

Поведенческие паттерны проектирования

Основные понятия

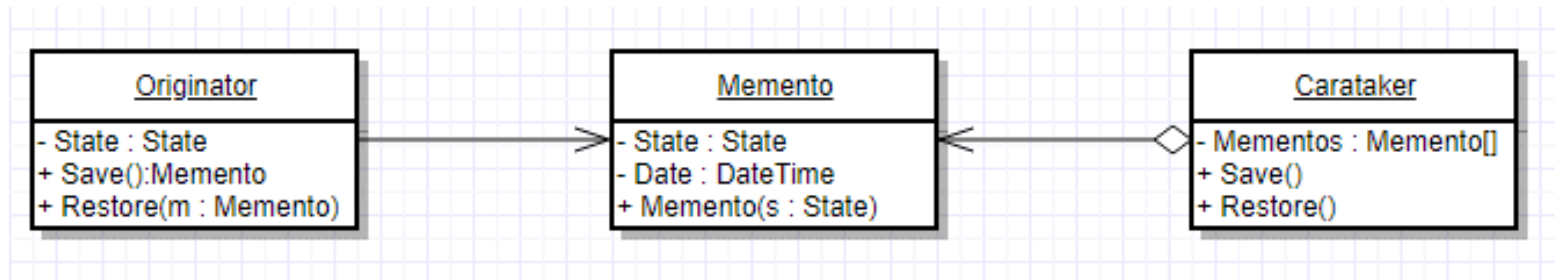
Поведенческие паттерны предназначены для обеспечения эффективного взаимодействия между объектами

Основные представители:

- Хранитель
- Наблюдатель
- Состояние
- Посетитель
- Посредник
- Итератор
- Команда и т.д.

Паттерн «хранитель»

- делает снимки состояния объектов, скрывая подробности их реализации. По снимкам можно восстанавливать состояние объектов

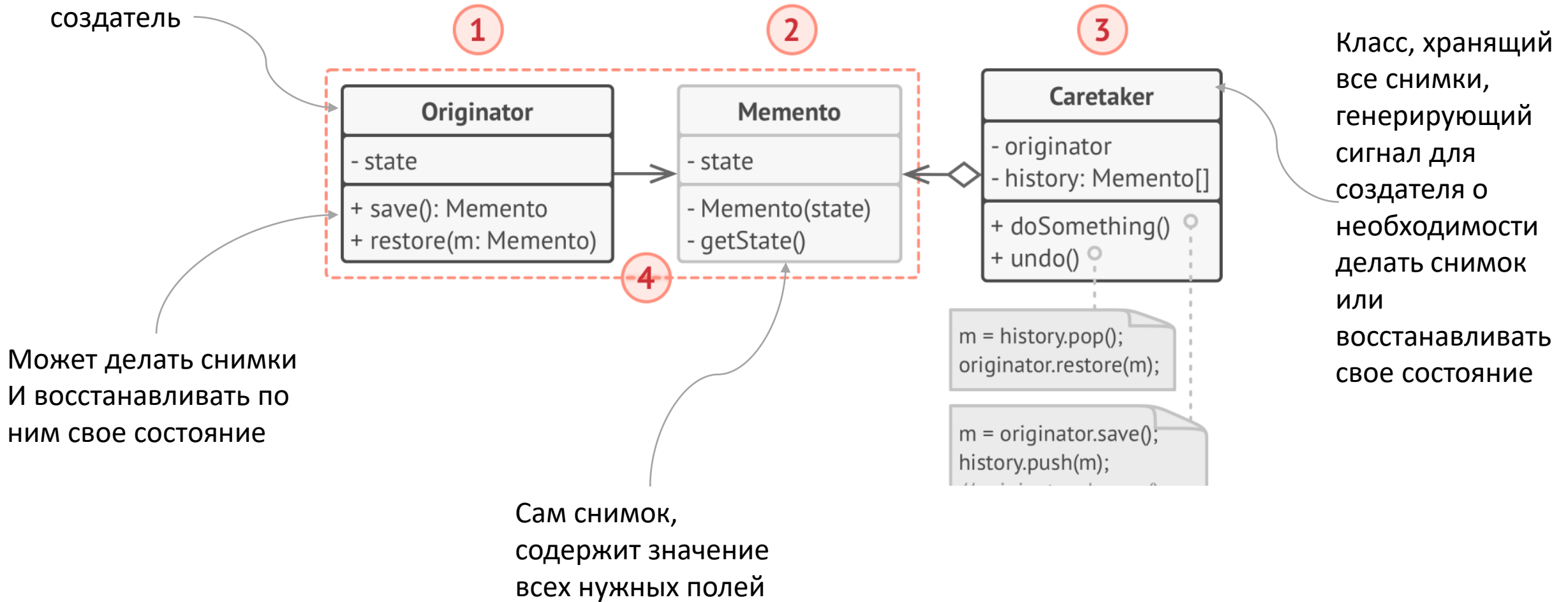


- Originator — основной объект, состояние которого требуется хранить и восстанавливать
- Memento — снимок состояния основного объекта.
- Carataker — хранилище снимков.

Особенности:

- Объект сам делает снимок своего состояния, не нарушая правил инкапсуляции
- Снимок полностью доступен только своему создателю
- Используются вложенные классы

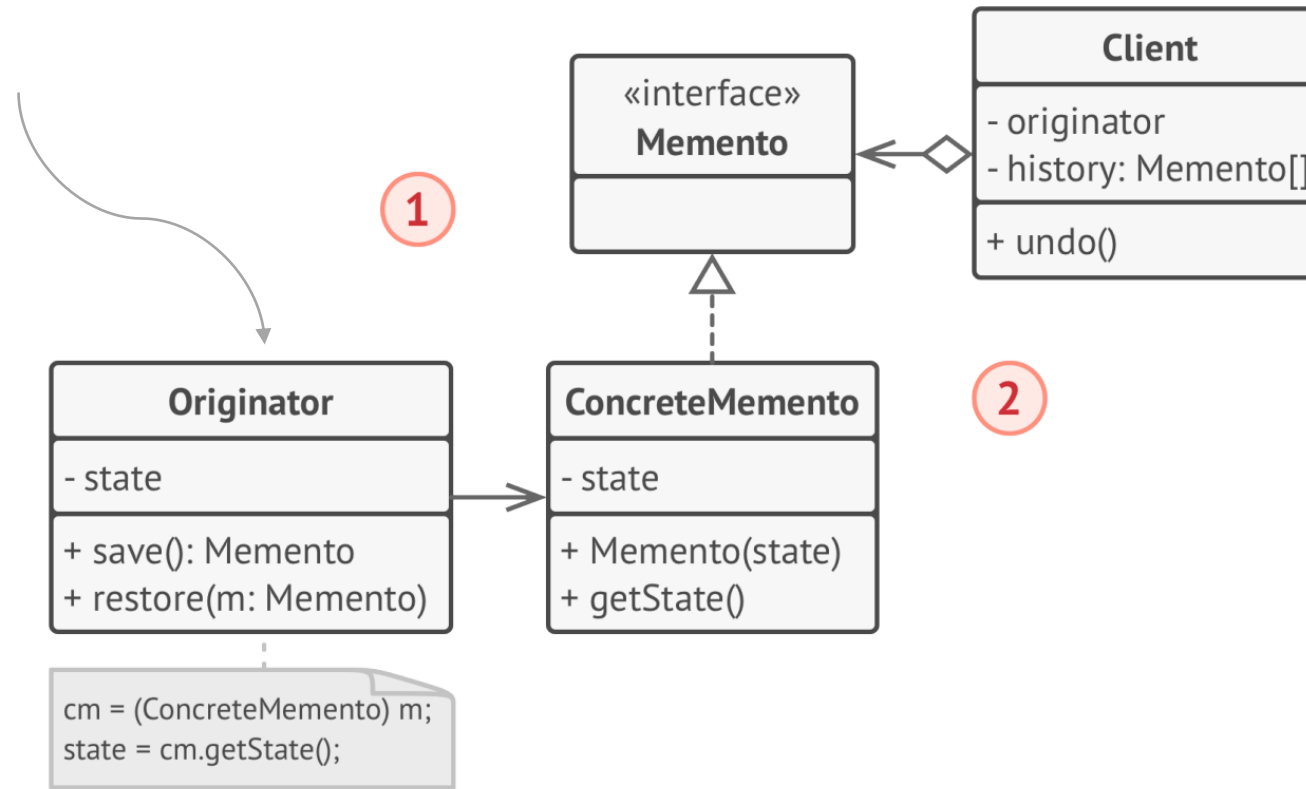
Классическая организация паттерна «хранитель»



- Memento является внутренним классом по отношению к классу создатель => имеет полным доступ к его состоянию
- Caretaker доступа к внутреннему состоянию Originator не имеет

Организация паттерна «хранитель» через интерфейс

Создатель работает с классом снимка, давая ему доступ к своему состоянию



Хранитель снимков работает с ограниченным интерфейсом снимка

Создатель так же имеет полный доступ к снимку, а хранитель нет

Области применения

- При необходимости сохранять состояние объекта или его части для дальнейшего восстановления
- При необходимости скрыть детали реализации класса при прямом доступе от других объектов

Преимущества и недостатки

+

- Не нарушает инкапсуляции исходного объекта.
- Упрощает структуру исходного объекта.

-

- Потребность в памяти
- Не во всея языках возможна полноценная реализация

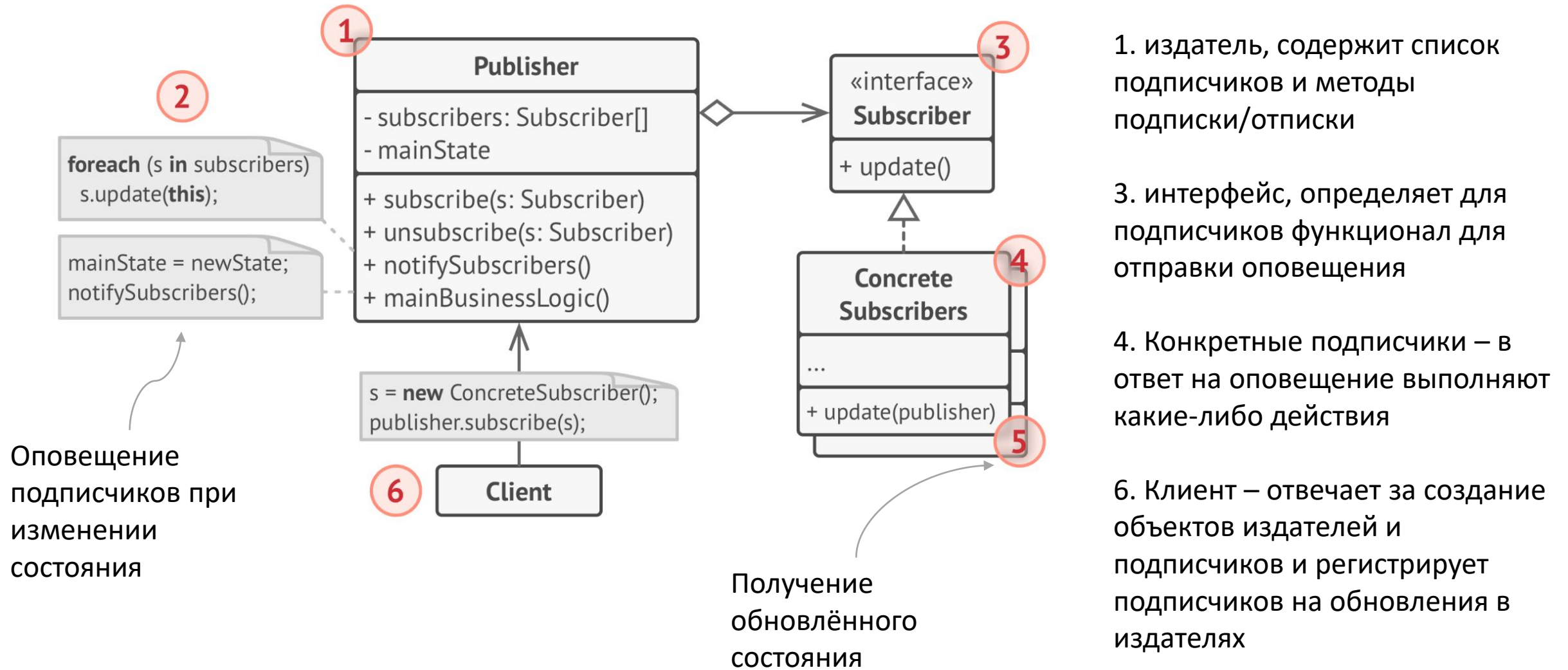
Паттерн «наблюдатель»

- организует механизм подписки, благодаря которой одни объекты могут следить и реагировать на события, происходящие в других объектах.

Особенности:

- Издатель – это объекты, содержащее интересующее состояние
- Подписчики – объекты, интересующиеся состоянием издателя
- Подписчики самостоятельно добавляют себя в список издателя и удаляются из него
- При активации интересующего состояния, издатель рассылает уведомление всем подписчикам

Организация паттерна «наблюдатель»



Области применения

- При изменении объекта требуется производить действия в других, но их список заранее определить невозможно
- При необходимости отслеживать состояние других объектов, но интересует только определенное состояние

Преимущества и недостатки

+

- Издатель не зависит от конкретных классов подписчиков.
- Вы можете подписывать и отписывать получателей на лету.
- Поддерживает принцип открытости/закрытости.

-

- Оповещение происходит в случайном порядке