1. **FrameWork**

MusicStore/

├── App/

│ ├── main.cpp # Điểm vào chương trình với các tùy chọn UI

│

├── Core/ # Domain core - trọng tâm học OOP

│ ├── Models/ # Domain models - thay vì Entities

│ │ ├── AbstractAudioItem.h/cpp # Lớp cơ sở trừu tượng

│ │ ├── Music.h/cpp # Kế thừa từ AbstractAudioItem

│ │ ├── Album.h/cpp # Kế thừa từ AbstractAudioItem

│ │ ├── Artist.h/cpp

│ │ └── Playlist.h/cpp # Minh họa Composite pattern

│ │

│ ├── Interfaces/ # Các interface minh họa tính đa hình

│ │ ├── IPlayable.h # Interface cho các đối tượng có thể play

│ │ ├── ISortable.h # Interface cho các đối tượng có thể sắp xếp

│ │ ├── IRepository.h # Generic repository interface

│ │ └── IStorable.h # Interface cho đối tượng lưu trữ được

│ │

│ └── Collections/ # Các collection class tùy chỉnh

│ └── MusicLibrary.h/cpp # Minh họa Iterator pattern

│

├── Services/ # Các dịch vụ thể hiện SRP & OCP

│ ├── MusicPlayerService.h/cpp # Singleton pattern

│ ├── MusicLibraryService.h/cpp

│ └── FileSystemService.h/cpp # Strategy pattern với nhiều định dạng

│

├── UI/ # Giao diện đơn giản hơn

│ ├── AbstractView.h/cpp # Lớp cơ sở cho các views (Template Method)

│ ├── ConsoleView.h/cpp # Giao diện console

│ ├── GraphicalView.h/cpp # Giao diện đồ họa

│ └── Assets/

│ └── ...

│

├── Persistence/ # Đơn giản hóa phần lưu trữ

│ ├── FileRepository.h/cpp # Mẫu Repository đơn giản

│ ├── DatabaseRepository.h/cpp # Factory Method pattern

│ └── DatabaseConnection.h/cpp # Adapter pattern

│

├── Utils/ # Các tiện ích

│ ├── Observable.h/cpp # Observer pattern

│ ├── StringUtils.h/cpp # Static utility class

│ └── Logger.h/cpp # Decorator pattern

│

├── Patterns/ # Minh họa các Design Patterns riêng biệt

│ ├── Creational/ # Mẫu thiết kế khởi tạo

│ │ ├── MusicFactory.h/cpp # Factory method

│ │ └── LibrarySingleton.h/cpp # Singleton

│ ├── Structural/ # Mẫu thiết kế cấu trúc

│ │ ├── AudioDecorator.h/cpp # Decorator

│ │ └── UIAdapter.h/cpp # Adapter

│ └── Behavioral/ # Mẫu thiết kế hành vi

│ ├── PlayCommand.h/cpp # Command

│ └── MusicIterator.h/cpp # Iterator

│

├── Tests/ # Kiểm thử đơn giản hơn

│ ├── ModelTests.cpp # Kiểm thử models

│ ├── ServiceTests.cpp # Kiểm thử services

│ └── test\_main.cpp

│

├── Docs/ # Tài liệu học tập

│ ├── ClassDiagrams/ # Sơ đồ lớp UML

│ ├── DesignPatterns.md # Giải thích mẫu thiết kế sử dụng

│ └── OOPPrinciples.md # Giải thích nguyên lý SOLID và OOP

│

├── CMakeLists.txt

└── README.md

1. **Quy Tắc của FrameWork**
2. SOLID
3. OOP core Principles
   * Inheritance
   * Polymorphism
   * Encapsulation
   * Abstraction
   * Composition over inheritance
4. Design Patterns
   * Architecture
     + MVC
     + Repository
   * Creational
     + Factory Method
     + Singleton
   * Structural
     + Adapter
     + Decorator
   * Behavioral
     + Obsever
     + Command
     + Template Method
     + Iterator
     + Strategy
5. Design Principles
   * KISS
   * DRY
   * YAGNI
   * SoC
   * Reusability
   * Testability
6. Clean Architecture
7. Layered Architecture (3 -tier)
8. Dependency Rule
9. Interface Driven Design
10. Unit Test Structure
11. Dry & Reusability
12. Documentation-first / Developer-friendly
13. Static Utils