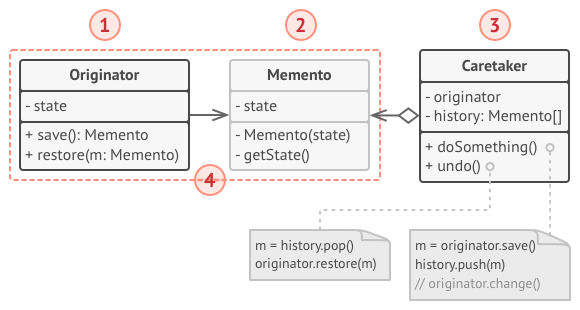
**ХРАНИТЕЛЬ** - позволяет сохранять и восстанавливать прошлые состояния объектов, не раскрывая подробностей их реализации.

ПРИМЕНЕНИЕ:

- когда нужно сохранять мгновенные снимки состояния объекта (или его части), чтобы впоследствии объект можно было восстановить в том же состоянии

- когда прямое получение состояния объекта раскрывает приватные детали его реализации, нарушая инкапсуляцию

РЕАЛИЗАЦИЯ:

**1. СОЗДАТЕЛЬ**: может производить снимки своего состояния, а также воспроизводить прошлое состояние, если подать в него готовый снимок.

**2. СНИМОК**: простой объект данных, содержащий состояние создателя. Надёжнее всего сделать объекты снимков неизменяемыми, передавая в них состояние только через конструктор.

**3. ОПЕКУН**: должен знать, когда делать снимок создателя и когда его нужно восстанавливать. Может хранить историю прошлых состояний создателя в виде стека из снимков. Когда понадобится отменить выполненную операцию, он возьмёт «верхний» снимок из стека и передаст его создателю для восстановления.

**4.** В данной реализации снимок — это внутренний класс по отношению к классу создателя. Именно поэтому он имеет полный доступ к полям и методам создателя, даже приватным. С другой стороны, опекун не имеет доступа ни к состоянию, ни к методам снимков и может всего лишь хранить ссылки на эти объекты

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 снимкая