

Паттерны (шаблони) проектирования

Структурные паттерны



Eugeny Berkunsky, Computer Science dept.,
National University of Shipbuilding
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>

Структурные паттерны

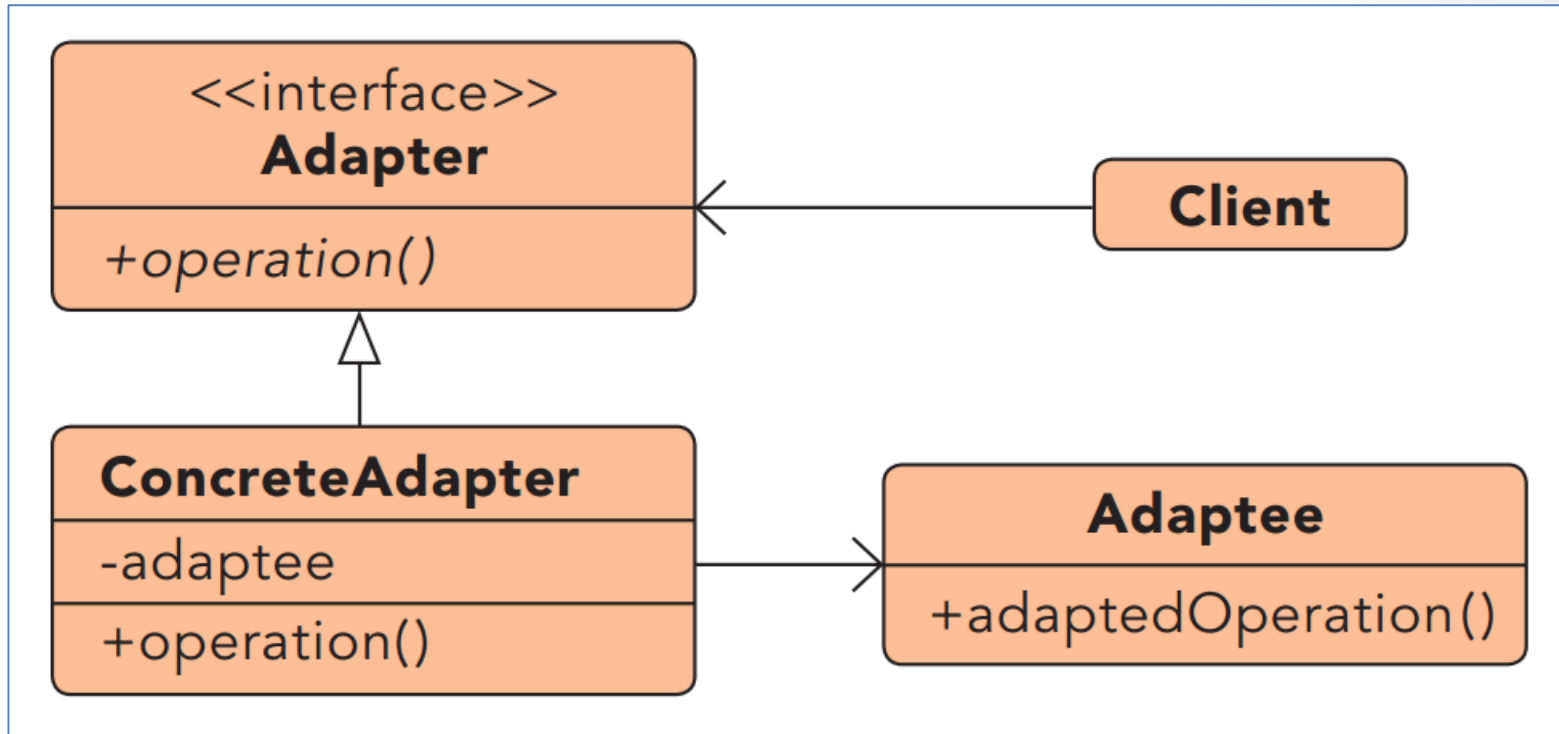
Шаблоны проектирования, в которых рассматривается вопрос о том, как из классов и объектов образуются более крупные структуры.



Структурные паттерны

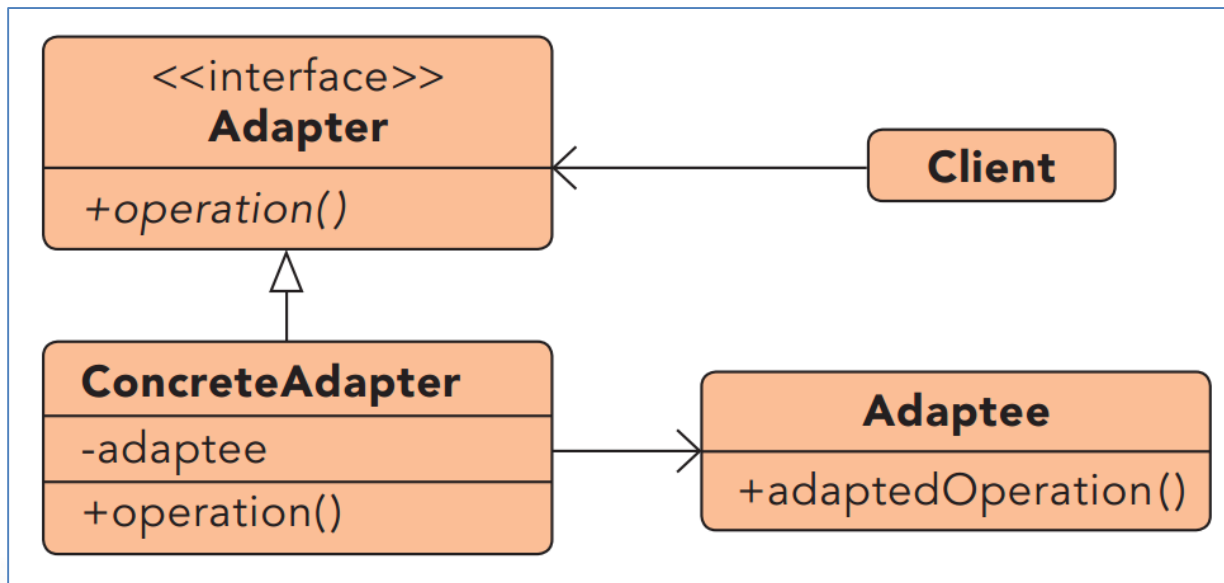
- Adapter/Адаптер
- Bridge/Мост
- Composite/Компоновщик
- Decorator/Декоратор
- Facade/Фасад
- Flyweight/Приспособленец
- Proxy/Заместитель
- Front Controller (перевод не придумали)

Adapter/Адаптер

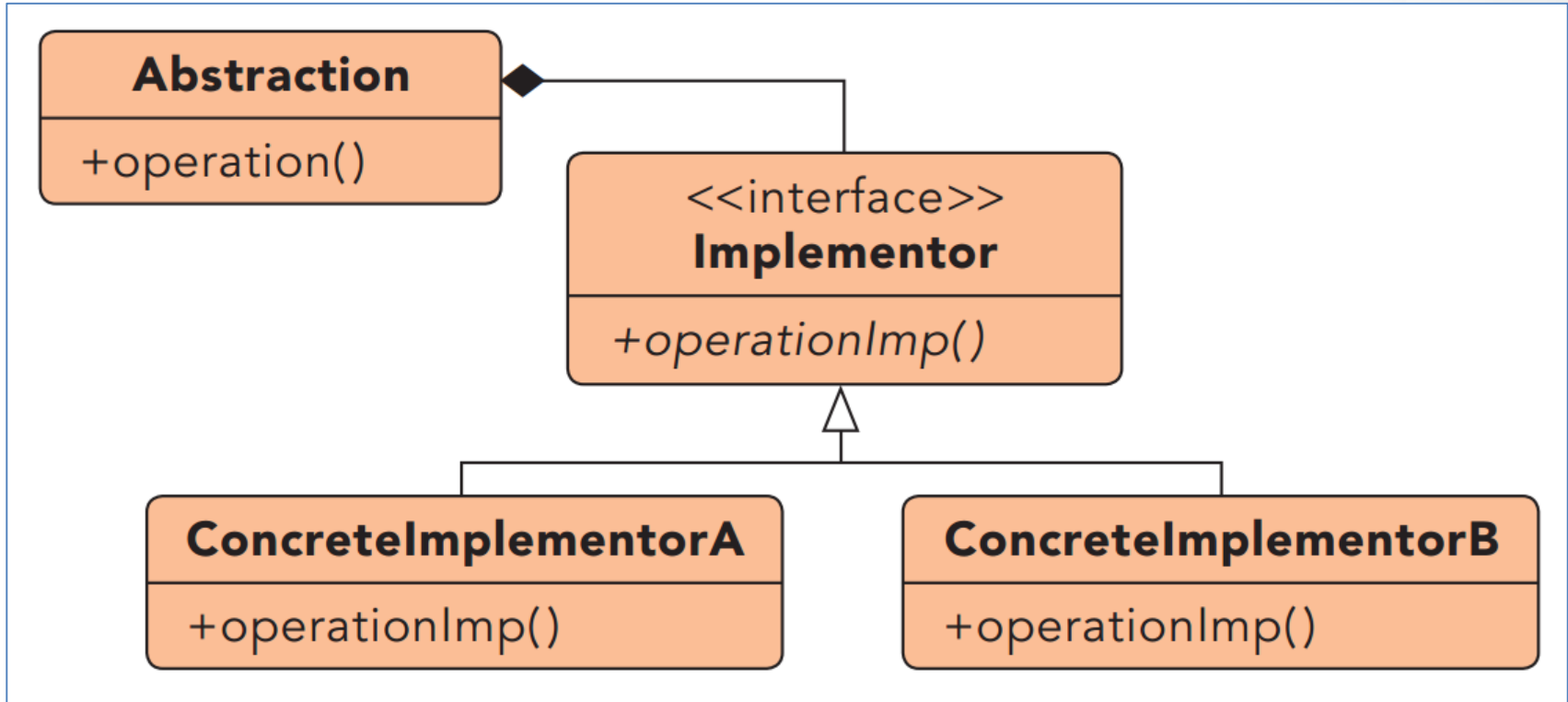


Adapter/Адаптер

Адаптер дозволяє організувати роботу класів з несовместимими інтерфейсами після того, як вони були спроектовані

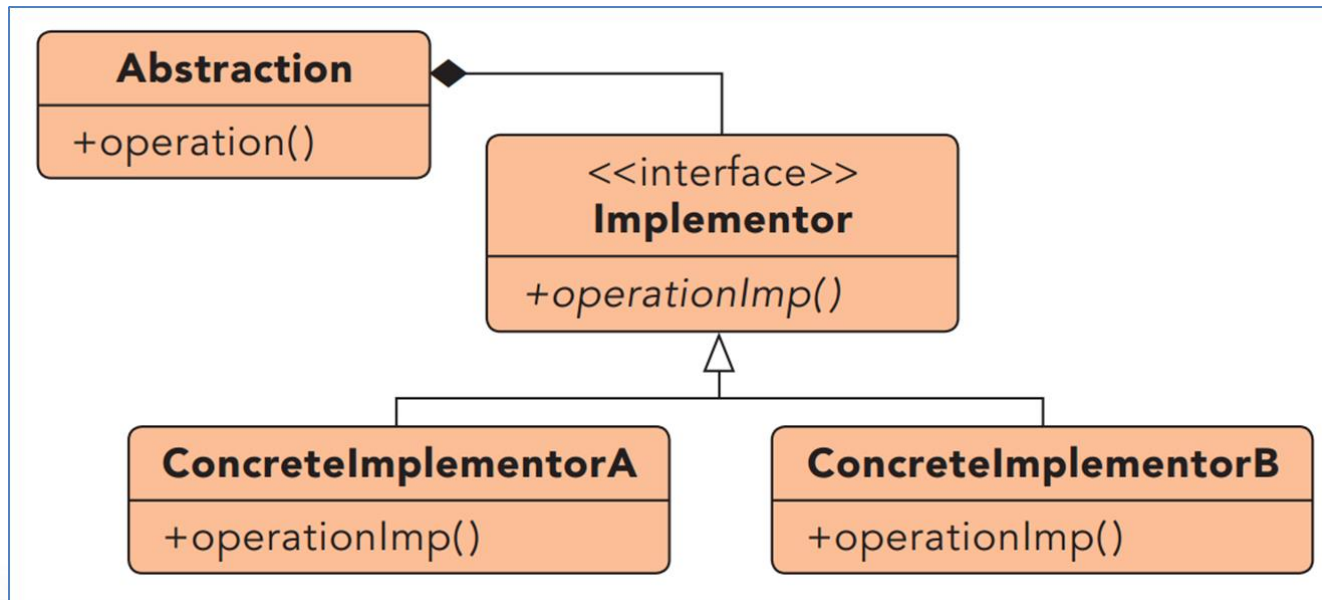


Bridge/Mост



Bridge/Мост

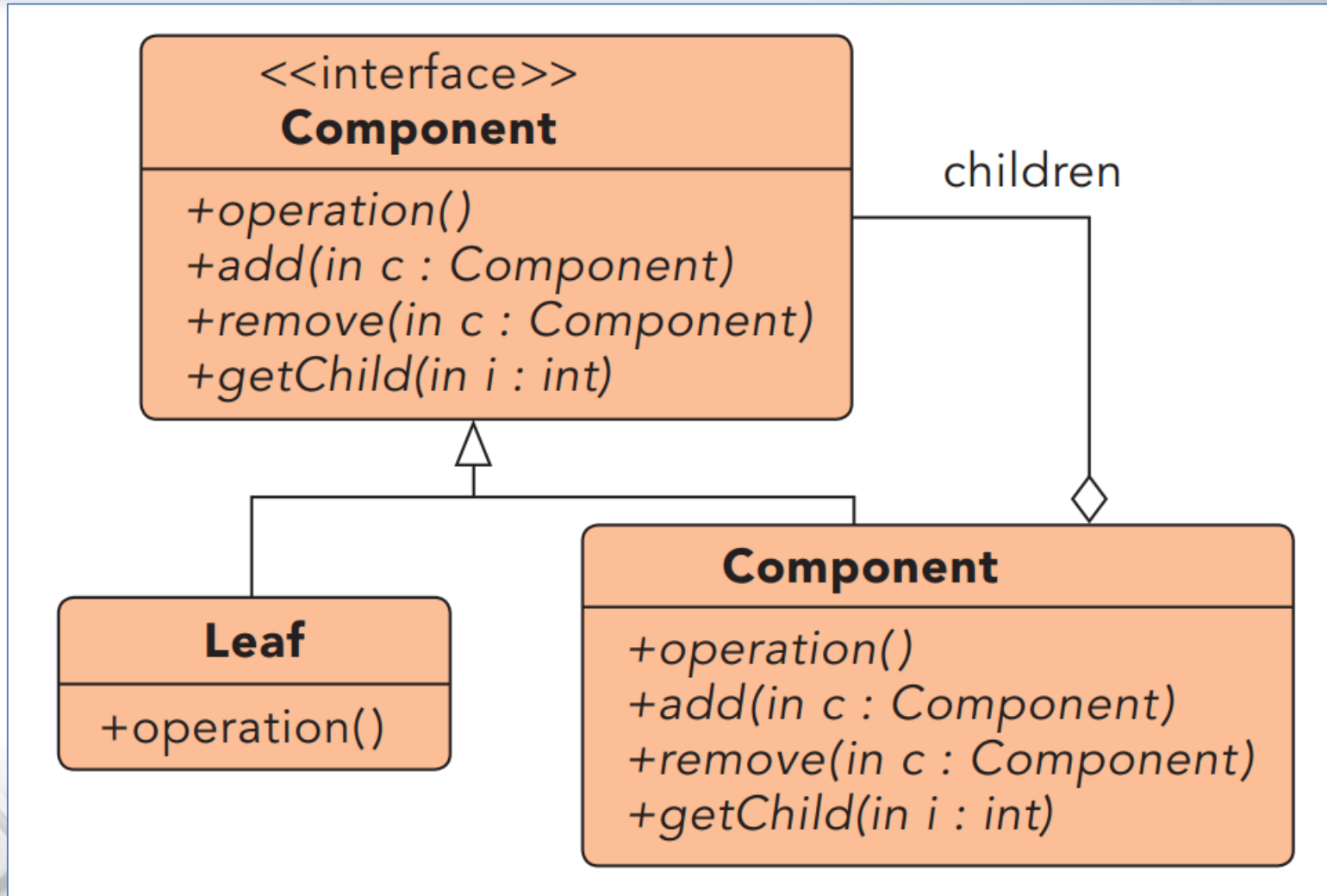
При реалізації через паттерн мост, змінення структури інтерфейса не перешкоджає зміненню структури реалізації.



Bridge/Мост

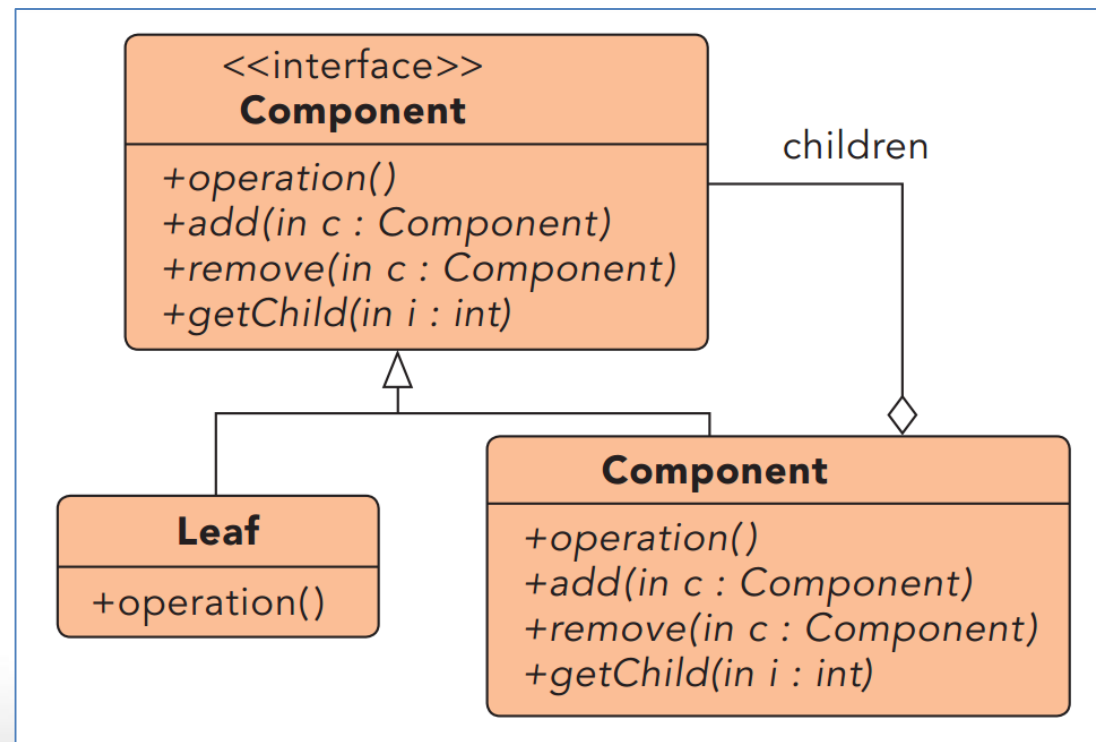
- Рассмотрим такую абстракцию как фигура.
- Есть множество типов фигур, каждая со своими свойствами и методами. Однако есть что-то, что объединяет все фигуры.
- Каждая фигура должна уметь рисовать себя, масштабироваться и т. п.
- Однако, рисование графики может отличаться в зависимости от типа ОС, или графической библиотеки.
- Фигуры должны иметь возможность рисовать себя в различных графических средах, но реализовывать в каждой фигуре все способы рисования или модифицировать фигуру каждый раз при изменении способа рисования непрактично.
- В этом случае помогает шаблон Bridge, позволяя создавать новые классы, которые будут реализовывать рисование в различных графических средах.

Composite/Компоновщик



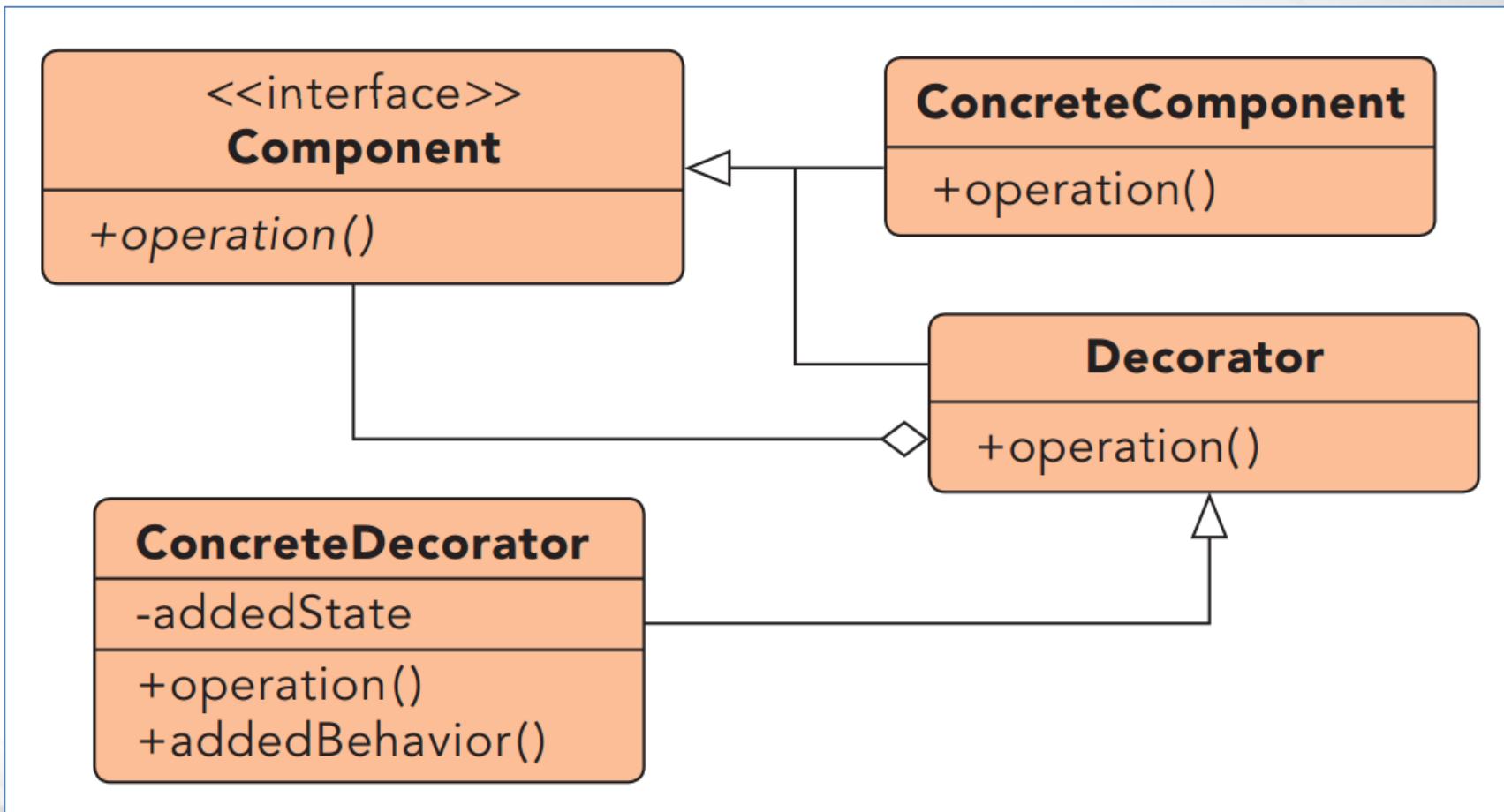
Composite/Компоновщик

Компоновщик дозволяє клієнтам звертатися до окремих об'єктів і до груп об'єктів однаково.



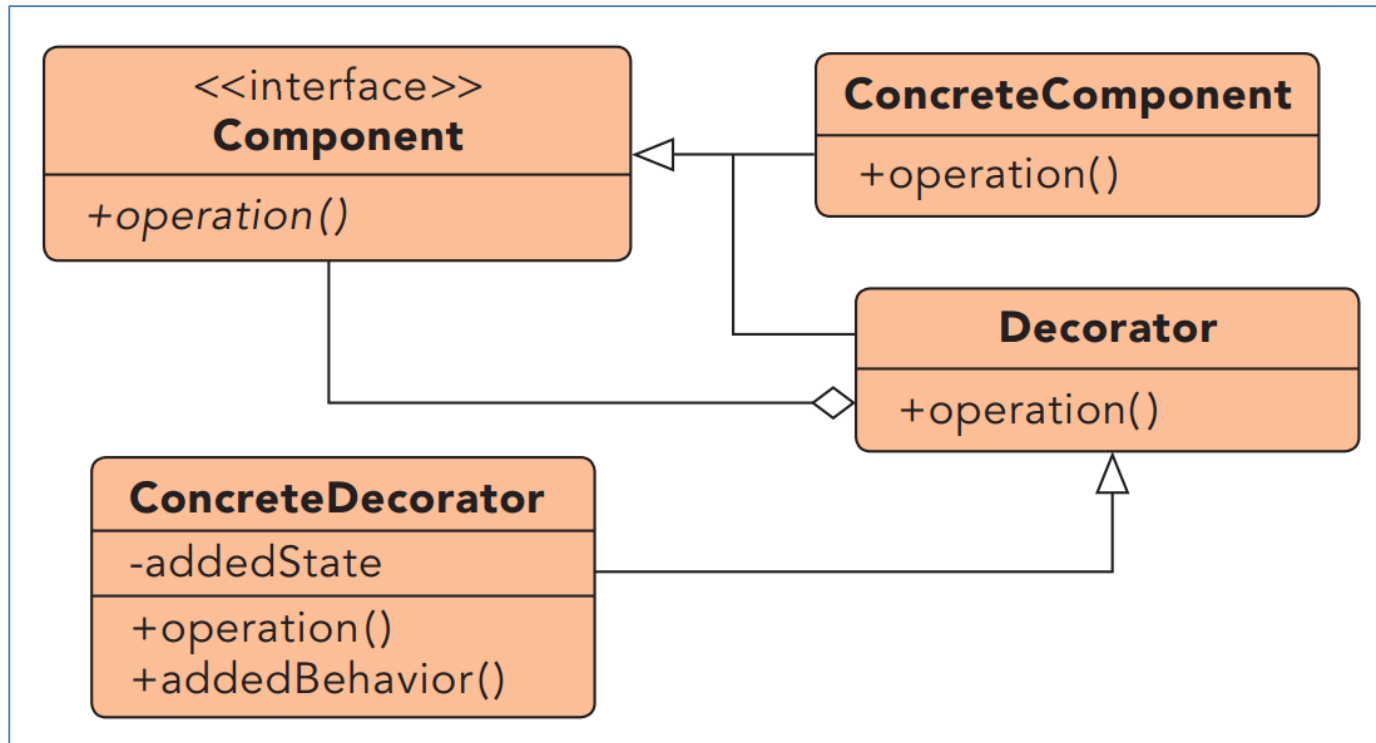
Три шаблони GoF основані на рекурсивній композиції: Компоновщик, Декоратор і Цепочка Відповідальності.

Decorator/Декоратор



Decorator/Декоратор

Декоратор предоставляет гибкую альтернативу практике создания подклассов с целью расширения функциональности.

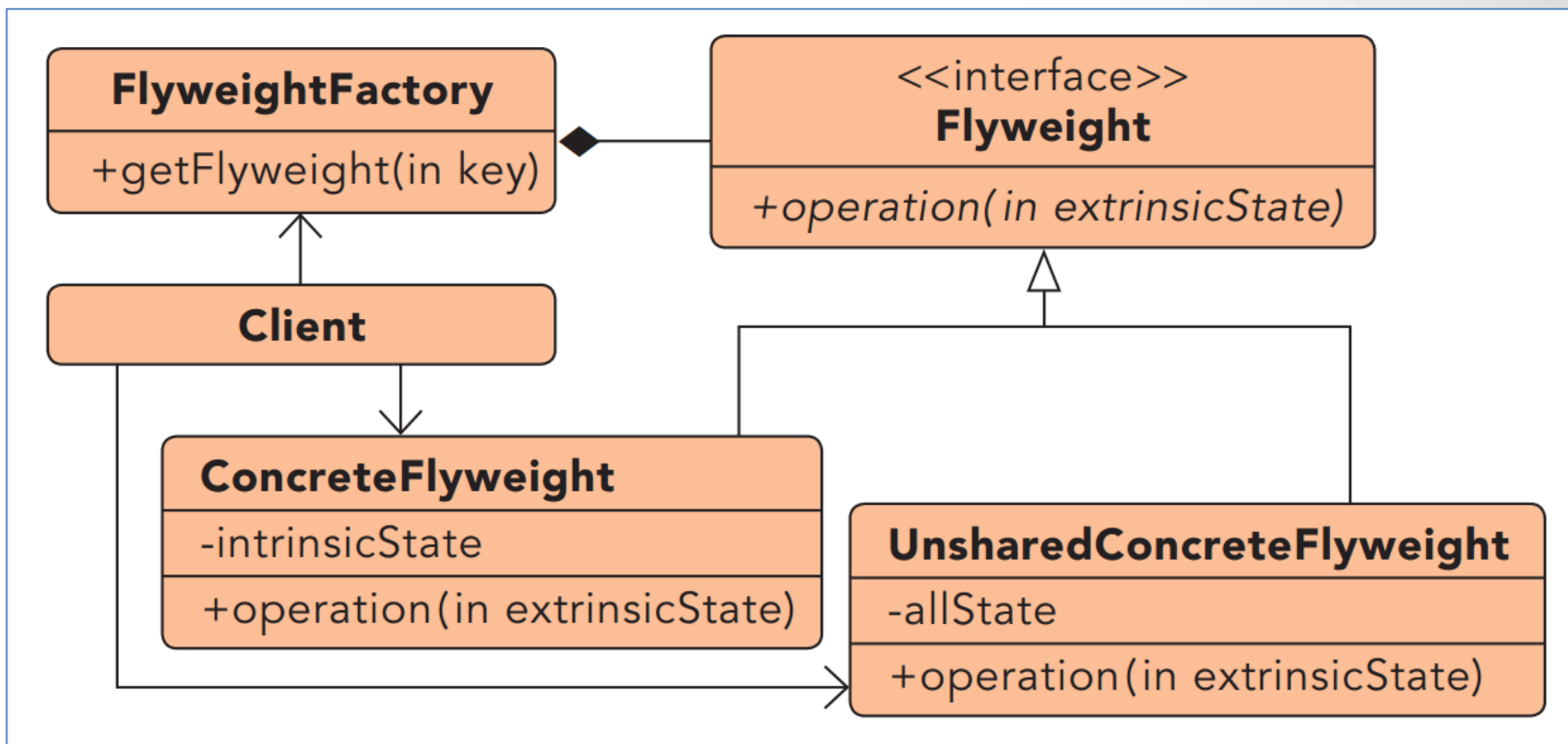


Декоратор предназначен для того, чтобы позволить добавлять объектам ответственность без создания подклассов.

Компоновщик фокусируется не на украшении, а на представлении.

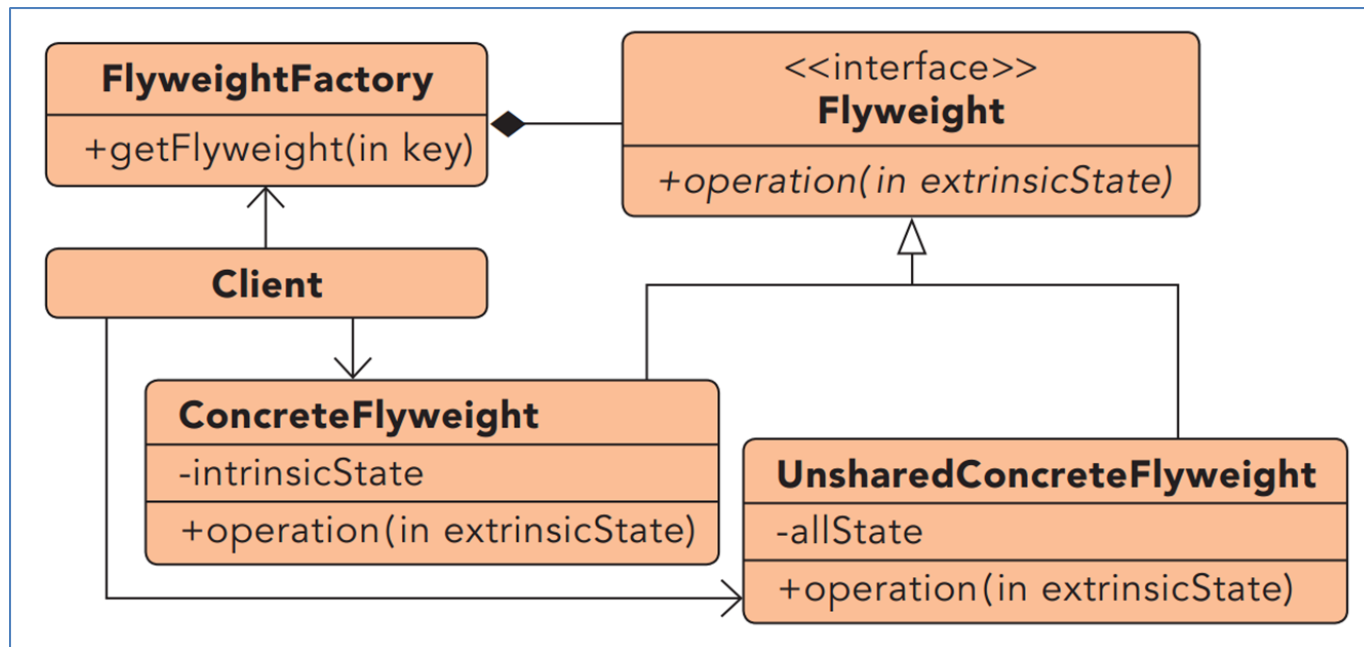
Следовательно, Компоновщик и Декоратор очень часто используются вместе.

Flyweight/Приспособленец



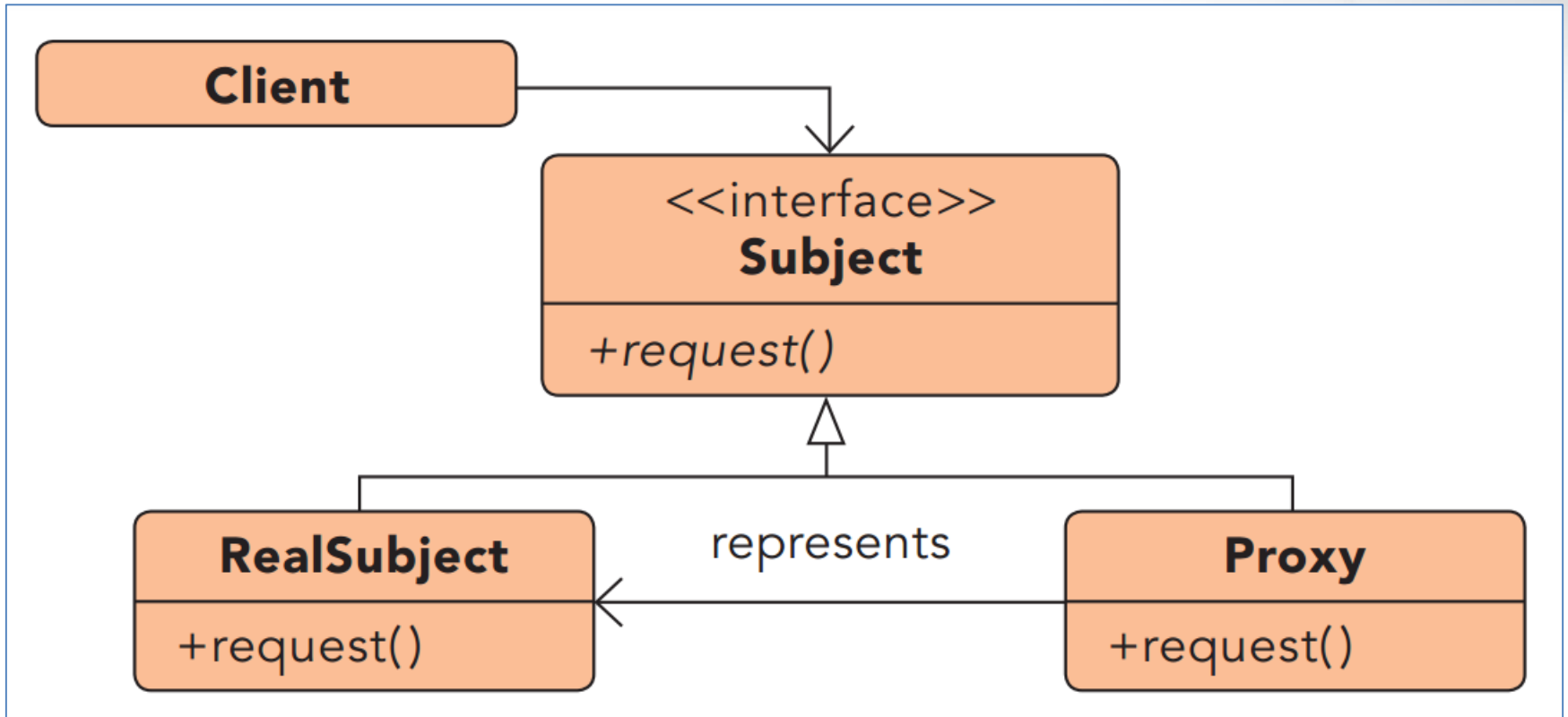
Flyweight/Приспособленец

Приспособленець призначений для роботи з великою кількістю малих об'єктів



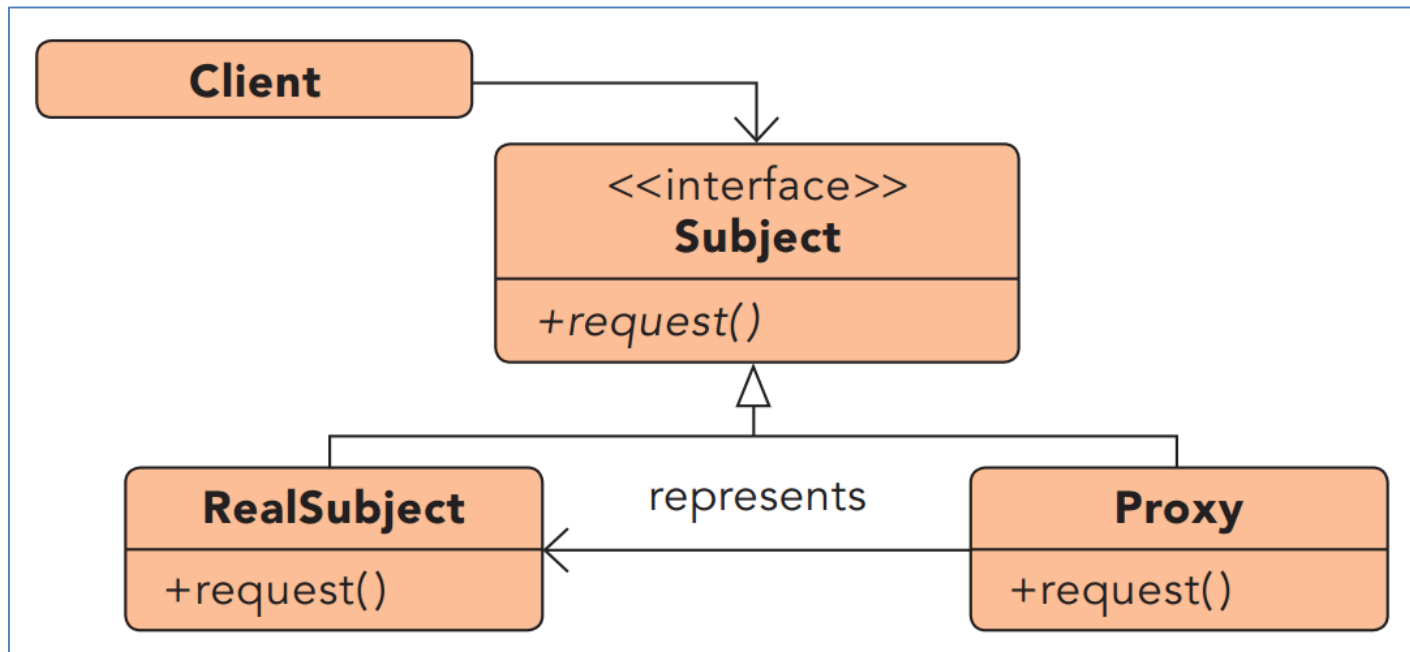
Приспособленець часто використовується разом з Компоновщиком для представлення ієрархічної структури в вигляді ациклічного направленного графа з розділяемими листовими вершинами.

Proxy/Заместитель



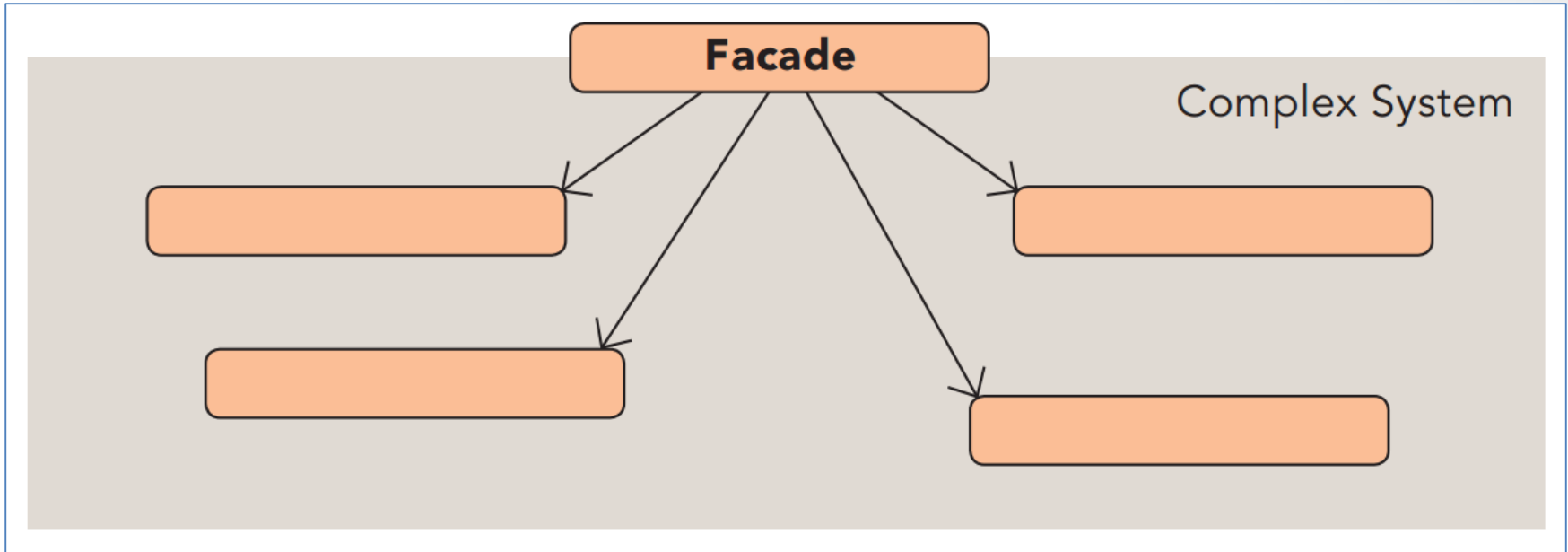
Прoxy/Заместитель

Предназначен для решения проблемы, когда необходимо управлять доступом к объекту так, чтобы создавать громоздкие объекты «по требованию»



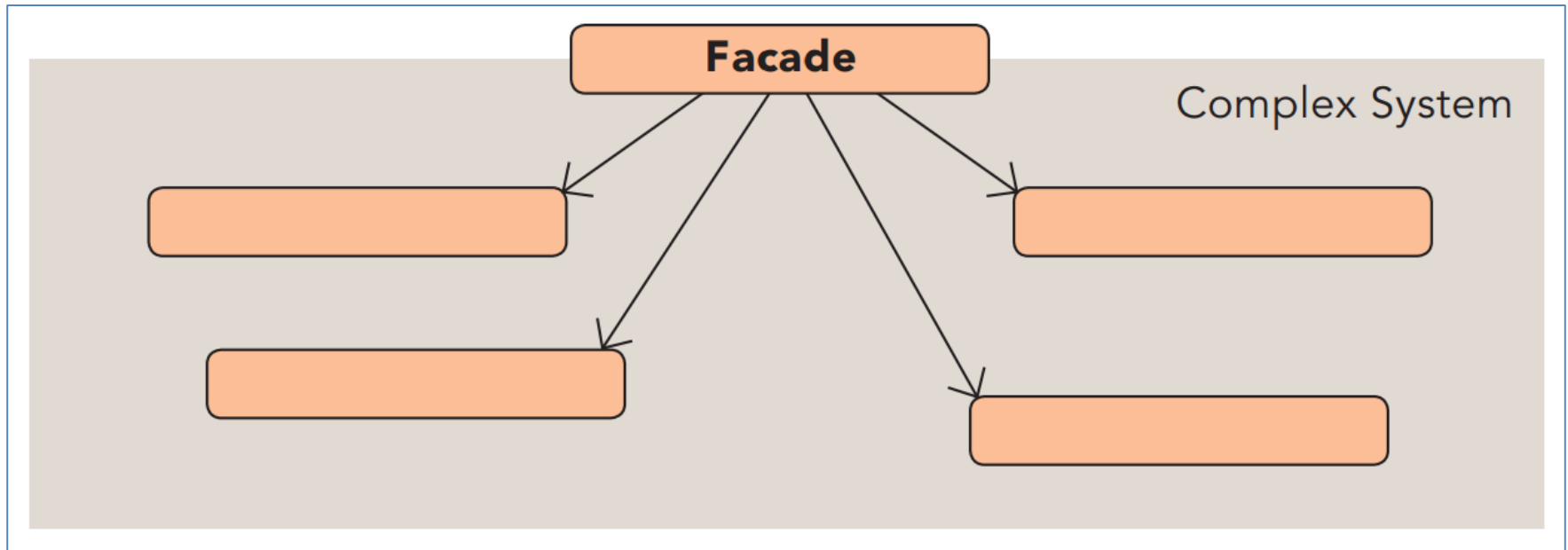
Декоратор и Заместитель имеют разные назначения, но похожую структуру. Оба описывают как предоставить уровень косвенности для другого объекта, и реализация содержит ссылку на объект, которому они передают запросы

Facade/Фасад



Facade/Фасад

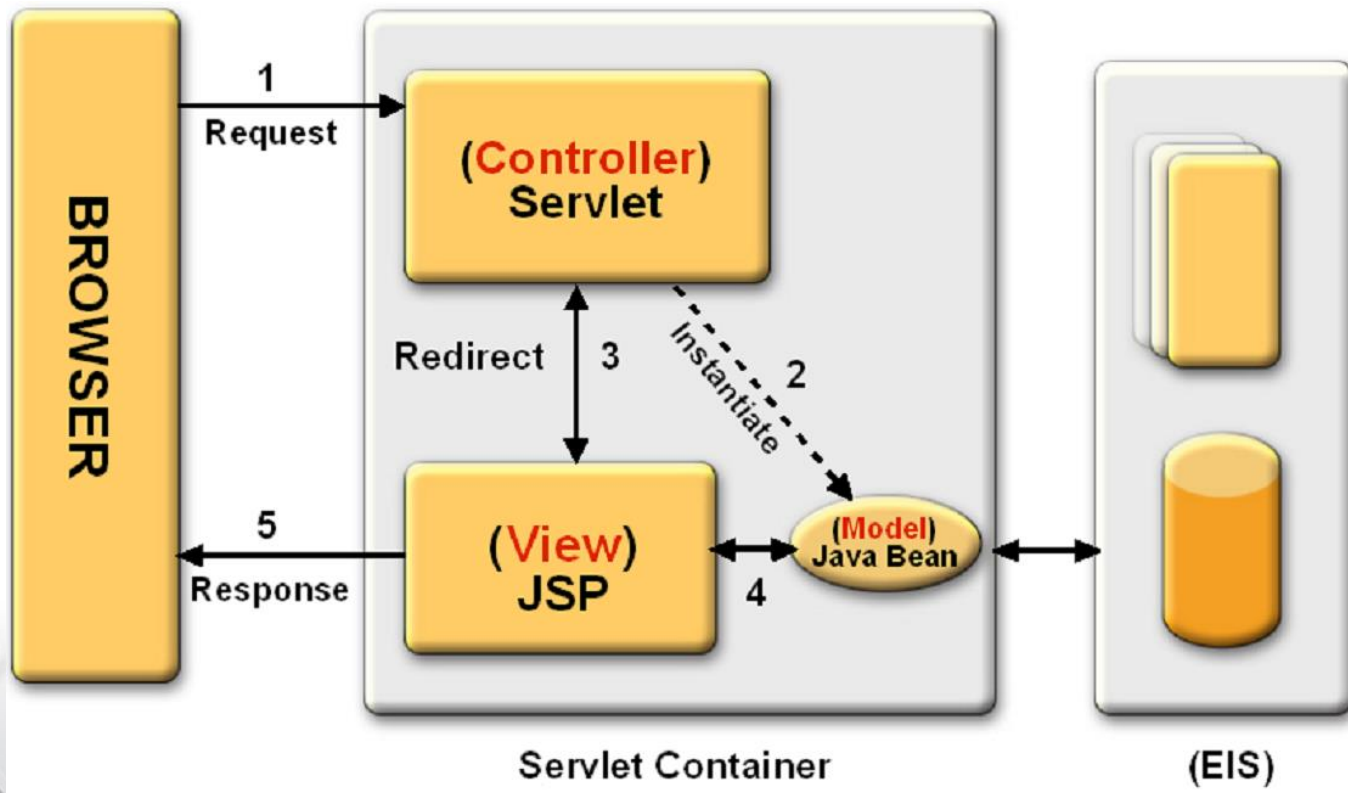
Фасад позволяет скрыть сложность системы путем сведения всех возможных внешних вызовов к одному объекту, делегирующему их соответствующим объектам системы.



Фасад определяет новый интерфейс, в то время как Адаптер использует старый интерфейс. Нужно помнить, что Адаптер позволяет заставить два существующих интерфейса работать вместе, вместо определения абсолютно нового интерфейса

Front Controller

Задачей фронт-контроллера является предоставление единой точки входа для обработки всех запросов и вызов соответствующего поведения в зависимости от запроса



Вопросы?



Паттерны (шаблони) проектирования

Структурные паттерны



Eugeny Berkunsky, Computer Science dept.,
National University of Shipbuilding
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>