



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 №2
Технологія розробки програмного забезпечення
«Основи проектування.»

Виконала
студентка групи ІА–32:
Ткачук М. С.

Перевірив:
Мягкий М. Ю.

Київ 2025

Зміст

Теоратичні відомості	4
Діаграма прецендентів.....	5
Прецендент 1	6
Прецендент 2	7
Прецендент 3	8
Діаграма класів.....	9
1. Repository Pattern.....	9
2. Моделі (предметна область аудіоредактора)	9
3. Зв'язки між класами.....	10
4. Репозиторії.....	10
5. Utility-класи	11
6. База даних та з'єднання.....	11
Структура бази даних	11
Висновок	12

Тема: Основи проектування

Мета: Обрати зручну систему побудови UML-діаграм та навчитися будувати діаграми варіантів використання для системи що проектується, розробляти сценарії варіантів використання та будувати діаграми класів предметної області

Завдання:

Ознайомитись з короткими теоретичними відомостями.

- Проаналізувати тему та спроектувати діаграму варіантів використання відповідно до обраної теми лабораторного циклу.
- Спроектувати діаграму класів предметної області.
- Вибрати 3 варіанти використання та написати за ними сценарії використання.
- На основі спроектованої діаграми класів предметної області розробити основні класи та структуру бази даних системи. Класи даних повинні реалізувати шаблон Repository для взаємодії з базою даних.
- Нарисувати діаграму класів для реалізованої частини системи.
- Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму варіантів використання відповідно, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних

Аудіо редактор (singleton, adapter, observer, mediator, composite, client-server)

Аудіо редактор повинен володіти наступним функціоналом: представлення аудіо даних будь-якого формату в WAVE-формі, вибір і подальші операції копіювання / вставки / вирізання / деформації по сегменту аудіозапису, можливість роботи з декількома звуковими доріжками, кодування в найбільш поширених форматах (ogg, flac, mp3).

Теоретичні відомості

UML (Unified Modeling Language) — універсальна мова візуального моделювання для специфікації, візуалізації, проектування й документування програмних систем та бізнес-процесів.

UML дозволяє створювати концептуальні, логічні та фізичні моделі складних систем на різних рівнях абстракції.

В ООАП (об'єктно-орієнтований аналіз і проектування) модель системи складається з різних уявлень, які відображають її поведінку або структуру.

Діаграма варіантів використання (Use Case Diagram) – це тип діаграми UML, що описує функціональність системи з точки зору її користувачів (акторів) і взаємодії між ними та системою. Вона показує, які дії (варіанти використання) можуть виконуватися користувачами, але не вдається у внутрішні механізми їх реалізації.

Сценарії варіантів використання (Use Case Scenarios) – це текстовий опис варіантів використання, де детально викладається, як система повинна реагувати на дії користувачів у кожній конкретній ситуації. Включає в себе основний потік подій та альтернативні шляхи розвитку сценарію.

Діаграма класів (Class Diagram) – це структура, яка моделює класи системи, їх властивості, методи, а також зв'язки між ними. Класи представляють основні об'єкти системи, які мають атрибути та операції, а також відображають взаємодію між різними компонентами.

Концептуальна модель системи – це абстрактне представлення об'єктів та зв'язків між ними, що відображає ключові аспекти системи з точки зору бізнесу або 4 предметної області. Вона описує основні компоненти, їх взаємодію та структуру, але не деталізує технічну реалізацію. Ці діаграми дозволяють аналізувати вимоги до системи та планувати її розробку.

Хід Роботи

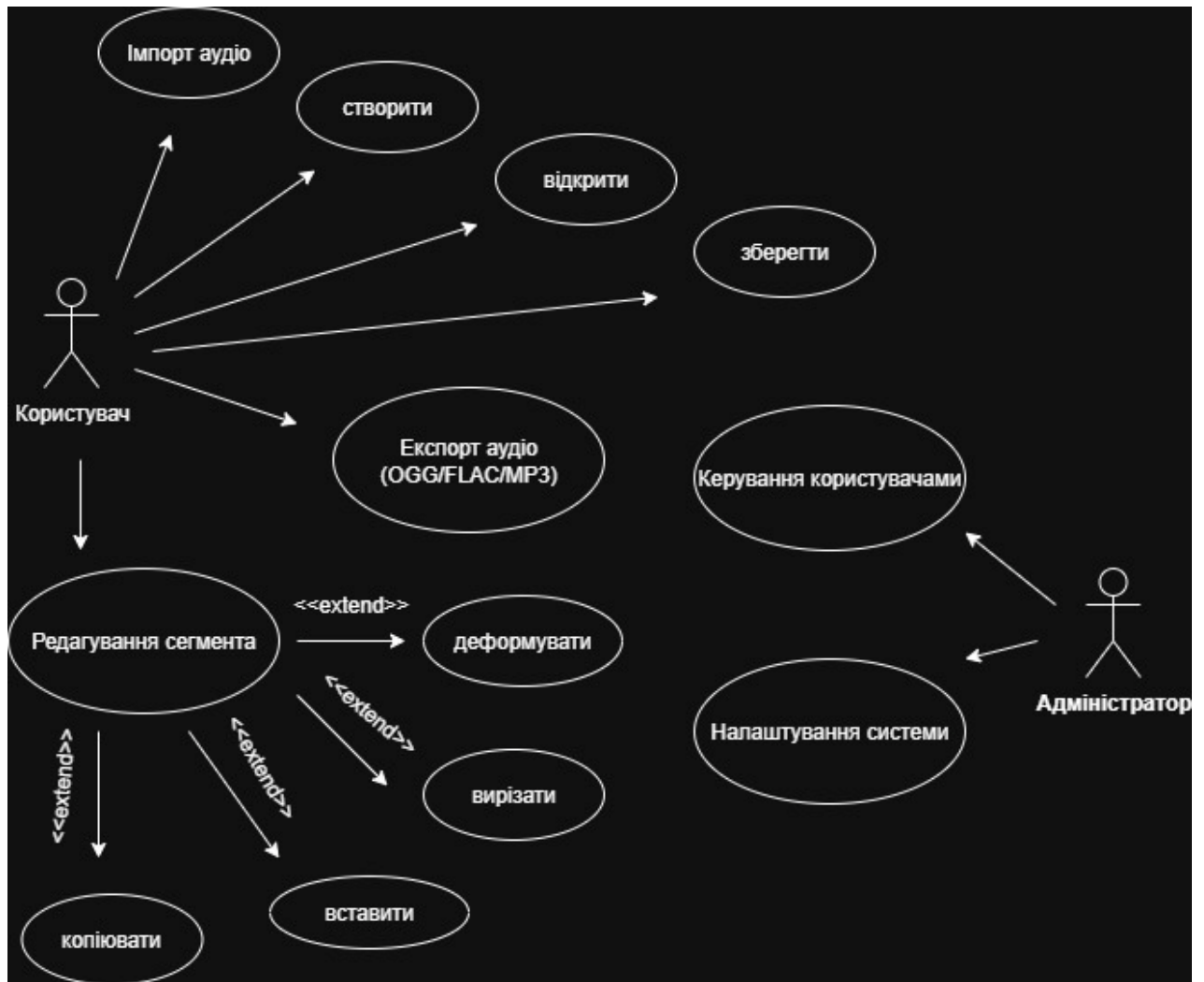


Рис 1. Діаграма прецендентів

Користувач запускає аудіоредактор та може:

- Імпортувати аудіо
- Редагувати сегмент (копіювати, вставляти, вирізати, деформувати)
- Експортувати аудіо (OGG/FLAC/MP3)
- створити/відкрити/зберегти проєкт;

Адміністратор керує користувачами й налаштуваннями

Прецедент 1: Імпорт аудіо

- Передумови: користувач має локальний аудіофайл; редактор запущений; відкритий проєкт (новий або існуючий)
- Постумови: файл додано до проєкту як доріжка/кліп; на таймлайн доступне WAVE-подання. У разі помилки - показано повідомлення, імпорт не виконується
- Сторони взаємодії: користувач, система аудіоредактора
- Короткий опис: завантаження аудіофайлу до проєкту з візуалізацією у вигляді хвильової форми

Основний потік подій:

1. Користувач обирає дію «Імпорт аудіо»
2. Система відкриває діалог вибору файлу
3. Користувач обирає файл(и) і підтверджує
4. Система додає файл у проєкт, генерує WAVE-подання та показує його на таймлайн
5. Система повідомляє про успішний імпорт.

Винятки:

- Виняток №1: непідтримуваний формат/пошкоджений файл - система показує помилку та пропонує вибрати інший файл
- Виняток №2: нестача ресурсів (пам'яті/місця) - система зупиняє імпорт і пропонує звільнити місце або зменшити обсяг
- Виняток №3: відмова у доступі до файлу - система просить надати доступ або вибрати інше розташування

Прецедент 2: Редагування сегмента

- Передумови: у проєкті є хоча б одна доріжка з аудіо; користувач бачить WAVE-подання на таймлайні
- Постумови: у проєкті збережено зміни (нові або змінені кліпи/ділянки). У разі помилки змін не застосовано
- Сторони взаємодії: користувач, система аудіоредактора
- Короткий опис: користувач виділяє ділянку сигналу та виконує над нею операції редагування

Основний потік подій:

1. Користувач обирає інструмент виділення та задає межі сегмента на таймлайні
2. Користувач виконує одну або кілька операцій: копіювати, вставити, вирізати, деформувати (зміна швидкості/довжини)
3. Система оновлює відображення таймлайна (кліпи/доріжки) та програвє короткий прев'ю за бажанням
4. Користувач зберігає зміни у проєкті

Винятки:

- Виняток №1: виділення поза межами аудіо/порожній сегмент - система просить скоригувати межі
- Виняток №2: конфлікт вставки (накладання поверх заблокованого/захищеного фрагмента) - система пропонує обрати інше місце або підтвердити заміну
- Виняток №3: деформація призводить до перевищення допустимої довжини або артефактів система попереджає та пропонує інші параметри

Прецедент 3: Експорт аудіо (OGG/FLAC/MP3)

- Передумови: у проєкті є доріжки/кліпи; користувач завершив редагування та готовий зберегти результат у файл
- Постумови: створено вихідний аудіофайл у вибраному форматі; система повідомляє про успіх або про помилку (файл не створено)
- Сторони взаємодії: користувач, система аудіоредактора
- Короткий опис: зведення доріжок проєкту в один аудіофайл і кодування у вибраний формат

Основний потік подій:

1. Користувач обирає «Експорт аудіо (OGG/FLAC/MP3)»
2. Система пропонує вибрати формат і параметри та шлях збереження
3. Користувач підтверджує експорт
4. Система виконує зведення доріжок та кодування у вибраний формат
5. Після завершення система повідомляє про успішний експорт і показує шлях до файлу

Винятки:

- Виняток №1: немає вільного місця/відсутні права на запис - система просить вибрати іншу теку або звільнити місце
- Виняток №2: конфлікт параметрів (некоректні значення бітрейту/частоти) - система пропонує стандартні/рекомендовані налаштування
- Виняток №3: збій під час кодування - система зупиняє експорт, зберігає лог і пропонує повторити з іншими параметрами

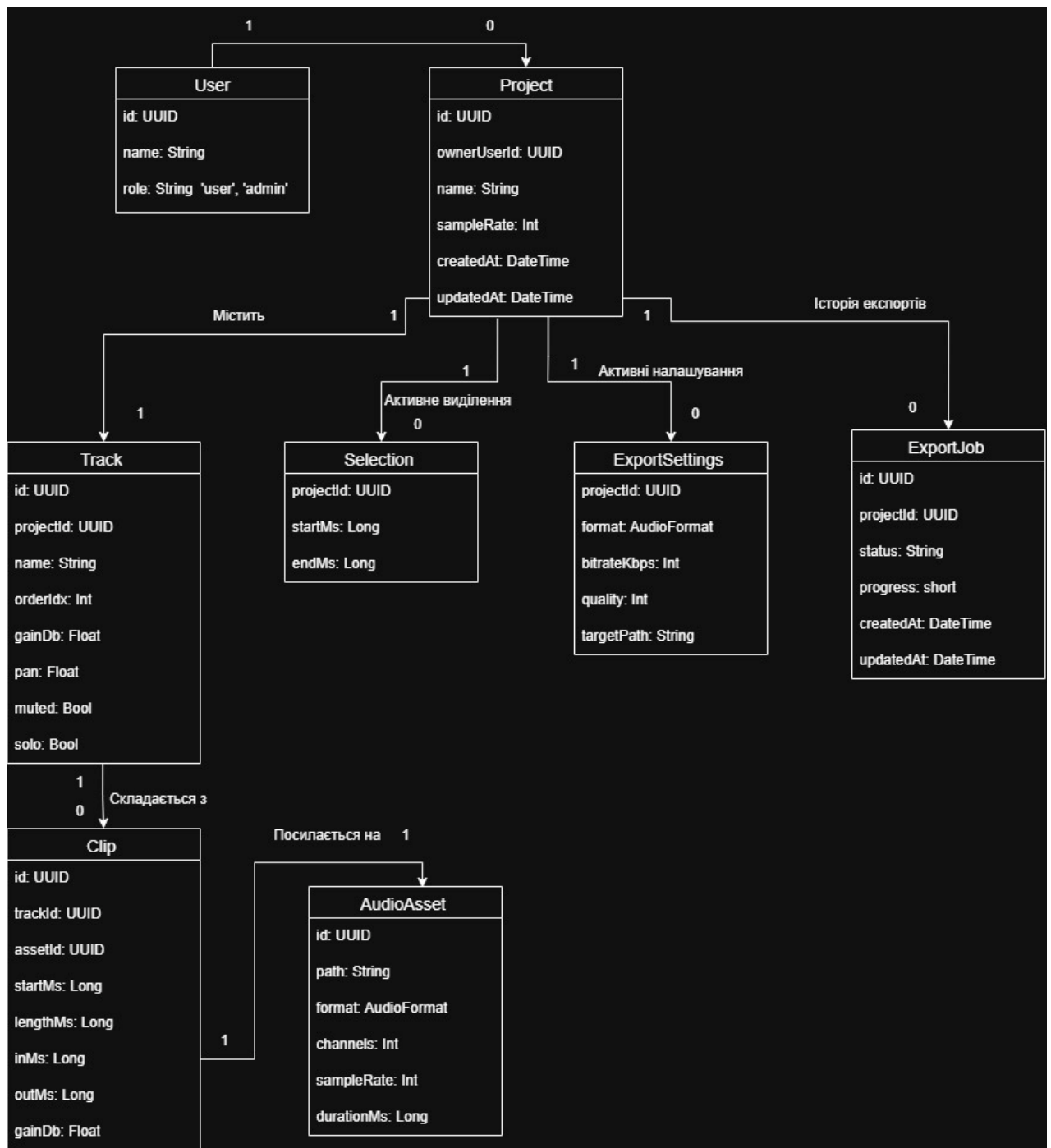


Рис 2. Діаграма класів

Repository Pattern

- Repository<T>: save(T t), findById(UUID id), delete(UUID id), findAll().

Моделі (предметна область аудіоредактора)

- Project { id, ownerUserId?, name, sampleRate, createdAt, updatedAt }
- Track { id, projectId, name, orderIdx, gainDb, pan, muted, solo }

- AudioAsset { id, path, format, channels, sampleRate, durationMs }
- Clip { id, trackId, assetId, startMs, lengthMs, inMs, outMs?, gainDb }
- Selection { projectId, startMs, endMs } (активне виділення у поточному проєкті)
- ExportSettings { projectId, format, bitrateKbps?, quality?, targetPath }
- ExportJob { id, projectId, status, progress, createdAt, updatedAt } (для прогресу експорту)
- User { id, name, role } (роль: user/admin)

Репозиторії

ProjectRepository, TrackRepository, AssetRepository, ClipRepository, SelectionRepository (get/upsert/clear), ExportSettingsRepository (get/upsert/delete), ExportJobRepository (updateStatus, findByProject)
усі побудовані на базовому Repository<T> де доречно.

Зв'язки між класами

- User 1 — 0..* Project: користувач може мати багато проєктів (власник — опційно).
- Project 1 — 1..* Track: проєкт містить одну або більше доріжок.
- Track 1 — 0..* Clip (композиція): кліпи існують лише в межах своєї доріжки.
- Clip 1 — 1 AudioAsset: кожен кліп посилається на один аудіоресурс.
- Project 1 — 0..1 Selection: у проєкті може бути одне активне виділення.
- Project 1 — 0..1 ExportSettings: один набір активних налаштувань експорту на проєкт.
- Project 1 — 0..* ExportJob: історія/черга задач експорту.

Utility-класи

- AudioFileUtils: fileExists(path), copyFile(src,dst), deleteFile(path), getFileSize(path)
- WaveformUtils: generatePeaks(assetId) (генерація WAVE-піків/даунсемпл)
- ChecksumUtils: calculateChecksum(path)
- MixdownUtils: mixTracks(projectId) (зведення PCM перед експортом)

DatabaseConnection - це окремий клас, який відповідає за роботу з БД. Він ізолює логіку створення і закриття з'єднань від бізнес-логіки та репозиторіїв.

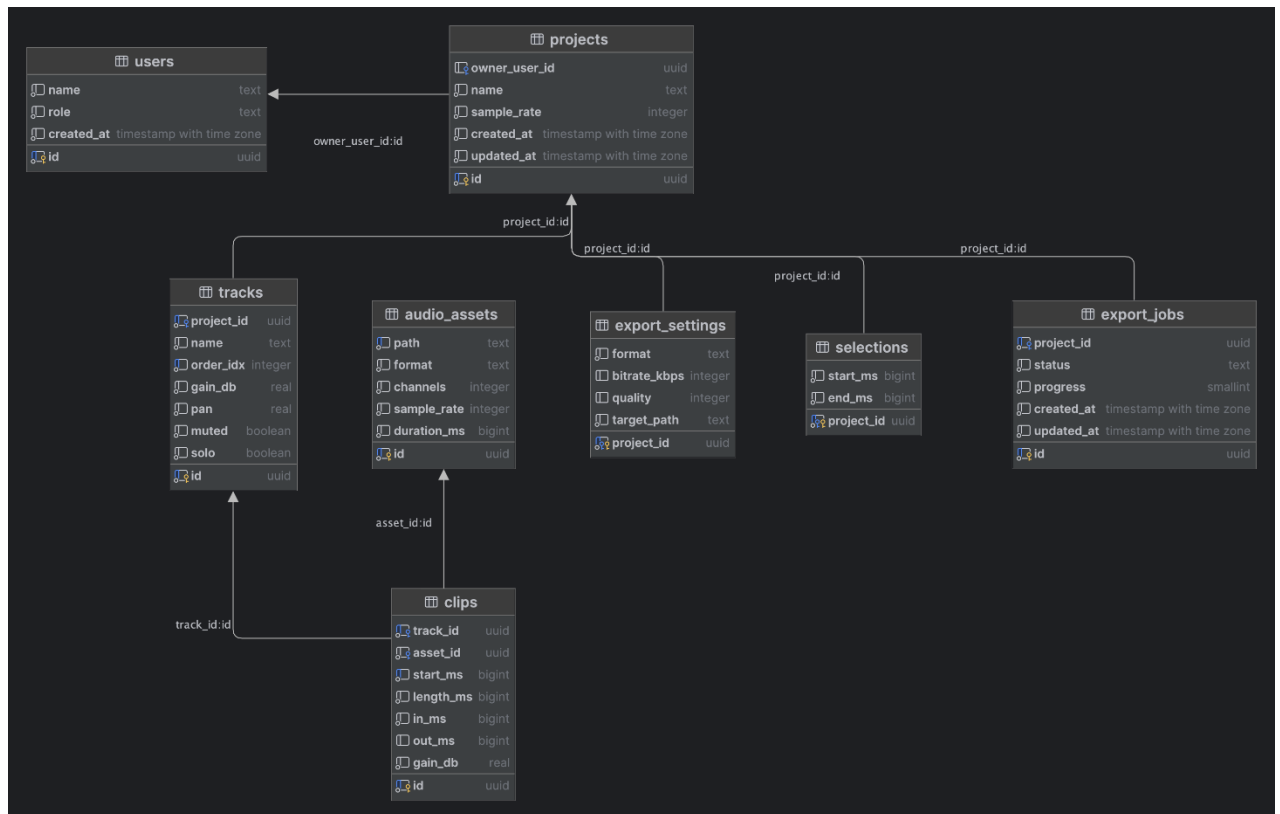


Рис 3. Структура бази даних

Висновок: У результаті виконання завдання було спроектовано діаграму класів предметної області аудіоредактора. Діаграма відображає основні сутності системи: користувачів, проекти, доріжки, кліпи, аудіоресурси, виділення, параметри експорту та історію експортів. Між класами визначено зв'язки та кардинальності, що дозволяє чітко зрозуміти структуру системи та взаємодію її компонентів. Розроблена модель є узгодженою з вимогами системи й може слугувати основою для побудови структури бази даних та подальшої реалізації репозиторіїв відповідно до шаблону Repository.