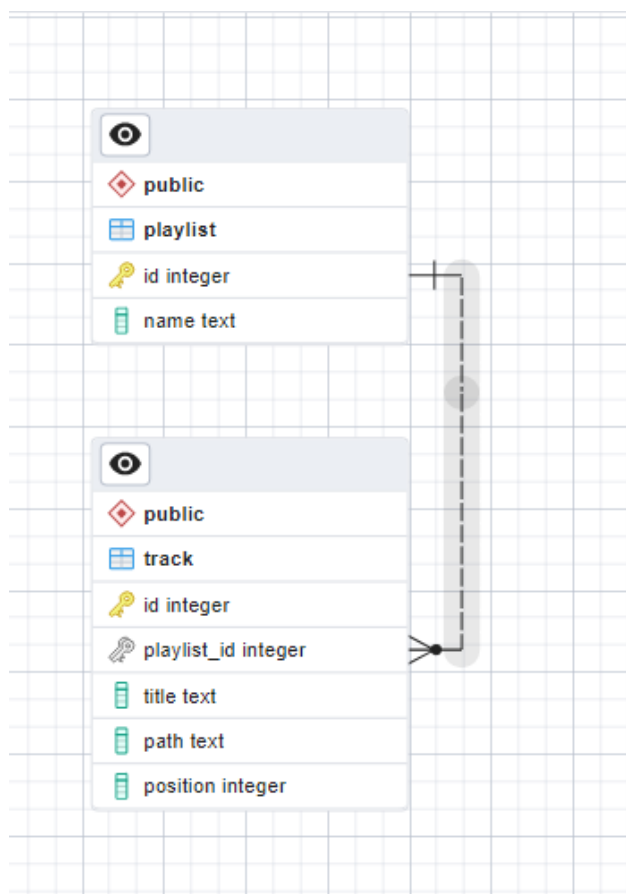
Playlist:

Id – Індивідуальний номер кожного плейлисту, **PK**

Name – Назва плейлисту

Tracks – всі треки які належать цьому плейлисту

Структура бази даних:



Висновок: виконуючи дану лабораторну роботу, я опрацював діаграму прецендетів, детально описав три з них та описав діаграму класів. Також мною було додано моделі та репозиторії до мого проєкту, і розроблена БД, з використанням Spring Boot.