



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 № 7**

з дисципліни: «Технології розроблення програмного забезпечення»

Виконав:

Тимчук Владислав

ІА-34

Перевірив:

Мягкий М. Ю.

**Тема:** Патерни проектування.

**Мета:** Вивчити структуру шаблонів «Mediator», «Facade», «Bridge», «Template method» та навчитися застосовувати їх в реалізації програмної системи.

**Тема Лабораторного Практикуму:**

**Музичний програвач** (iterator, command, memento, facade, visitor, clientserver)

Музичний програвач становить собою програму для програвання музичних файлів або відтворення потокової музики з можливістю створення, запам'ятовування і редагування списків програвання, перемішування/повторення (shuffle/repeat), розпізнавання різних аудіоформатів, еквалайзер.

## Вступ

Метою цієї лабораторної роботи є реалізація частини функціональності програмного забезпечення «Музичний програвач» із використанням шаблону проектування Facade (Фасад). У реальній системі музичного програвача існує кілька підсистем: управління плейлистами, бібліотекою треків, пошук музики, відтворення аудіо тощо. Клієнтський інтерфейс (UI) не повинен знати всі деталі взаємодії з цими підсистемами, тому доцільно надати єдиний спрощений інтерфейс — фасад.

## Хід роботи

### 1 Опис реалізованої функціональності

Реалізована функціональність моделює простий сценарій: клієнт (умовний UI) хоче:

- створити плейлист;
- додати в нього трек;
- знайти треки за назвою;
- відтворити вибраний трек.

Замість того, щоб самому звертатися до PlaylistService, LibraryService та AudioEngine, клієнт використовує один клас — AudioPlayerFacade, який:

- всередині викликає відповідні методи сервісів;
- приховує від клієнта деталі реалізації та структуру підсистеми;
- спрощує код клієнта й зменшує зв'язність.

## 2 Діаграма класів

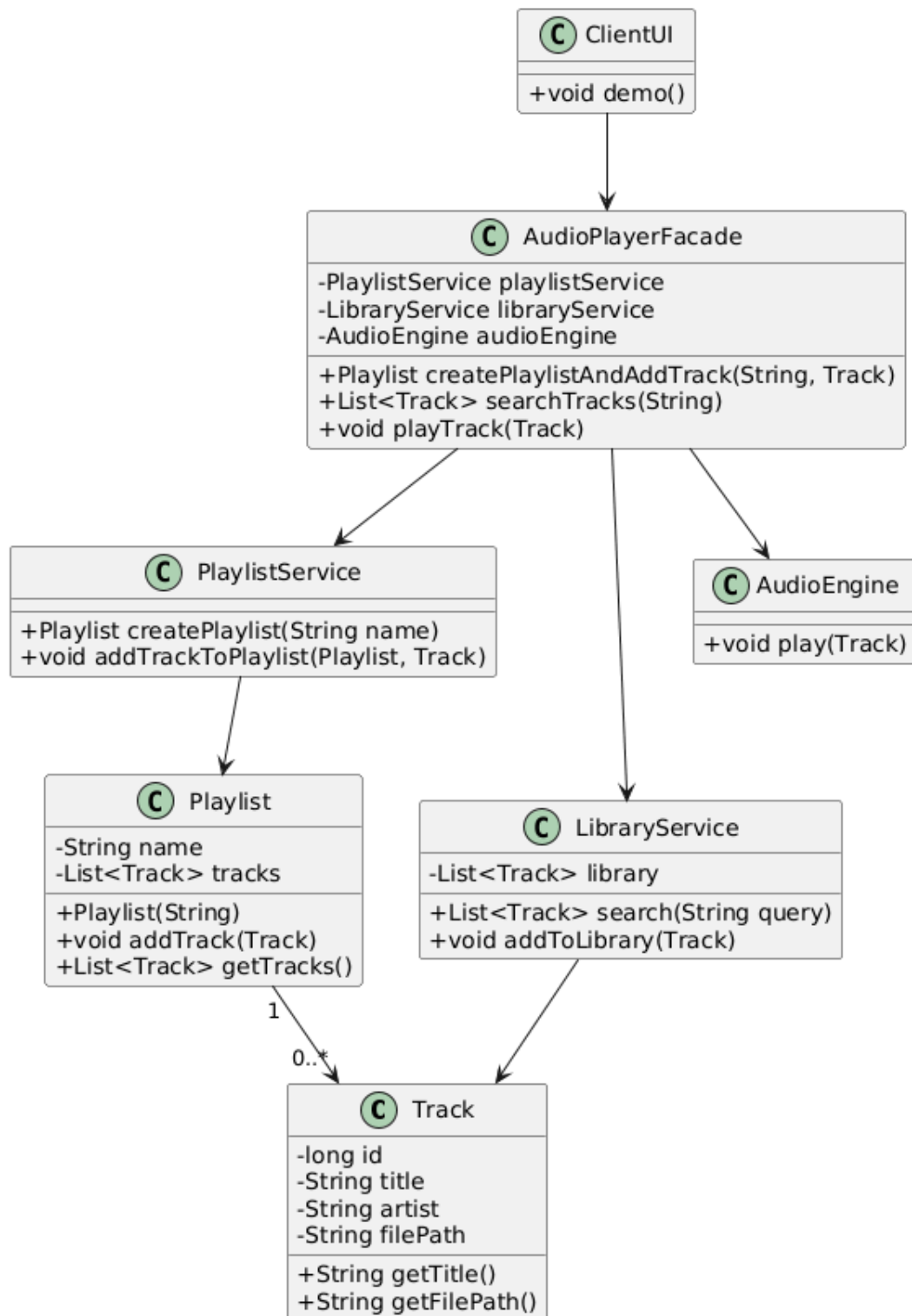


Рис. 1 – Діаграма класів

## 3 Фрагменти коду реалізації шаблону Facade

Увесь код можна знайти на віддаленому репозиторії за посиланням:  
<https://github.com/fromz67/TRPZ/tree/main/lab7>

## ВИСНОВОК

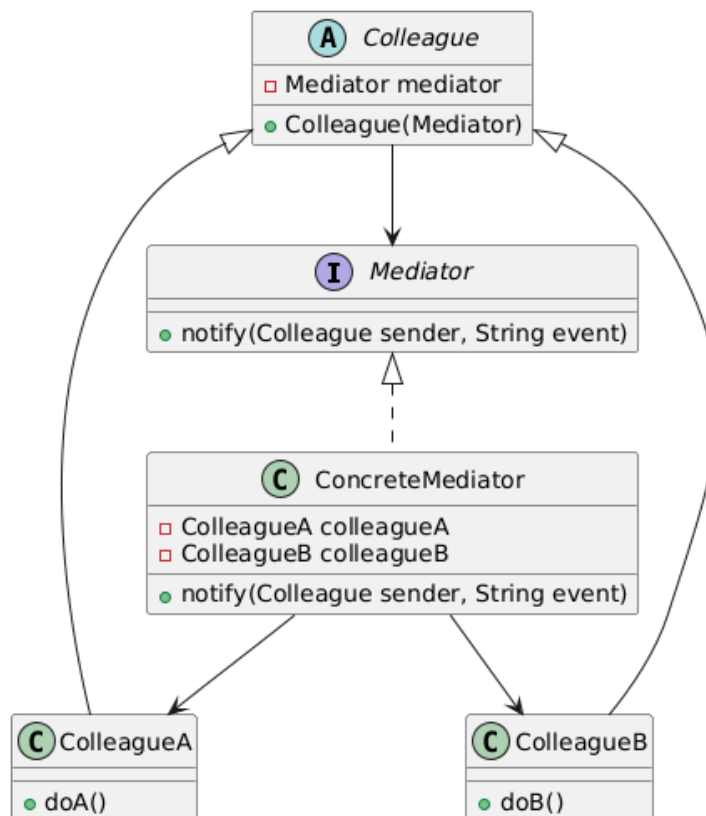
Під час виконання лабораторної роботи було реалізовано частину функціональності музичного програвача з використанням шаблону проектування Facade. Використання фасаду зменшує зв'язність, спрощує клієнтський код, полегшує подальшу підтримку й розширення системи.

## КОНТРОЛЬНІ ЗАПИТАННЯ

### 1. Яке призначення шаблону «Посередник»?

Шаблон «Посередник» призначений для організації взаємодії між об'єктами через один центральний об'єкт-посередник, щоб зменшити кількість прямих зв'язків між компонентами й уникнути «павутини» залежностей.

### 2. Нарисуйте структуру шаблону «Посередник».



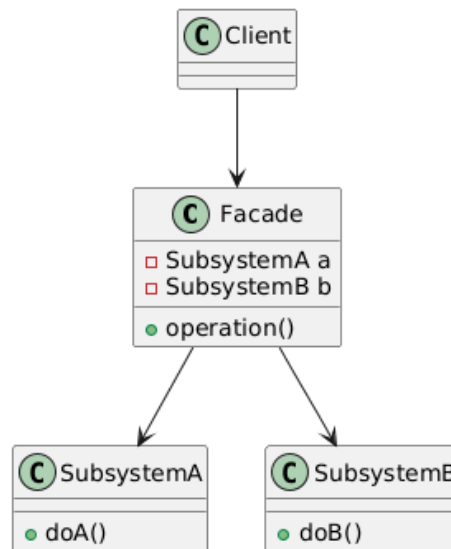
### 3. Які класи входять в шаблон «Посередник», та яка між ними взаємодія?

У шаблон входять: **Mediator** (інтерфейс посередника), **ConcreteMediator** (реалізує координацію між об'єктами), абстрактний **Colleague** (базовий клас учасників) та **ConcreteColleague** (конкретні учасники); колеги не спілкуються напряму, вони повідомляють посередника про події, а той вирішує, кого і як сповістити.

4. **Яке призначення шаблону «Фасад»?**

«Фасад» надає спрощений єдиний інтерфейс до складної підсистеми, приховуючи її внутрішню структуру та деталі реалізації, щоб клієнт працював з одним «вхідним пунктом», а не з великою кількістю класів.

5. **Нарисуйте структуру шаблону «Фасад».**



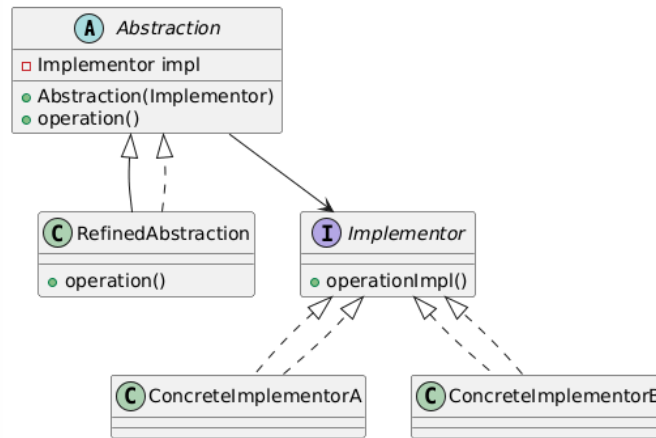
6. **Які класи входять в шаблон «Фасад», та яка між ними взаємодія?**

У шаблон входять: Facade (публічний спрощений інтерфейс), кілька класів Subsystem (реальна логіка підсистеми) та Client; клієнт звертається тільки до Facade, а фасад всередині викликає потрібні методи підсистем у правильному порядку.

7. **Яке призначення шаблону «Міст»?**

«Міст» призначений для розділення абстракції та її реалізації в окремі ієрархії, щоб їх можна було незалежно змінювати й комбінувати, зменшуючи вибух кількості класів при поєднанні різних варіантів.

8. **Нарисуйте структуру шаблону «Міст».**



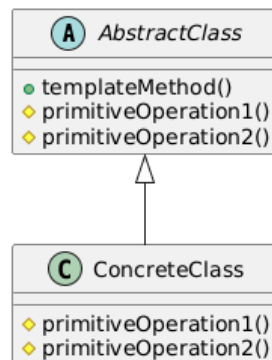
**9. Які класи входять в шаблон «Міст», та яка між ними взаємодія?**

У шаблон входять: Abstraction (зовнішній інтерфейс), RefinedAbstraction (уточнена абстракція), Implementor (інтерфейс реалізації) та ConcreteImplementor (конкретні реалізації); Abstraction тримає посилання на Implementor і делегує йому частину роботи, тому можна незалежно додавати нові абстракції й нові реалізації.

**10. Яке призначення шаблону «Шаблонний метод»?**

«Шаблонний метод» визначає загальний каркас алгоритму в базовому класі, залишаючи окремі кроки для реалізації у підкласах, що дозволяє змінювати деталі алгоритму, не змінюючи його загальну структуру.

**11. Нарисуйте структуру шаблону «Шаблонний метод».**



**12. Які класи входять в шаблон «Шаблонний метод», та яка між ними взаємодія?**

Шаблон включає AbstractClass (містить templateMethod() – готовий алгоритм з викликами «примітивних» операцій) і ConcreteClass (перевизначає ці примітивні операції); клієнт викликає templateMethod() у AbstractClass, а конкретний підклас визначає, як саме виконуються кроки.

**13. Чим відрізняється шаблон «Шаблонний метод» від «Фабричного методу»?**

«Шаблонний метод» визначає скелет алгоритму в базовому класі, дозволяючи підкласам змінювати окремі його кроки, тоді як «Фабричний метод» спеціалізується саме на створенні об'єктів, делегуючи рішення про те, який конкретний продукт створити, підкласам; фабричний метод часто є лише одним кроком у шаблонному методі.

**14. Яку функціональність додає шаблон «Міст»?**

Шаблон «Міст» додає можливість незалежно розширювати ієрархію абстракцій та ієрархію реалізацій і вільно їх комбінувати, уникнувши множення класів типу «Абстракція×Реалізація» і зменшуючи зв'язність між ними.

**Хід роботи**

**ВИСНОВОК**

**КОНТРОЛЬНІ ЗАПИТАННЯ**