Memento Pattern

Resources

- 41. All Behavioral Design Patterns | Strategy, Observer, State, Template, Command, Visitor, Memento
- 38. Memento Design Pattern explanation | LLD System Design | Design pattern explanation in Java

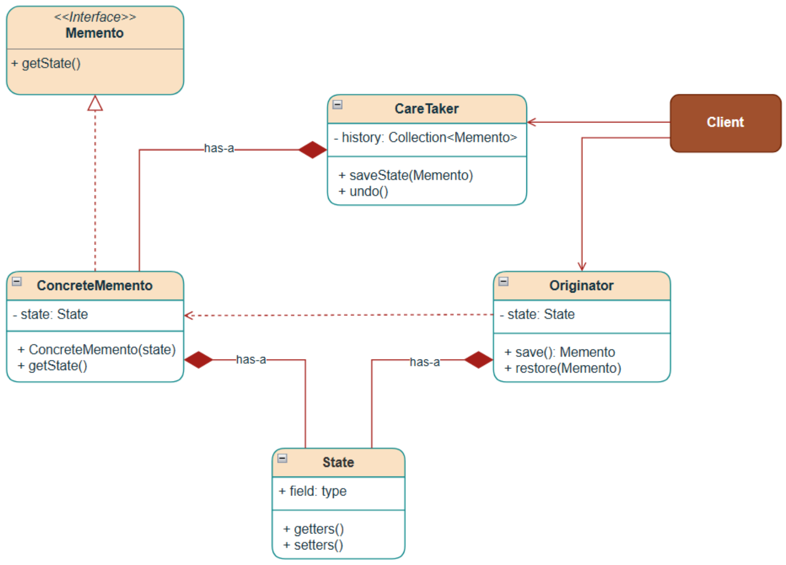
Definition

> ***The Memento pattern is a behavioral design pattern that allows you to capture and restore an object's internal state without violating encapsulation. It's widely used for implementing undo/redo functionality and checkpoints or versioning(systems that require restoring previous states when requested) in your application.***

Memento pattern is made up 3 types of classes, each responsible for particluar action:

1. **Originator**
   - It represents the object, for which state need to be saved and restored.
   - Expose methods to save and restore its state using Memento object.
2. **Memento**
   - It represents an Object which holds the state of the Originator.
3. **Caretaker**
   - Manages the list of States (i.e. list of Memento Objects).

Class Diagram



Structure of Memento Pattern

Understanding the structure of Memento Pattern using an example of IDE application's configuration that can be saved and restored:

1. **Memento (`ConfigurationMemento`)**:
   - Immutable object that stores the configuration state.
   - Only the Originator can access its internal data.
2. **Originator (`ApplicationConfiguration`)**:
   - Contains the actual configuration that changes over time.
   - `save()` method → creates a memento capturing current state
   - `restore()` method → brings back a previous state from a memento.
3. **Caretaker (`ConfigurationManager`)**:
   - Manages a history of mementos for undo functionality.
   - Doesn't know about the internal structure of mementos.
   - Provides clean interface to save and restore states.

Implementation

Here is example of an IDE application's configuration that can be saved and restored:

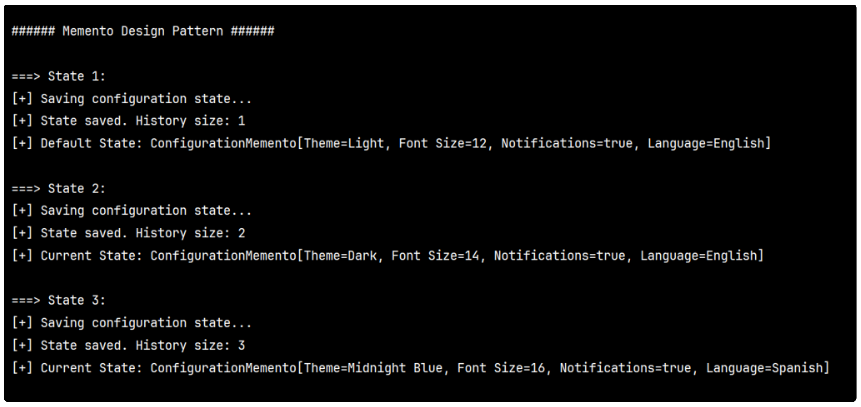
```java
// Originator class - creates and restores from mementos
public class ApplicationConfiguration {
    // State
    private String theme;
    private int fontSize;
    private boolean notificationsEnabled;
    private String language;

    public ApplicationConfiguration(String theme, int fontSize,
                                    boolean notificationsEnabled,
String language) {
        this.theme = theme;
        this.fontSize = fontSize;
        this.notificationsEnabled = notificationsEnabled;
        this.language = language;
    }

    // Create a memento with current state
    public ConfigurationMemento save() {
        System.out.println("[+] Saving configuration state...");
        return new ConfigurationMemento(theme, fontSize, notificationsEnabled, language);
    }

    // Restore state from memento
    public void restore(ConfigurationMemento memento) {
        this.theme = memento.getTheme();
        this.fontSize = memento.getFontSize();
        this.notificationsEnabled = memento.isNotificationsEnabled();
        this.language = memento.getLanguage();
        System.out.println("[+] Restored Previous Configuration State");
    }

    // Setters to modify state
    public void setTheme(String theme) {
        this.theme = theme;
    }

    public void setFontSize(int fontSize) {
        this.fontSize = fontSize;
    }

    public void setNotificationsEnabled(boolean enabled) {
        this.notificationsEnabled = enabled;
    }

    public void setLanguage(String language) {
        this.language = language;
    }

    @Override
    public String toString() {
        return String.format("Configuration[Theme=%s, Font Size=%d, Notifications=%b, Language=%s]",
                theme, fontSize, notificationsEnabled, language);
    }
}
```

```java
// Caretaker class - manages mementos
public class ConfigurationManager {
    private final Stack<ConfigurationMemento> history = new Stack<>();

    public void saveState(ApplicationConfiguration appConfig) {
        ConfigurationMemento configurationMemento = appConfig.save(); // creates a memento with current state
        history.push(configurationMemento); // stores the memento in the history
        System.out.println("[+] State saved. History size: " + history.size());
        System.out.println(history.size() == 1 ? "[+] Default State: " + configurationMemento : "[+] Current State: " + configurationMemento);
    }

    public void undo(ApplicationConfiguration appConfig) {
        if (history.size() > 1) {
            history.pop(); // removes and returns the last saved state
            ConfigurationMemento mementoToBeRestored = history.peek(); // returns the previous state to be restored
            appConfig.restore(mementoToBeRestored); // restores the application configuration to the previous saved state
            System.out.println("[+] Undo performed. History size: " + history.size());
            System.out.println(history.size() == 1 ? "[+] Default State: " + mementoToBeRestored : "[+] Current State: " + mementoToBeRestored);
        } else {
            System.out.println("[+] No more states to undo!");
            System.out.println("[+] Default State: " + history.peek());
        }
    }
}
```

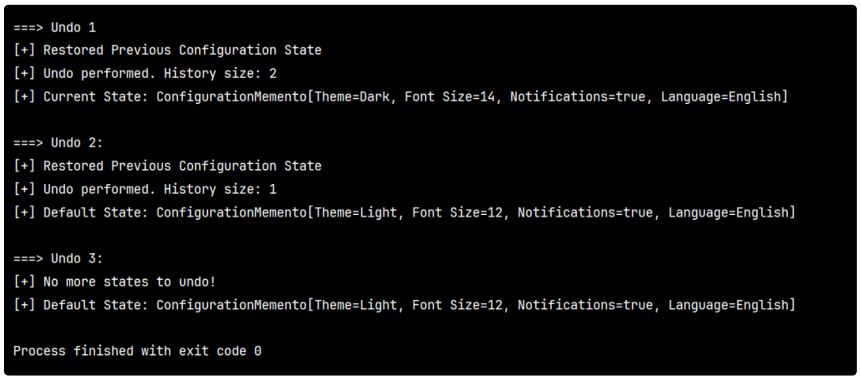
```java
// Memento class - stores the state
public class ConfigurationMemento {
    private final String theme;
    private final int fontSize;
    private final boolean notificationsEnabled;
    private final String language;

    public ConfigurationMemento(String theme, int fontSize,
                                boolean notificationsEnabled, String language) {
        this.theme = theme;
        this.fontSize = fontSize;
        this.notificationsEnabled = notificationsEnabled;
        this.language = language;
    }

    // Getters for restoration
    String getTheme() {
        return theme;
    }

    int getFontSize() {
        return fontSize;
    }

    boolean isNotificationsEnabled() {
        return notificationsEnabled;
    }

    String getLanguage() {
        return language;
    }

    @Override
    public String toString() {
        return String.format("ConfigurationMemento[Theme=%s, Font Size=%d, Notifications=%b, Language=%s]",
                theme, fontSize, notificationsEnabled, language);
    }
}
```

```java
// Demo Usage
public class MementoDemo {
    public static void main(String[] args) {
        System.out.println("\n###### Memento Design Pattern ######");

        // Create Originator Object
        ApplicationConfiguration appConfig = new ApplicationConfiguration(
                "Light", 12, true, "English"
        );

        // Create Caretaker Object
        ConfigurationManager configurationManager = new ConfigurationManager();

        // Default State
        System.out.println("\n===> State 1: ");
        configurationManager.saveState(appConfig); // Default State

        // State 2
        appConfig.setTheme("Dark");
        appConfig.setFontSize(14);
        System.out.println("\n===> State 2: ");
        configurationManager.saveState(appConfig); // Creates a memento and stores it in history

        // State 3

        appConfig.setTheme("Midnight Blue");
        appConfig.setFontSize(16);
        appConfig.setLanguage("Spanish");
        System.out.println("\n===> State 3: ");
        configurationManager.saveState(appConfig); // Creates a memento and stores it in history

        // Undo 1
        System.out.println("\n===> Undo 1 ");
        configurationManager.undo(appConfig); // Restores the application configuration to the previous saved state

        // Undo 2
        System.out.println("\n===> Undo 2: ");
        configurationManager.undo(appConfig); // Restores the application configuration to the previous saved state

        // Undo 3: Try to undo when no history
        System.out.println("\n===> Undo 3: ");
        configurationManager.undo(appConfig); // Default State
    }
}
```

Output



```
###### Memento Design Pattern ######

===> State 1:
[+] Saving configuration state...
[+] State saved. History size: 1
[+] Default State: ConfigurationMemento[Theme=Light, Font Size=12, Notifications=true, Language=English]

===> State 2:
[+] Saving configuration state...
[+] State saved. History size: 2
[+] Current State: ConfigurationMemento[Theme=Dark, Font Size=14, Notifications=true, Language=English]

===> State 3:
[+] Saving configuration state...
[+] State saved. History size: 3
[+] Current State: ConfigurationMemento[Theme=Midnight Blue, Font Size=16, Notifications=true, Language=Spanish]
```



```
===> Undo 1
[+] Restored Previous Configuration State
[+] Undo performed. History size: 2
[+] Current State: ConfigurationMemento[Theme=Dark, Font Size=14, Notifications=true, Language=English]

===> Undo 2:
[+] Restored Previous Configuration State
[+] Undo performed. History size: 1
[+] Default State: ConfigurationMemento[Theme=Light, Font Size=12, Notifications=true, Language=English]

===> Undo 3:
[+] No more states to undo!
[+] Default State: ConfigurationMemento[Theme=Light, Font Size=12, Notifications=true, Language=English]

Process finished with exit code 0
```