



Design Patterns

by Versus 2020

Visitor
Посетитель

Поведенческий шаблон





Суть шаблона

Добавляет в программу новые операции, не изменяя классы объектов, над которыми эти операции могут выполняться.



Проблема

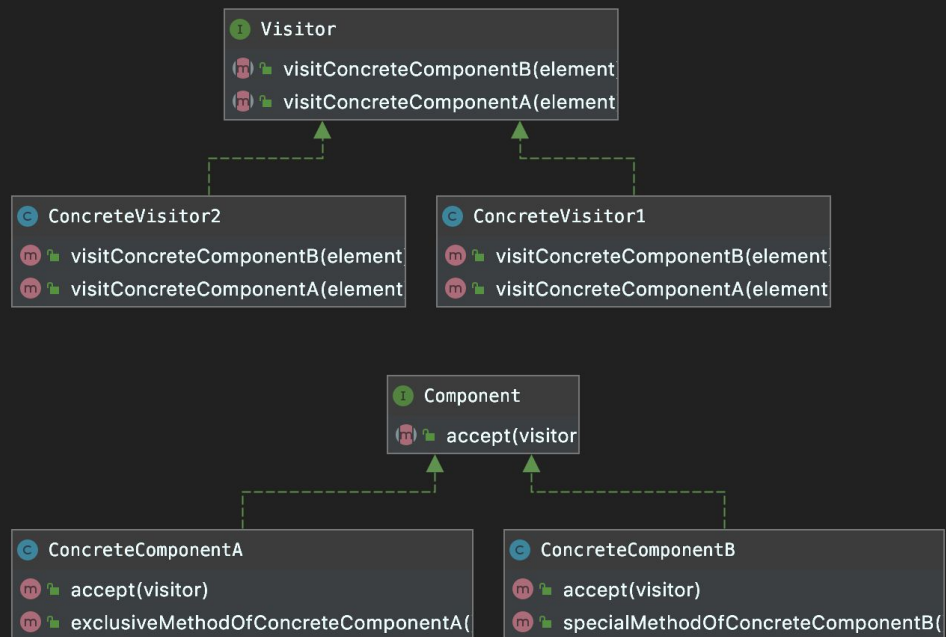
Нельзя трогать и изменять класс нужного вам программного узла для расширения функционала.



Решение

Разместить новое поведение в отдельном классе, вместо того чтобы множить его сразу в нескольких классах.

Структура



Пример кода





Применимость

Нужно выполнить какую-то операцию над всеми элементами сложной структуры объектов.

Над объектами сложной структуры объектов надо выполнять некоторые не связанные между собой операции.

Новое поведение имеет смысл только для некоторых классов из существующей иерархии.



Шаги реализации

1. Создайте интерфейс посетителя и объявите в нём методы «посещения» для каждого класса элемента.
2. Опишите интерфейс элементов.
3. Реализуйте методы принятия во всех конкретных элементах.
4. Иерархия элементов должна знать только о базовом интерфейсе посетителей.
5. Для каждого нового поведения создайте конкретный класс посетителя.



Преимущества

Упрощает добавление операций, работающих со сложными структурами объектов.

Объединяет родственные операции в одном классе.

Посетитель может накапливать состояние при обходе структуры элементов.



Недостатки

Паттерн не оправдан, если иерархия элементов часто меняется.

Может привести к нарушению инкапсуляции элементов.



Связь с другими шаблонами

Команда

Компоновщик

Итератор

ИТОГ

