**ПОСЕТИТЕЛЬ** - позволяет добавлять в программу новые операции, не изменяя классы объектов, над которыми эти операции могут выполняться.

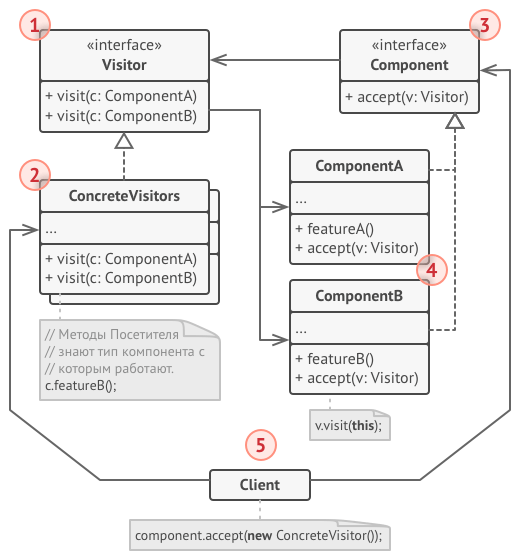
ПРИМЕНЕНИЕ:

- когда нужно выполнить какую-то операцию над всеми элементами сложной структуры объектов, например, деревом

- когда над объектами сложной структуры объектов надо выполнять некоторые не связанные между собой операции, но вы не хотите «засорять» классы такими операциями

- когда новое поведение имеет смысл только для некоторых классов из существующей иерархии

РЕАЛИЗАЦИЯ:

**1. ПОСЕТИТЕЛЬ**: описывает общий интерфейс для всех типов посетителей. Он объявляет набор методов, отличающихся типом входящего параметра, которые нужны для запуска операции для всех типов конкретных компонентов. В языках, поддерживающих перегрузку методов, эти методы могут иметь одинаковые имена, но типы их параметров должны отличаться.

**2. КОНКРЕТНЫЕ ПОСЕТИТЕЛИ**: реализуют какое-то особенное поведение для всех типов компонентов, которые можно подать через методы интерфейса посетителя.

**3. КОМПОНЕНТ**: описывает метод принятия посетителя. Этот метод должен иметь единственный параметр, объявленный с типом интерфейса посетителя.

**4. КОНКРЕТНЫЕ КОМПОНЕНТЫ**: реализуют методы принятия посетителя. Цель этого метода — вызвать тот метод посещения, который соответствует типу этого компонента. Так посетитель узнает, с каким именно компонентом он работает.

**5. КЛИЕНТОМ** зачастую выступает коллекция или сложный составной объект, например, дерево Компоновщика. Зачастую клиент не привязан к конкретным классам компонентов, работая с ними через общий интерфейс компонентов

ПЛЮСЫ:

- упрощает добавление операций, работающих со сложными структурами объектов

- объединяет родственные операции в одном классе

- посетитель может накапливать состояние при обходе структуры компонентов

МИНУСЫ:

- не оправдан, если иерархия компонентов часто меняется

- может привести к нарушению инкапсуляции компонентов