Java Design Patterns

Memento

Тема

Memento

Суть паттерна

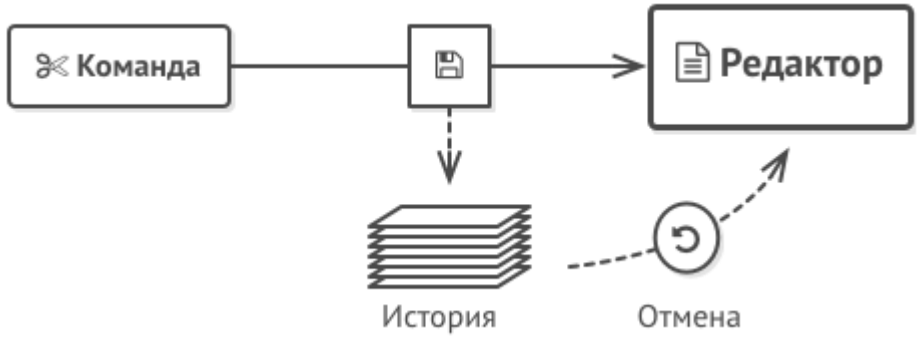
Снимок

Снимок — это поведенческий паттерн проектирования, который позволяет делать снимки состояния объектов, не раскрывая подробностей их реализации. Затем снимки можно использовать, чтобы восстановить прошлое состояние объектов.

Проблема

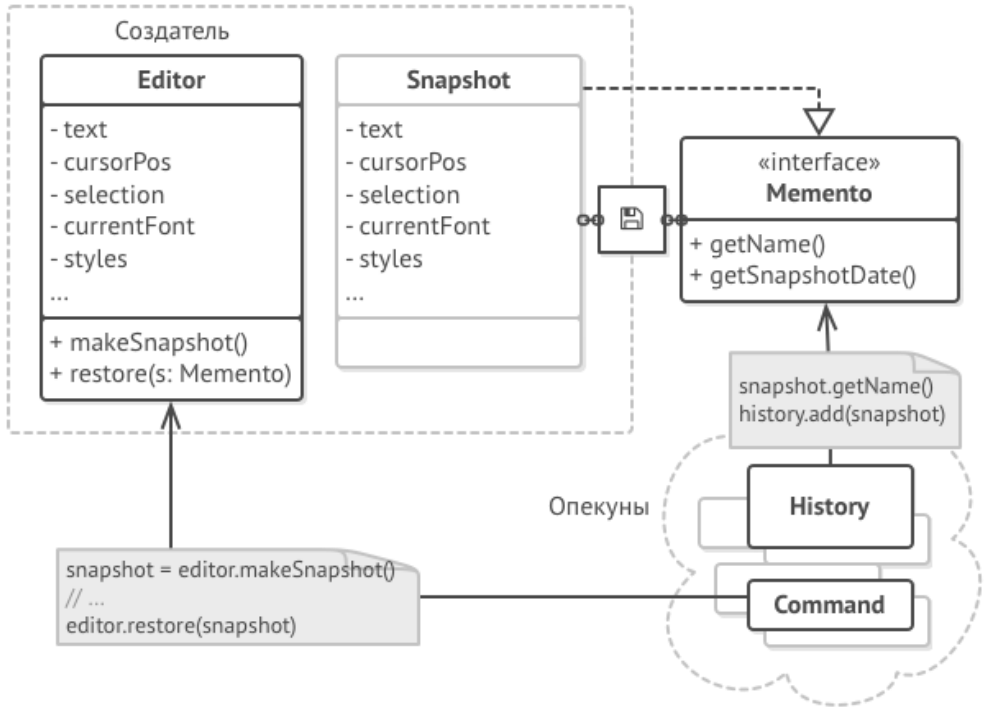
Постановка задачи

Предположим, что вы пишете текстовый редактор. Помимо обычного редактирования, ваш редактор позволяет менять форматирование текста, вставлять картинки и прочее.



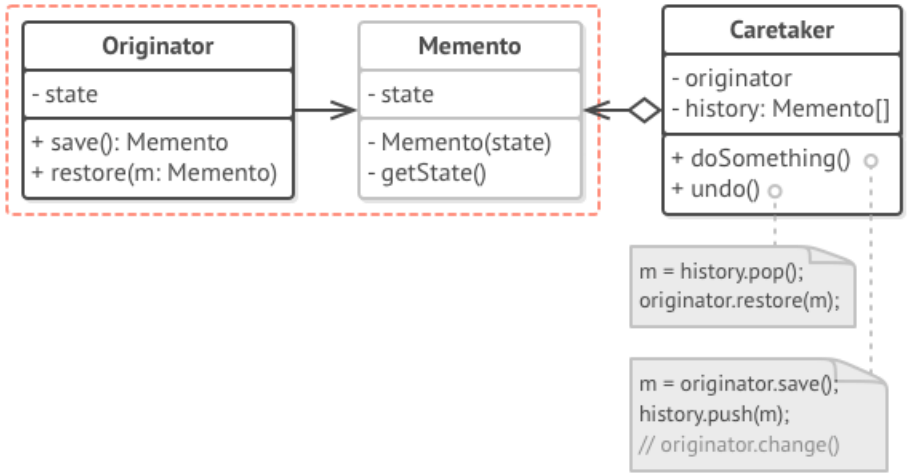
Решение задачи

Все проблемы, описанные выше, возникают из-за нарушения инкапсуляции. Это когда одни объекты пытаются сделать работу за других, влезая в их приватную зону, чтобы собрать необходимые для операции данные.



Структура паттерна

1. Создатель делает снимки своего состояния по запросу, а также воспроизводит прошлое состояние, если подать в него готовый снимок.
2. Снимок — это простой объект данных, содержащий состояние создателя. Надёжней всего сделать объекты снимков неизменяемыми и передавать в них состояние только через конструктор.
3. Опекун должен знать, когда и зачем делать снимок создателя, а также когда его нужно восстанавливать.
4. В этой реализации снимок — это внутренний класс по отношению к классу создателя, поэтому тот имеет полный доступ к его полям и методам, несмотря на то, что они объявлены приватными. Опекун же не имеет доступа ни к состоянию, ни к методам снимков и может всего лишь хранить ссылки на эти объекты.



Применимость

Применение паттерна

1. Когда вам нужно сохранять мгновенный снимок состояния объекта (или его части), чтобы впоследствии объект можно было восстановить в том же состоянии.
2. Когда прямое получение состояния объекта раскрывает детали его реализации и нарушает инкапсуляцию.

Шаги реализации

Алгоритм реализации паттерна

1. Определите класс создателя, объекты которого должны создавать снимки своего состояния.
2. Создайте класс снимка и опишите в нём все те же поля, которые имеются в оригинальном классе-создателе.
3. Сделайте объекты снимков неизменяемыми. Они должны получать начальные значения только один раз, через свой конструктор.
4. Если ваш язык программирования это позволяет, сделайте класс снимка вложенным в класс создателя.
5. Добавьте в класс создателя метод получения снимков. Создатель должен создавать новые объекты снимков, передавая значения своих полей через конструктор.
6. Добавьте в класс создателя метод восстановления из снимка. Что касается привязки к типам, руководствуйтесь той же логикой, что и в пункте 4.
7. Опекуны, будь то история операций, объекты команд или нечто иное, должны знать о том, когда запрашивать снимки у создателя, где их хранить, и когда восстанавливать.
8. Связь опекунов с создателями можно перенести внутрь снимков. В этом случае каждый снимок будет привязан к своему создателю и должен будет сам восстанавливать его состояние. Но это будет работать либо если классы снимков вложены в классы создателей, либо если создатели имеют сеттеры для установки значений своих полей.

Преимущества и недостатки

Плюсы и недостатки

Плюсы:

- Не нарушает инкапсуляции исходного объекта.
- Упрощает структуру исходного объекта. Ему не нужно хранить историю версий своего состояния.

Минусы:

- Требует много памяти, если клиенты слишком часто создают снимки.
- Может повлечь дополнительные издержки памяти, если объекты, хранящие историю, не освобождают ресурсы, занятые устаревшими снимками.
- В некоторых языках (например, PHP, Python, JavaScript) сложно гарантировать, чтобы только исходный объект имел доступ к состоянию снимка.

Отношение с другими паттернами

- Команду и Снимок можно использовать сообща для реализации отмены операций. В этом случае объекты команд будут отображать выполненные действие над объектом, снимки — хранить копию состояния этого объекта до того, как команда была выполнена.
- Снимок можно использовать вместе с Итератором, чтобы сохранить текущее состояние обхода структуры данных и вернуться к нему в будущем, если потребуется.
- Снимок иногда можно заменить Прототипом, если объект, чьё состояние требуется сохранять в истории, довольно простой, не имеет активных ссылок на внешние ресурсы, либо их можно легко восстановить.

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association