



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 № 8

з дисципліни: «Технології розроблення програмного забезпечення»

Виконав:

Тимчук Владислав

ІА-34

Перевірив:

Мягкий М. Ю.

Тема: Патерни проектування.

Мета: Вивчити структуру шаблонів «Composite», «Flyweight» (Пристосуванець), «Interpreter», «Visitor» 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

У даній роботі реалізовано частину функціоналу музичного програвача.

Метою роботи є використання структурного шаблону проектування Composite, який дозволяє працювати зі складеними об'єктами (плейлистами) та одиничними об'єктами (треками) однаковим способом. Шаблон забезпечує гнучкість структури та можливість створення вкладених плейлистів.

Хід роботи

Шаблон Composite:

У музичному програвачі логічно мати *композицію* елементів:

- окремий трек,
- плейлист, який може містити треки або інші плейлисти.

Патерн **Composite** дозволяє зручно працювати з обома однаково — як з «музичними елементами».

Діаграма класів:

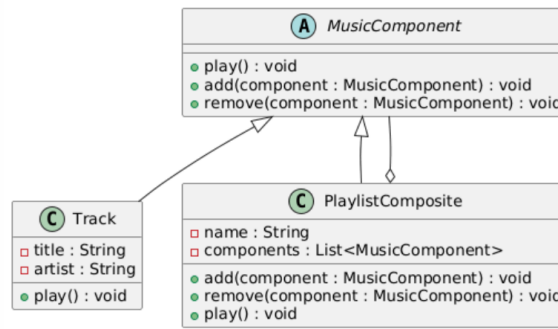


Рис. 1 – Діаграма класів

Розроблено структуру класів:

- **MusicComponent** — абстракція над усіма музичними елементами;
- **Track** — лист, окремий музичний файл;
- **PlaylistComposite** — композиція, яка містить інші елементи.

Реалізовано можливість:

- додавати елементи в плейлист,
- видаляти елементи,
- запускати відтворення всіх елементів дерева.

Увесь код можна знайти за посиланням на *GitHub*:

<https://github.com/fromz67/TRPZ/tree/main/lab8>

ВИСНОВОК

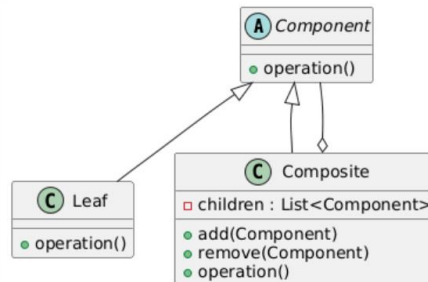
У ході виконання лабораторної роботи було реалізовано частину функціоналу музичного програвача з використанням шаблону Composite. Побудована система дозволяє працювати з треками й плейлистами у єдиній ієрархії, підтримує розширення та вкладені структури.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Яке призначення шаблону «Композит»?

Шаблон Композит використовується для представлення ієрархічних структур типу дерево, дозволяючи працювати з окремими об'єктами та їх композиціями однаковою чином через єдиний інтерфейс.

2. Нарисуйте структуру шаблону «Композит».



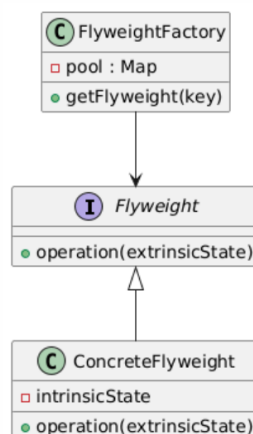
3. Які класи входять в шаблон «Композит», та яка між ними взаємодія?

Шаблон складається з **Component** (спільний інтерфейс), **Leaf** (окремий елемент без дочірніх об'єктів) і **Composite** (контейнер, що містить інші компоненти), де клієнт працює з ними через один інтерфейс, а **Composite** делегує операції всім своїм дочірнім елементам.

4. Яке призначення шаблону «Легковаговик»?

Шаблон «Легковаговик» призначений для економії пам'яті через повторне використання об'єктів, виділяючи внутрішній стан як спільний і зберігаючи зовнішній стан окремо в клієнта.

5. Нарисуйте структуру шаблону «Легковаговик».



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

Основні класи — Flyweight (інтерфейс спільних операцій), ConcreteFlyweight (містить внутрішній стан і виконує операції), та FlyweightFactory (керує пулом об'єктів і повертає вже створені екземпляри), при цьому клієнт передає зовнішній стан під час виклику операцій.

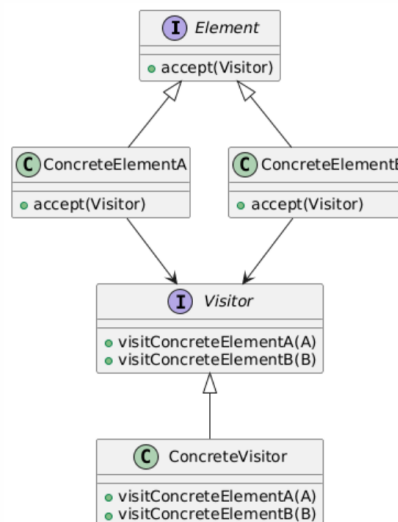
7. Яке призначення шаблону «Інтерпретатор»?

Шаблон «Інтерпретатор» призначений для опису граматики простої мови та створення об'єктів-виразів, які можуть інтерпретуватися для виконання команд або обчислення значень.

8. Яке призначення шаблону «Відвідувач»?

Шаблон «Відвідувач» дозволяє додавати нові операції для різних класів об'єктів без зміни їхньої структури, передаючи обробку окремому об'єкту-відвідувачу.

9. Нарисуйте структуру шаблону «Відвідувач».



10. Які класи входять в шаблон «Відвідувач», та яка між ними взаємодія?

Шаблон складається з **Element** (інтерфейс елементів), **ConcreteElement** (реалізації елементів), **Visitor** (інтерфейс відвідувачів) та **ConcreteVisitor** (конкретні операції), де елементи викликають `асерт()`, передаючи себе

відвідувачу, а відвідувач виконує потрібну операцію відповідно до типу елемента.