SDET Course

Design Patterns - Memento

3 Types of Design Patterns

- Creational
  - Singleton
  - Builder
  - Prototype
  - Factory Method
  - Abstract Factory
- Structural
  - Adapter
  - Composite
  - Proxy
  - Flyweight
  - Bridge
  - Facade
  - Decorator
- Behavioral
  - Strategy
  - Observer
  - Command
  - **Memento**
  - State
  - Template Method
  - Mediator
  - Chain of Responsibility
  - Interpreter
  - Visitor
  - Iterator

Agenda

- Description
- Diagram
- Code sample (Java)
- Use cases

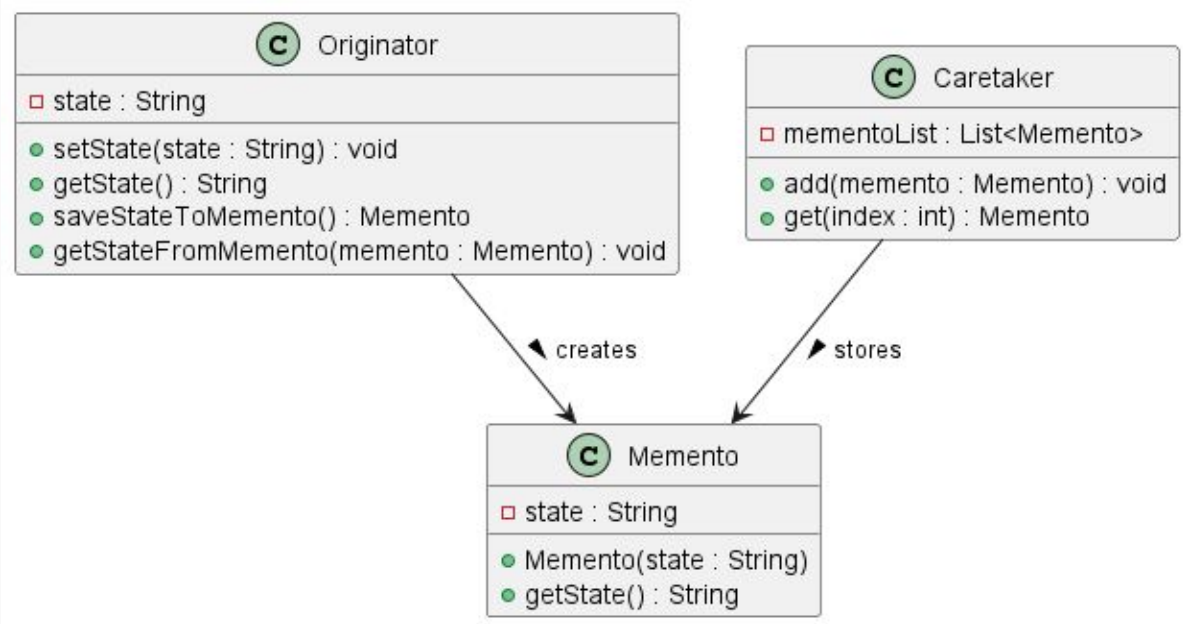
VERISOFT

Descripti
on

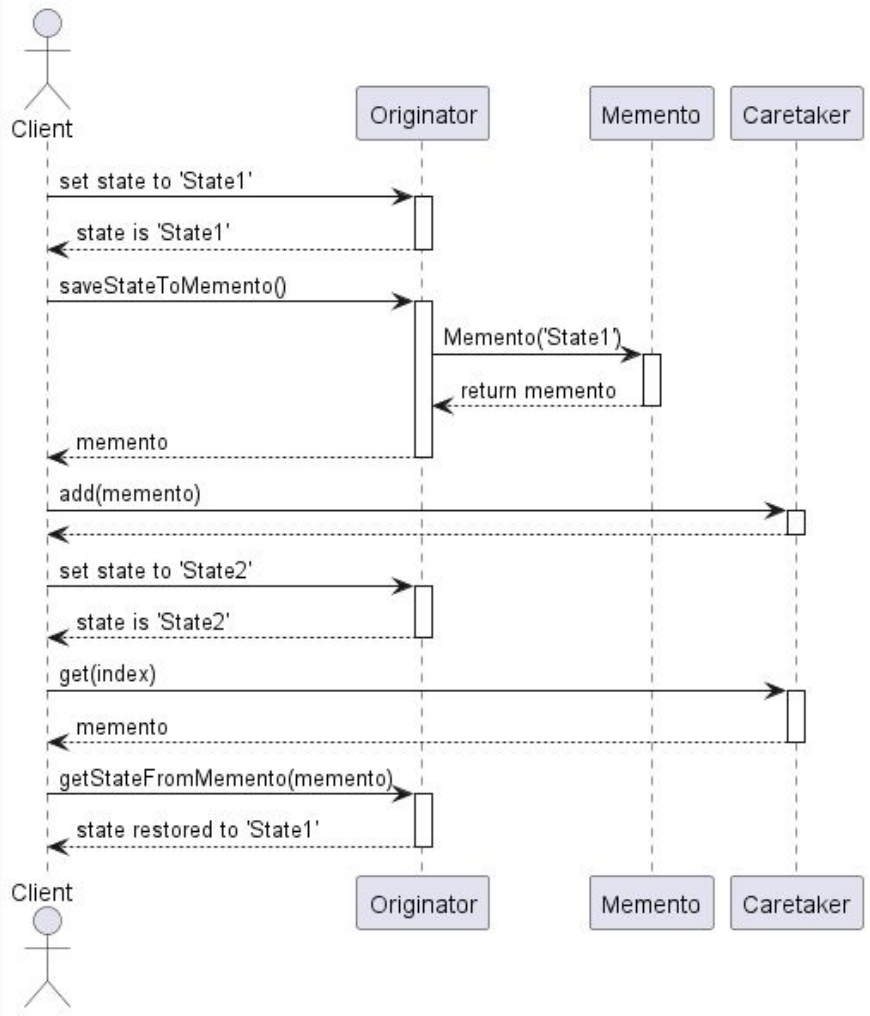
Description

The Memento Design Pattern is a behavioral design pattern that provides the ability to restore an object to its previous state without revealing the details of its implementation. This pattern is particularly useful in scenarios where direct access to the object's internal state is not desirable or violates encapsulation, such as in undo mechanisms or snapshot-style save points. The pattern involves three key roles: the Originator, which is the object whose state needs to be saved and restored; the Memento, which is a lightweight representation of the Originator's state at a particular time; and the Caretaker, which requests a save from the Originator and holds onto the Memento until a rollback is required. This separation of concerns allows for the internal state of the Originator to be saved and restored externally by the Caretaker, without compromising the principles of encapsulation and internal structure of the Originator.

Class Diagram



Sequence Diagram



Code Sample

VERISOFT

Use cases

- General
  - State Management in Workflows
  - Undo Mechanisms
  - Version Control

- In Test Automation
  - Multi-state Scenario Testing
    - Isolation Testing
    - Pathway Testing
    - Regression Testing

Happy Coding