

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут КНІТ  
Кафедра ПЗ**

**ЗВІТ**

До лабораторної роботи № 4

**З дисципліни:** *“Моделювання та аналіз програмного забезпечення”*

**На тему:** *“Структурні шаблони”*

**Лектор:**

доц. каф. ПЗ  
Сердюк П.В.

**Виконав:**

ст. гр. ПЗ-22  
Чаус Олег

**Прийняв:**

викл. каф. ПЗ  
Микуляк А. В.

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Σ= \_\_\_\_ .

Львів – 2023

**Тема роботи:** Структурні шаблони.

**Мета роботи:** Здобути навички використання структурних шаблонів проектування при моделюванні програмних систем.

### ТЕОРЕТИЧНІ ВІДОМОСТІ

Структурні шаблони (англ. structural patterns) — шаблони проектування, у яких розглядається питання про те, як із класів та об'єктів утворюються більші за розмірами структури. Структурні шаблони рівня класу використовують успадкування для утворення композицій із інтерфейсів та реалізацій. Структурні шаблони рівня об'єкта компонують об'єкти для отримання нової функціональності. Додаткова гнучкість у цьому разі пов'язана з можливістю змінювати композицію об'єктів під час виконання, що є неприпустимим для статичної композиції класів.

### ЗАВДАННЯ

Розробити структурні шаблони проектування відповідно до прецедентів обраного варіанту ігрової логіки. Вибрати один з прецедентів, для якого найбільш доцільно застосувати структурний шаблон (не всі прецеденти цього потребуватимуть).

Обов'язковий шаблон для реалізації:

- Розробити шаблон Фасад для моделі класів проекту свого проекту.

Розробити ще 2 структурні шаблони на вибір з наступних:

- Декоратор,
- Компонувальник,
- Легковаговик,
- Замісник (Проху).

Представити діаграму класів ігрового проекту (із відображенням на ній використаних шаблонів).

## UML-ДІАГРАМА КЛАСІВ

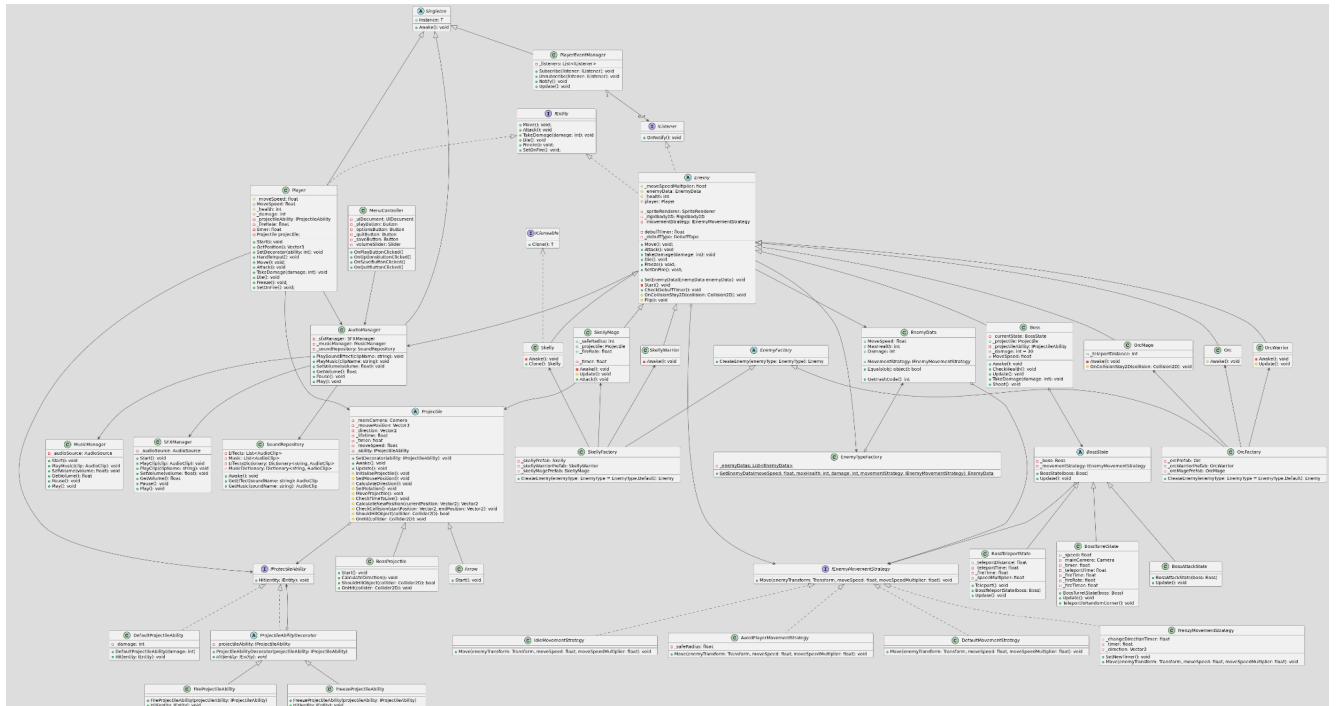


Рис. 1. Загальна UML-діаграма класів.

Було реалізовано три структурних патерни – фасад, декоратор, легковаговик.

- Фасад

Фасад — це структурний патерн проектування, який надає простий інтерфейс до складної системи класів, бібліотеки або фреймворку.

Фасад — це простий інтерфейс для роботи зі складною підсистемою, яка містить безліч класів. Фасад може бути спрощеним відображенням системи, що не має 100% тієї функціональності, якої можна було б досягти, використовуючи складну підсистему безпосередньо. Разом з тим, він надає саме ті «фічі», які потрібні клієнтові, і приховує все інше.

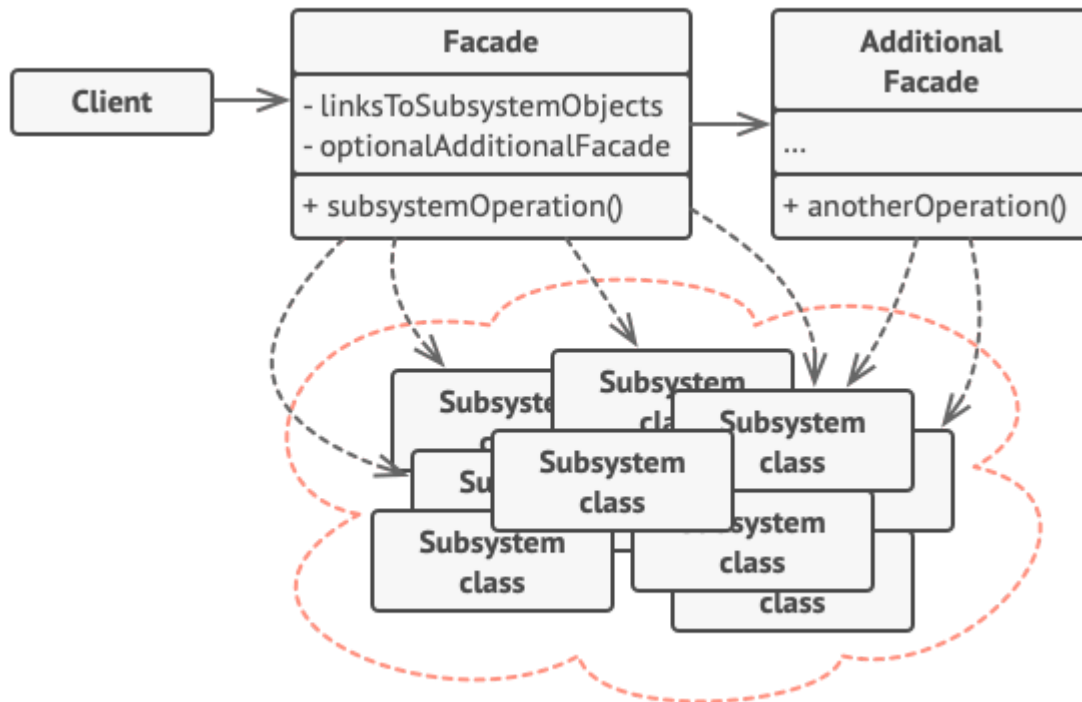


Рис. 2. Класичний випадок використання Facade

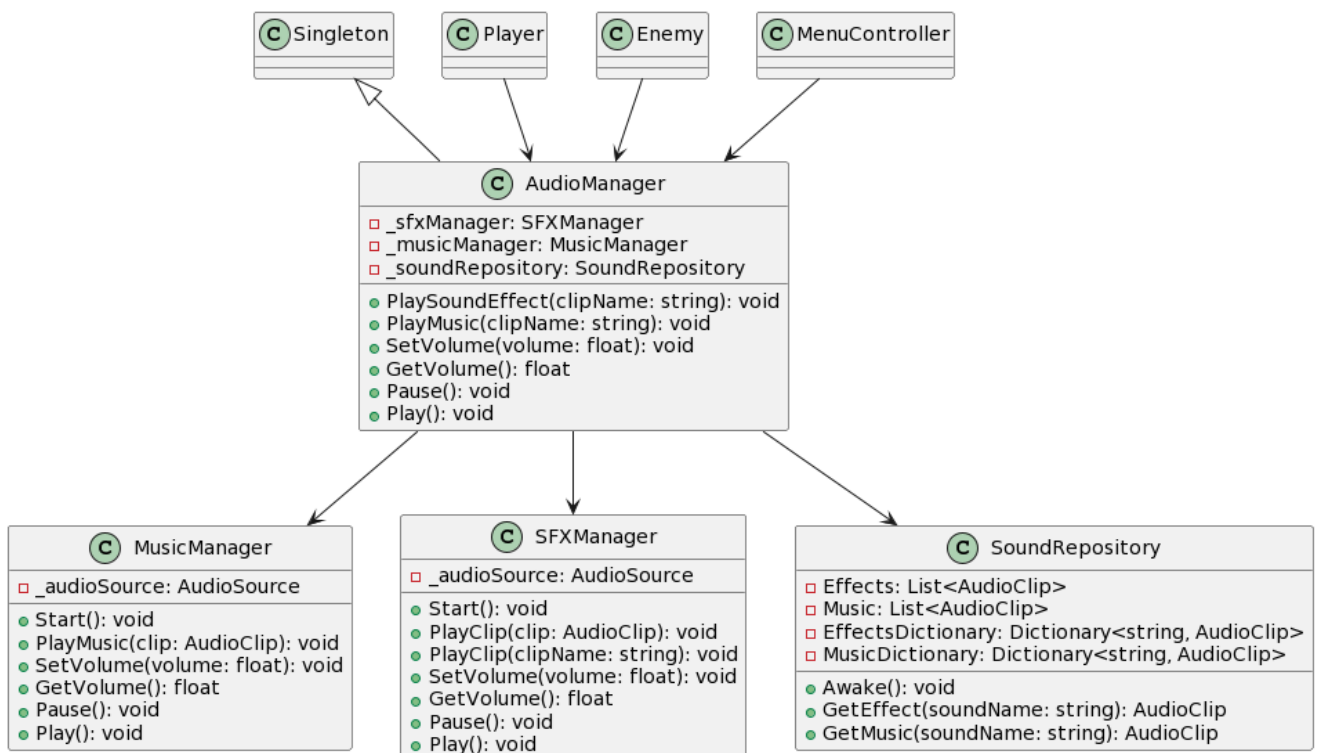


Рис. 3. Використання патерну Facade.

- Декоратор

Декоратор — це структурний патерн проектування, що дає змогу динамічно додавати об'єктам нову функціональність, загортаючи їх у корисні «обгортки». Декоратор має альтернативну назву — обгортка. Вона більш вдало описує суть

патерна: ви розміщуєте цільовий об'єкт у іншому об'єкті-обгортці, який запускає базову поведінку об'єкта, а потім додає до результату щось своє. Обидва об'єкти мають загальний інтерфейс, тому для користувача немає жодної різниці, з чим працювати — з чистим чи загорнутим об'єктом. Ви можете використовувати кілька різних обгортки одночасно — результат буде мати об'єднану поведінку всіх обгортки.

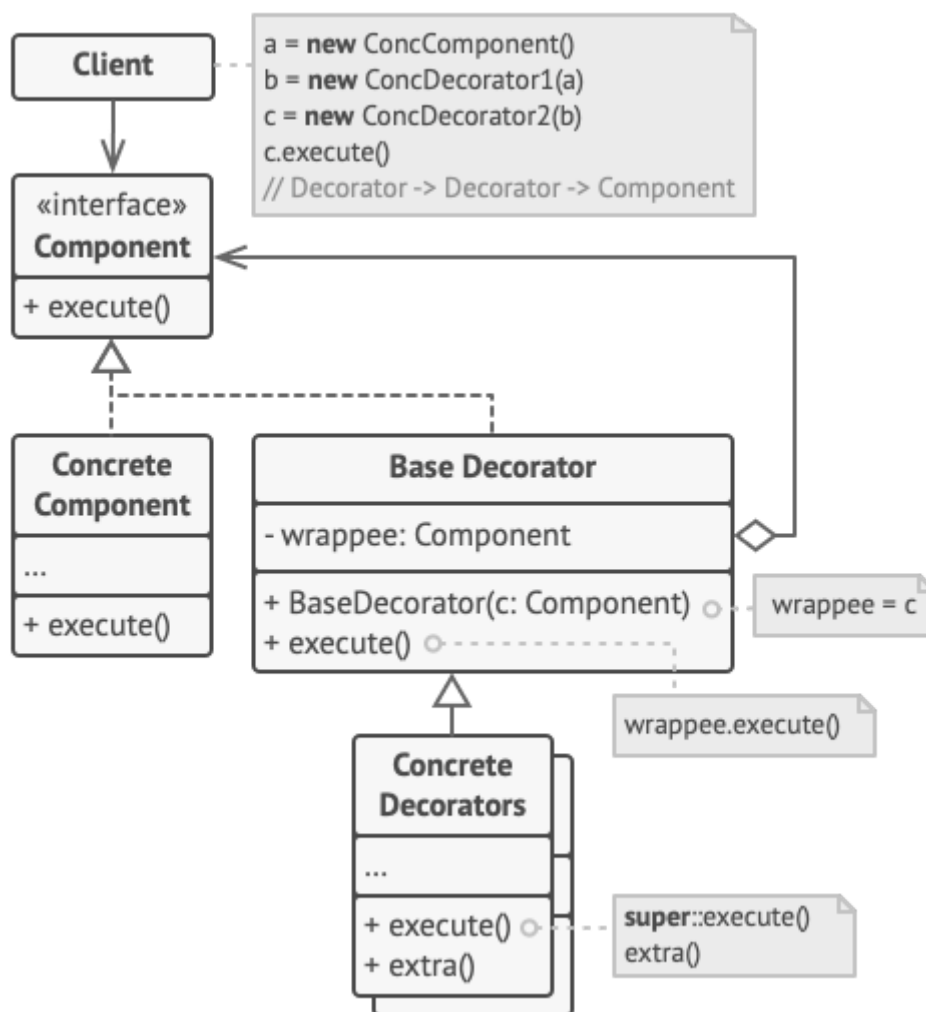


Рис. 4. Класичний випадок використання Decorator.

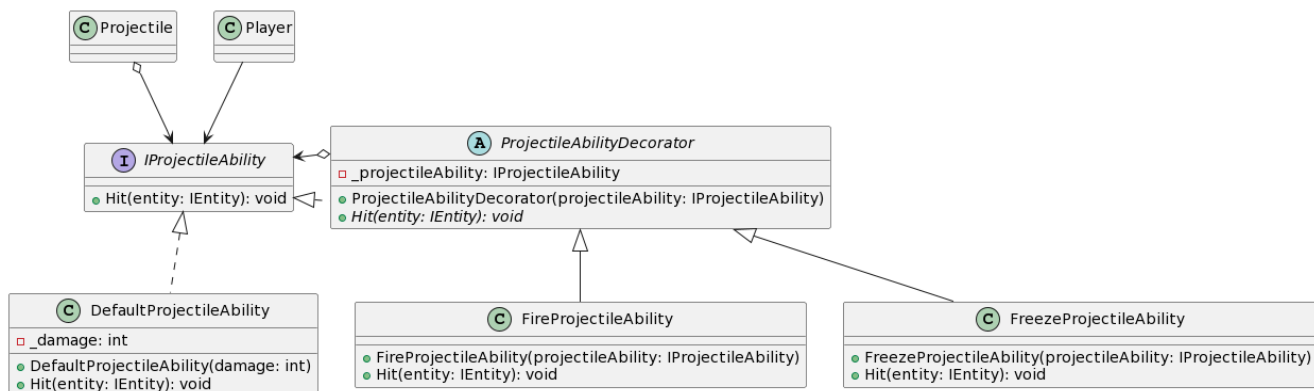


Рис. 5. Використання шаблону Decorator.

#### - Легковаговик

Легковаговик — це структурний патерн проектування, що дає змогу вмістити більшу кількість об'єктів у відведеній оперативній пам'яті. Легковаговик заощаджує пам'ять, розподіляючи спільний стан об'єктів між собою, замість зберігання однакових даних у кожному об'єкті. Шаблон Легковаговик використовує загальнодоступний легкий об'єкт (flyweight, легковаговик), який одночасно може використовуватися у великій кількості контекстів. Стан цього об'єкта поділяється на внутрішній, що містить інформацію, незалежну від контексту, і зовнішній, який залежить або змінюється разом з контекстом легковаговика. Об'єкти клієнтів відповідають за передачу зовнішнього стану легковаговика, коли йому це необхідно.

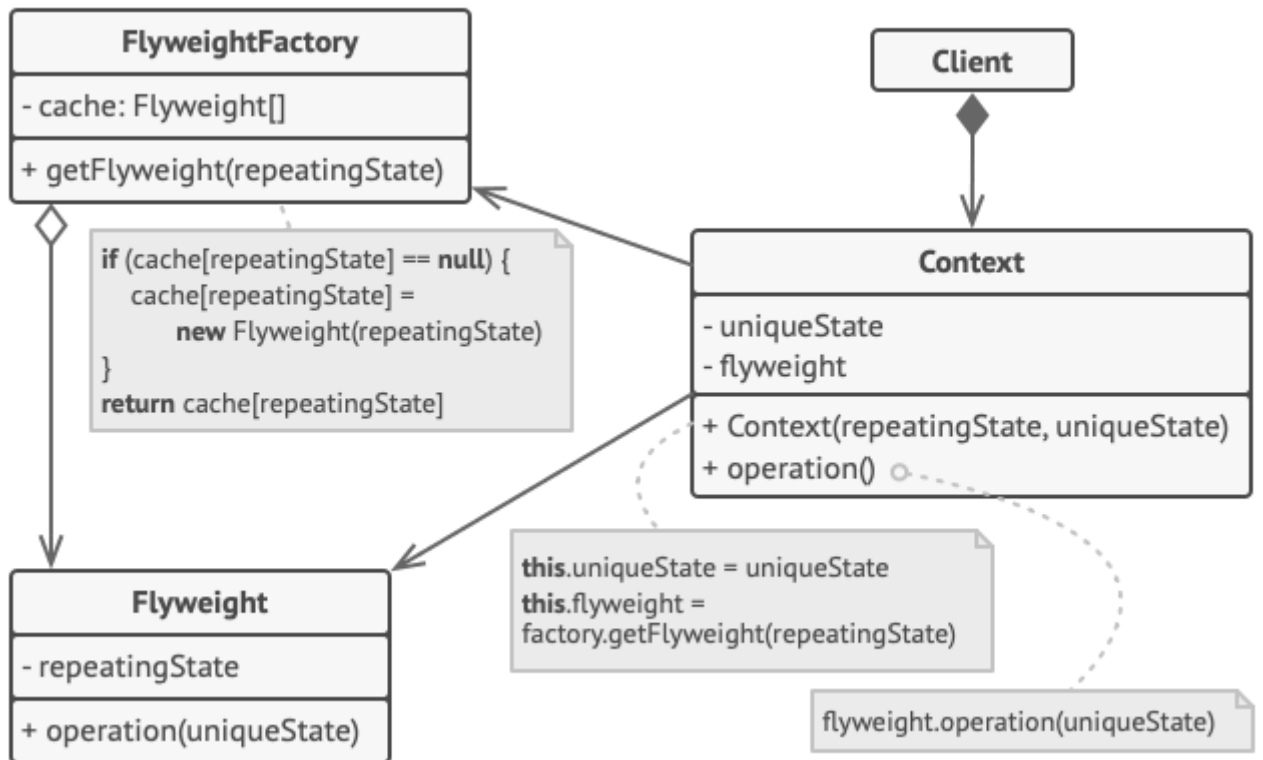


Рис. 6. Приклад використання патерну Flyweight.

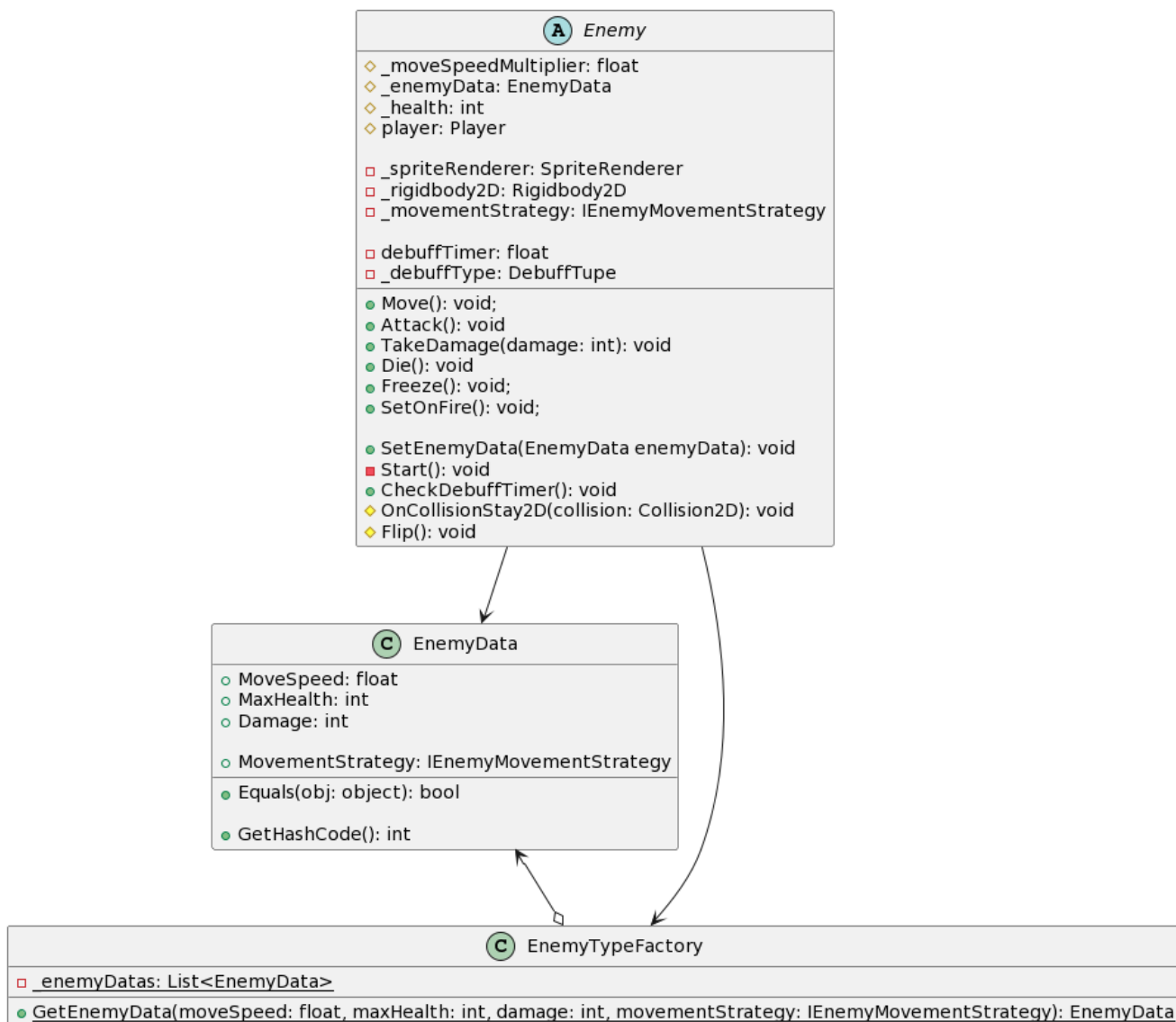


Рис. 7. Використання Легковаговика.

## ВИСНОВКИ

Здобув навички використання структурних шаблонів проектування при моделюванні програмних систем.