



Новые инструменты

Ваш инструментарий постепенно расширяется! В этой главе он пополнился паттерном, позволяющим инкапсулировать методы в объектах Команды: сохранять их, передавать и активизировать по мере необходимости.

Принципы

Инкапсулируйте то, что изменяется.
Предпочитайте композицию наследованию.
Программируйте на уровне интерфейсов.
Стремитесь к слабой связанности взаимодействующих объектов.
Классы должны быть открыты для расширения, но закрыты для изменения.
Код должен зависеть от абстракций, а не от конкретных классов.

Принципы
инкапсуляция
морфизм
следование

Если вам требуется отделить объект, выдающий запросы, от объектов, которые умеют эти запросы выполнять, — используйте паттерн Команда.

Паттерны

Стратегия — определяет

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

КЛЮЧЕВЫЕ МОМЕНТЫ



- Паттерн Команда отделяет объект, выдающий запросы, от объекта, который умеет эти запросы выполнять.
- Объект команды инкапсулирует получателя с операцией (или набором операций).
- Инициатор вызывает метод `execute()` объекта команды, что приводит к выполнению соответствующих операций с получателем.
- Возможна параметризация инициаторов командами (даже динамическая во время выполнения).
- Команды могут поддерживать механизм отмены, восстанавливающий объект в состоянии до последнего вызова метода `execute()`.
- Макрокоманды — простое расширение паттерна Команда, позволяющее выполнять цепочки из нескольких команд. В них также легко реализуется механизм отмены.
- На практике нередко встречаются «умные» объекты команд, которые реализуют запрос самостоятельно вместо его делегирования получателю.
- Команды также могут использоваться для реализации систем регистрации команд и поддержки транзакций.