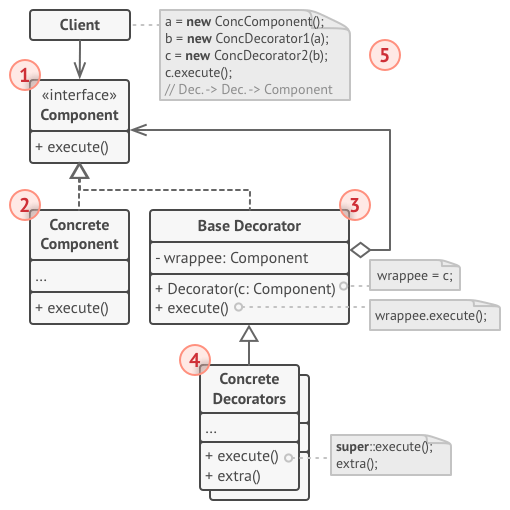
**ДЕКОРАТОР** - динамически наделяет объект новыми возможностями и является гибкой альтернативой субклассированию в области расширения функциональности. Также можно назвать оберткой.

ПРИМЕНЕНИЕ:

- когда нужно добавлять обязанности объектам на лету, незаметно для кода, который их использует

- когда нельзя расширить обязанности объекта с помощью наследования

РЕАЛИЗАЦИЯ:

**1. АБСТРАКЦИЯ КОМПОНЕНТА**, который будет декорироваться дополнительным фунционалом. Например, напиток + метод получения цены

**2. КОНКРЕТНЫЙ КОМПОНЕНТ**: разные конкретные реализации абстракции компонента. Например: кофе, чай + метод получения цены со своей ценой.

**3. АБСТРАКЦИЯ ДЕКОРАТОРА**: т.е. того, что будет представлять дополнительную функцию. Также должна наследоваться от абстракции компонента, чтобы ее можно было в итоге выдать за реальный компонент. Содержит ссылку на любой компонент, к которому она добавляется. Например: добавки

**4. КОНКРЕТНЫЕ ДЕКОРАТОРЫ**: разные конкретные реализации абстракции декоратора. Например: молоко, соя + метод получения цены со своей ценой + вызов метода получения цены компонента, к которому добавлен данный декоратор.

**5. КЛИЕНТ**: может оборачивать простые компоненты и декораторы в другие декораторы, работая со всеми объектами через общий интерфейс компонентов.

ПЛЮСЫ:

- большая гибкость, чем у наследования

- позволяет добавлять обязанности на лету

- можно добавлять несколько новых обязанностей сразу

- позволяет иметь несколько мелких объектов вместо одного объекта на все случаи жизни

МИНУСЫ:

- трудно конфигурировать многократно обёрнутые объекты

- обилие крошечных классов