

## Определение паттерна Команда

Итак, мы хорошо провели время в кафе Объектвиля, частично реализовали API пульта, а попутно составили довольно хорошее представление о том, как организовано взаимодействие классов и объектов в паттерне Команда. Теперь мы определим паттерн Команда и разберемся с подробностями.

Начнем с официального определения:

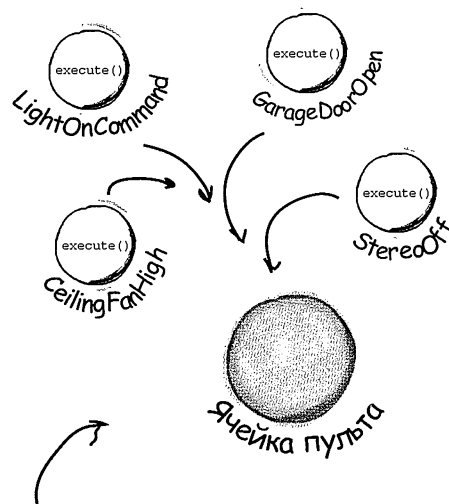
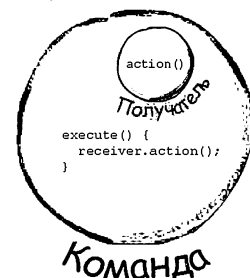
**Паттерн Команда** инкапсулирует запрос в виде объекта, делая возможной параметризацию клиентских объектов с другими запросами, организацию очереди или регистрацию запросов, а также поддержку отмены операций.

Объект команды *инкапсулирует запрос* посредством привязки набора операций к конкретному получателю. Для этого информация об операции и получателе «упаковывается» в объекте с единственным методом `execute()`. При вызове метода `execute()` выполняет операцию с данным получателем. Внешние объекты не знают, какие именно операции выполняются, и с каким получателем; они только знают, что при вызове метода `execute()` их запрос будет выполнен.

Мы уже рассмотрели пару примеров *параметризации объектов* в командах. В кафе Официантка параметризовалась разными заказами. В упрощенном примере с пультом ячейка сначала связывалась с командой включения света, а потом эта команда заменялась командой открытия двери гаража. Для ячейки пульта (как и для Официантки) совершенно неважно, какой объект команды с ней связан (при условии, что он реализует необходимый интерфейс команды).

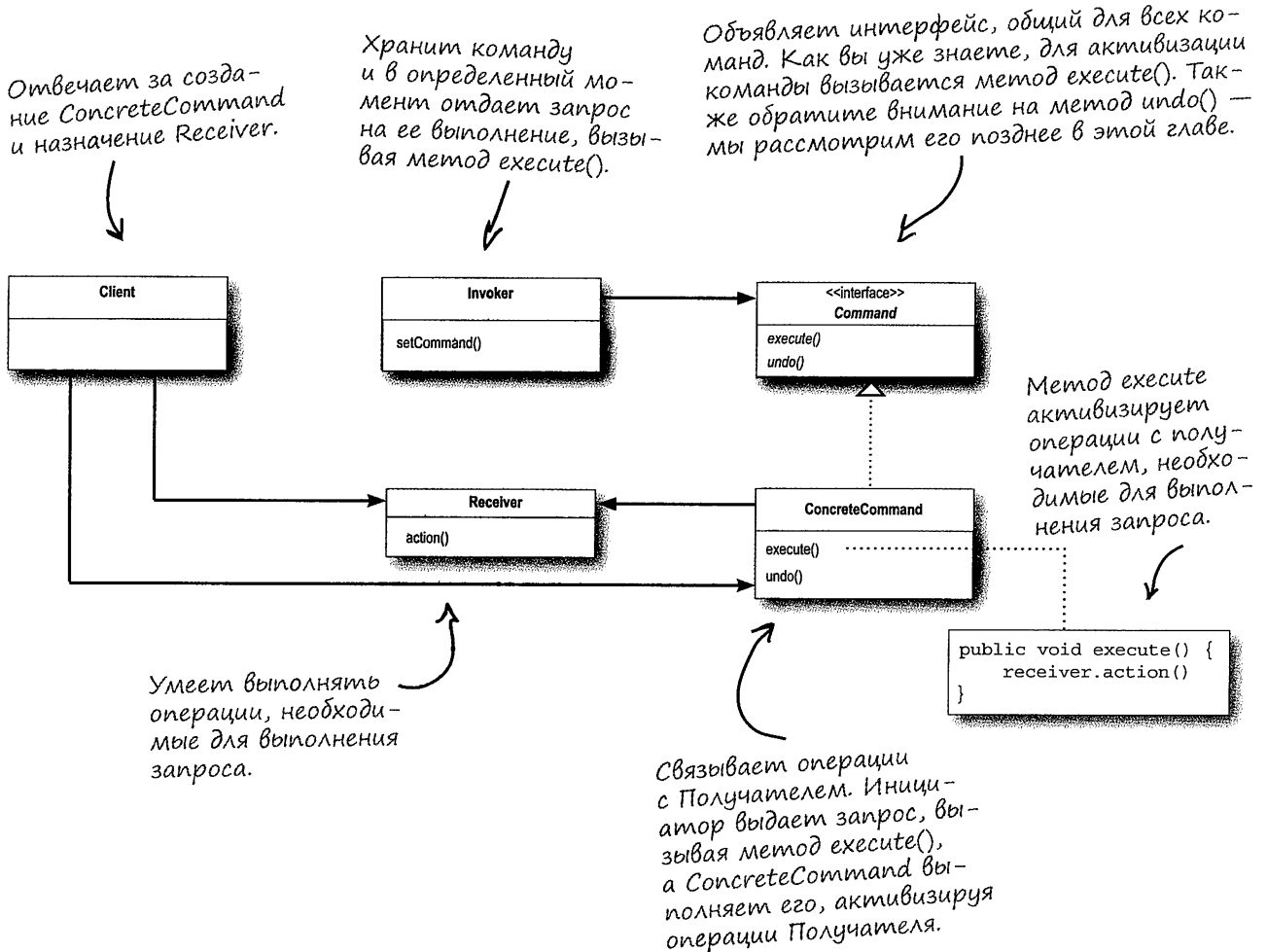
С использованием команд для реализации *очереди и регистрации*, а также *поддержки отмены* мы пока не сталкивались. Не беспокойтесь, это весьма тривиальные расширения базового паттерна Команда. А разобравшись с основной схемой паттерна, вы легко освоите паттерн Метакоманда — механизм создания макропоследовательностей для выполнения серий команд.

Инкапсулированный  
запрос.



Инициатор (например, ячейка пульта) может быть параметризован для разных запросов.

## Определение паттерна Команда: диаграмма классов



Каким образом архитектура паттерна Команда способствует отделению инициатора от получателя запроса?