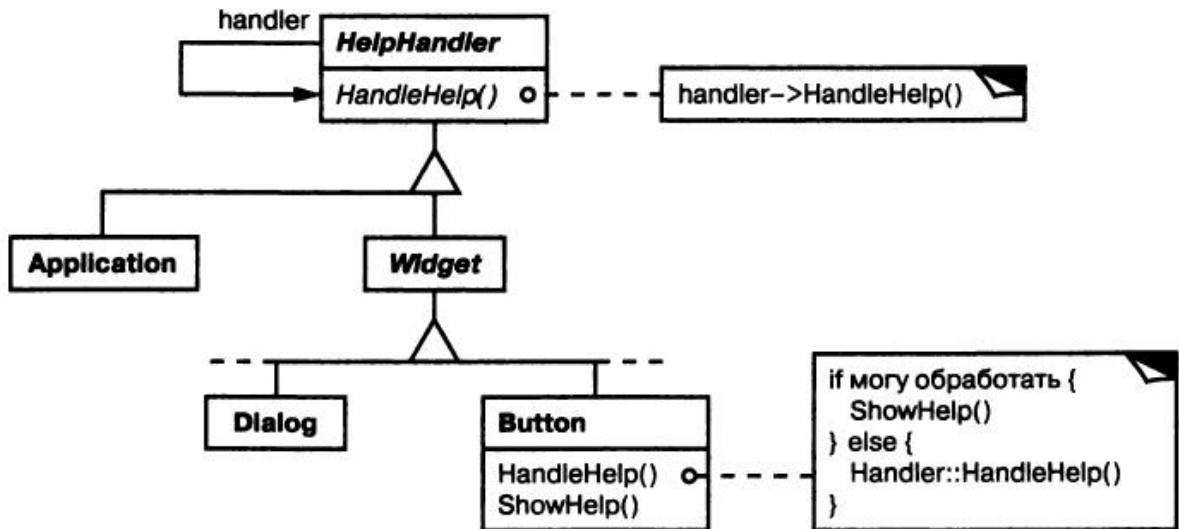


# Паттерны поведения

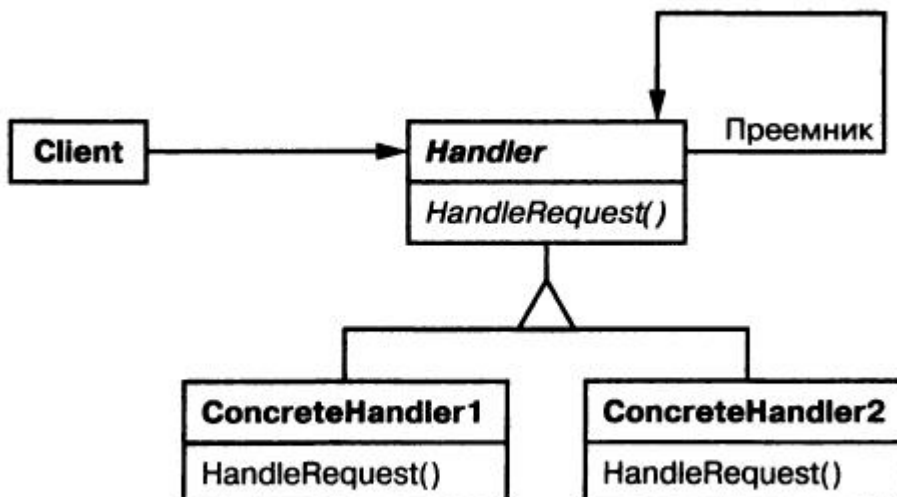
## Chain of Responsibility

Позволяет избежать привязки отправителя запроса к его получателю, давая шанс обработать запрос нескольким объектам. Связывает объекты-получатели в цепочку и передаёт запрос вдоль этой цепочки, пока его не обработают.

Пример: клик на кнопке вызывает помощью кнопки → панели → окна.



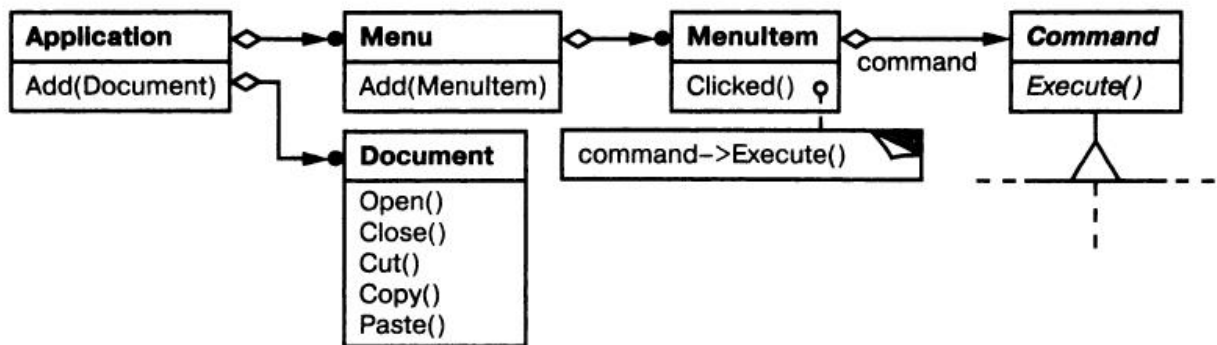
Структура:



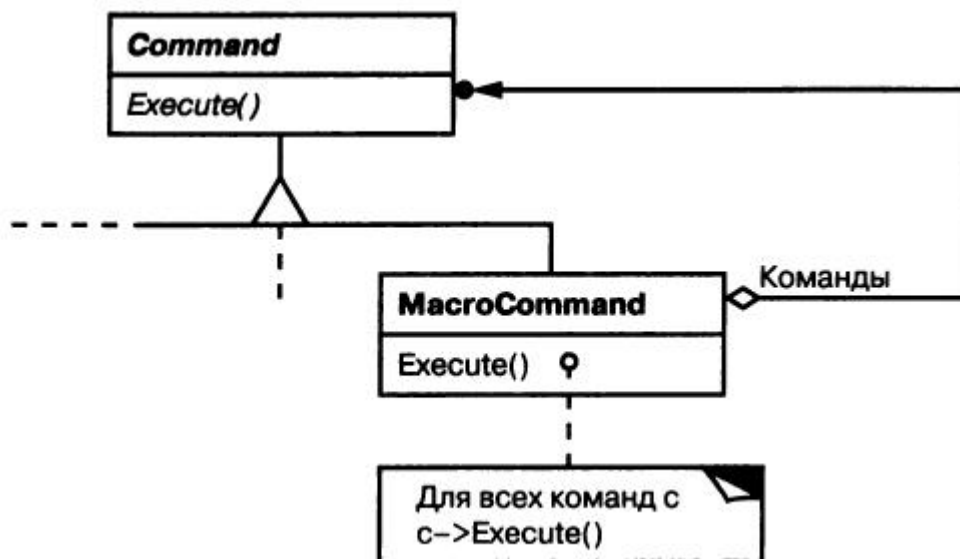
## Command

Инкапсулирует запрос как объект, позволяя тем самым задавать параметры клиентов для обработки соответствующих запросов, ставить запросы в очередь или протоколировать их, а также поддерживать отмену операций.

Пример: пункты в меню, отправляющие запрос неизвестным объектам приложения.



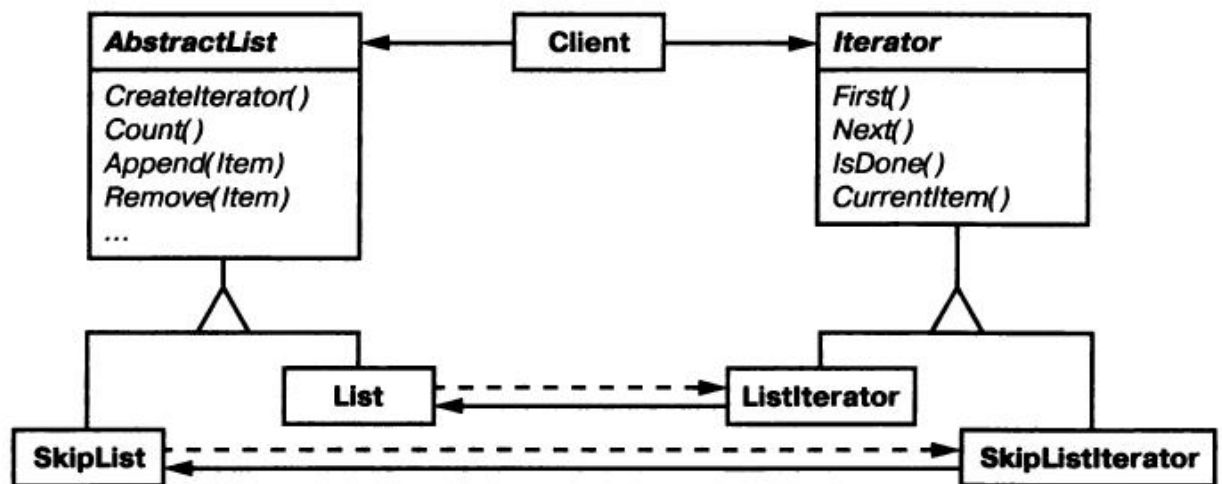
Структура:



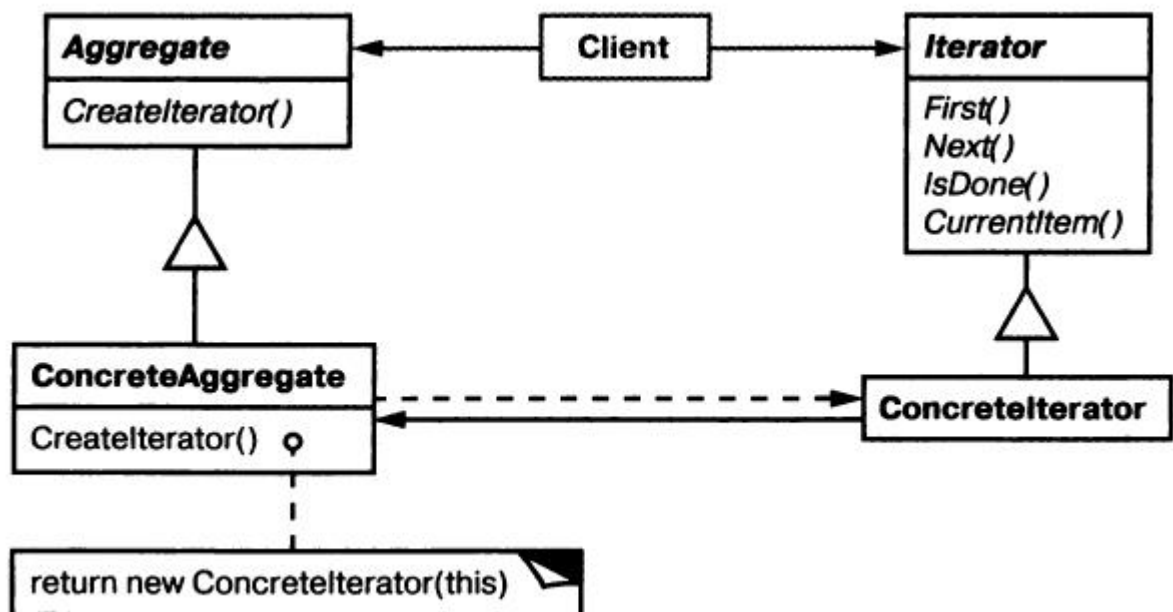
## Iterator

Представляет способ последовательного доступа ко всем элементам составного объекта, не раскрывая его внутреннего представления.

Пример: список пользователей как студентов и как пользователей и итерация по нему.



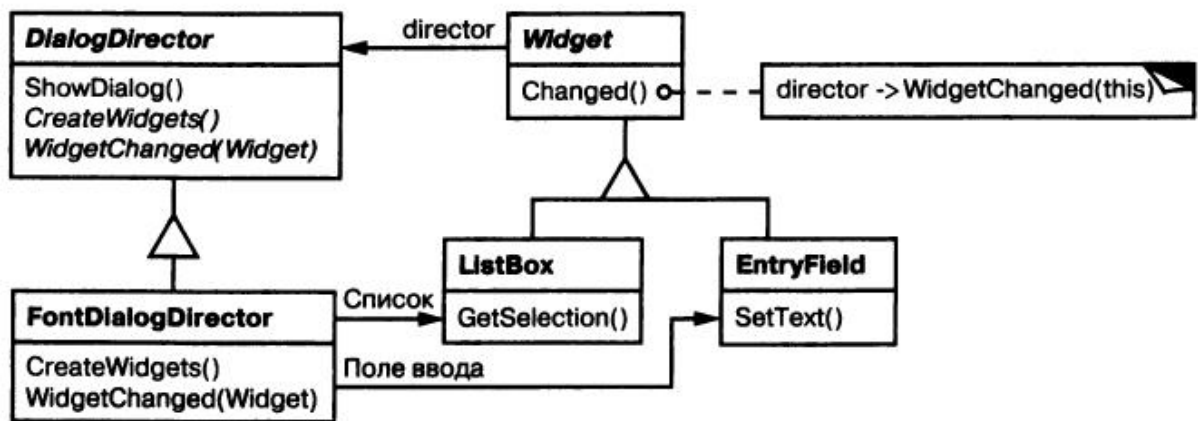
Структура:



## Mediator

Определяет объект, инкапсулирующий способ взаимодействия множества объектов. Обеспечивает слабую связанность системы, избавляя объекты от необходимости явно ссылаться друг на друга и позволяя тем самым независимо изменять взаимодействия между ними.

Пример: форма с разными типами фильтрации.



Структура:

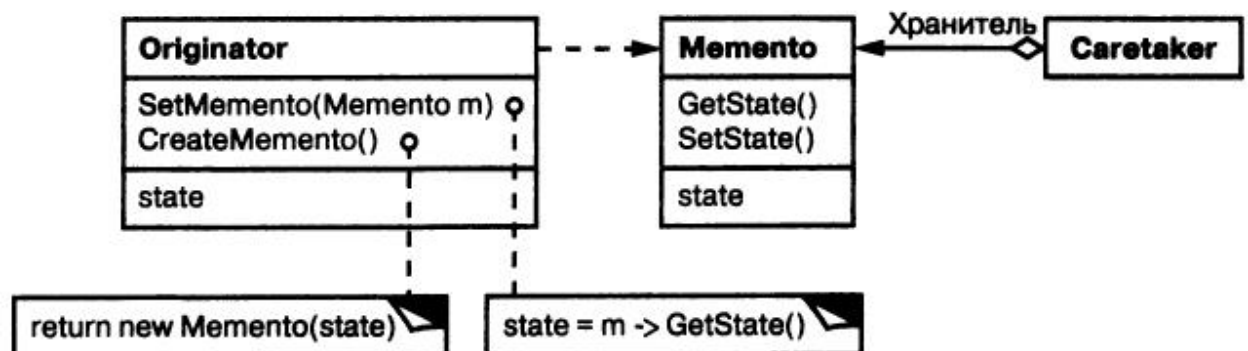


## Memento

Не нарушая инкапсуляции, фиксирует и выносит за пределы объекта его внутреннее состояние так, чтобы позднее можно было восстановить в нём объект.

Пример: undo и redo для текста.

Структура:

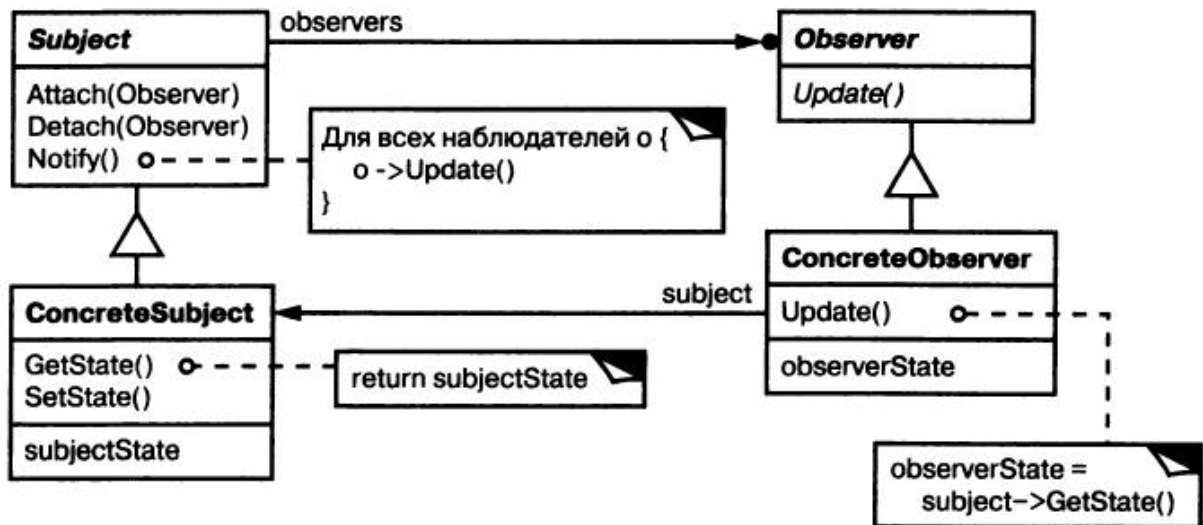


## Observer

Определяет зависимость типа «один ко многим» между объектами таким образом, чтобы при изменении состояния объекта все зависящие от него оповещаются и автоматически обновляются.

Пример: зависимость множества диаграмм в документе от одних данных.

Структура:

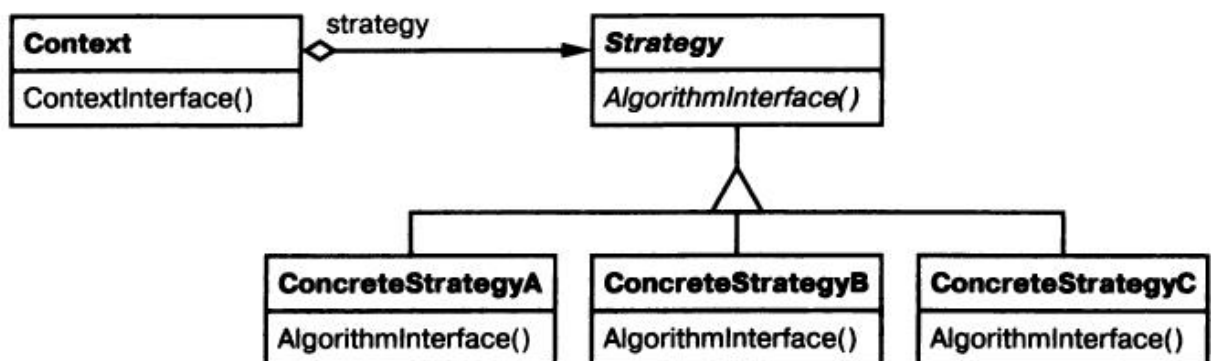


## Strategy

Определяет семейство алгоритмов, инкапсулирует каждый из них и делает взаимозаменяемыми. Позволяет изменить алгоритм независимо от клиентов, которые ими пользуются.

Пример: подмена алгоритма отрисовки элемента.

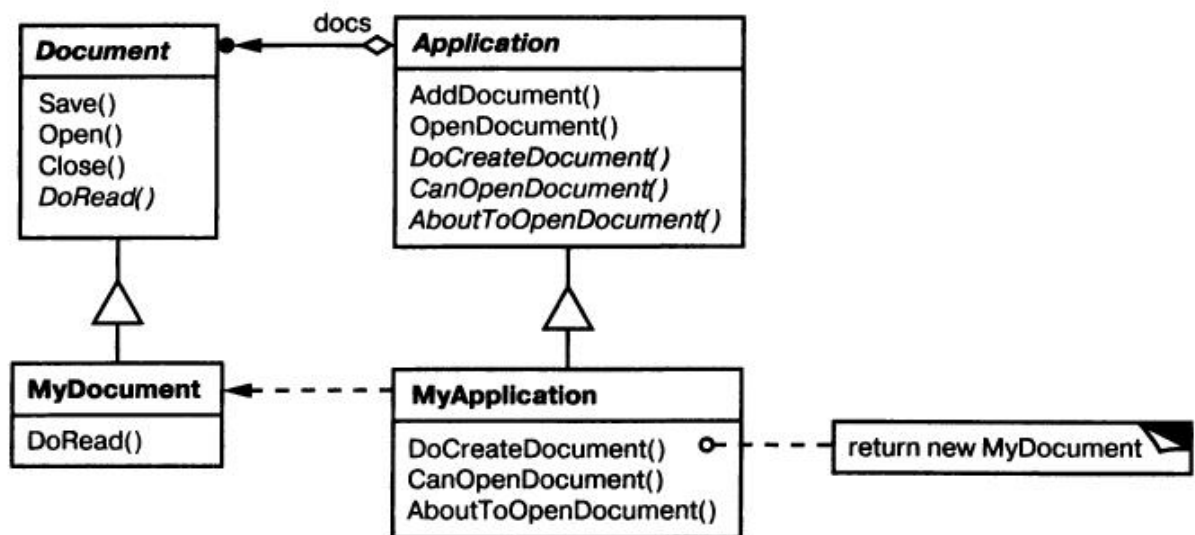
Структура:



## Template Method

Определяет основу алгоритма и позволяет подклассам переопределить некоторые шаги алгоритма, не изменяя его структуру в целом.

Пример: открытие документа в разных приложениях.



Структура:

