**Memento Design Pattern**

**Assignment - 1**

Name - Heet Dobariya Roll No. - 22BCP177 Group - G5

* **Memento Design Pattern :**

The Memento design pattern is a behavioral pattern used in software engineering. It allows you to capture and externalize an object's internal state so that the object can be restored to that state later, without violating encapsulation. This pattern is especially useful when you need to implement undo mechanisms or save and restore functionality in an application.

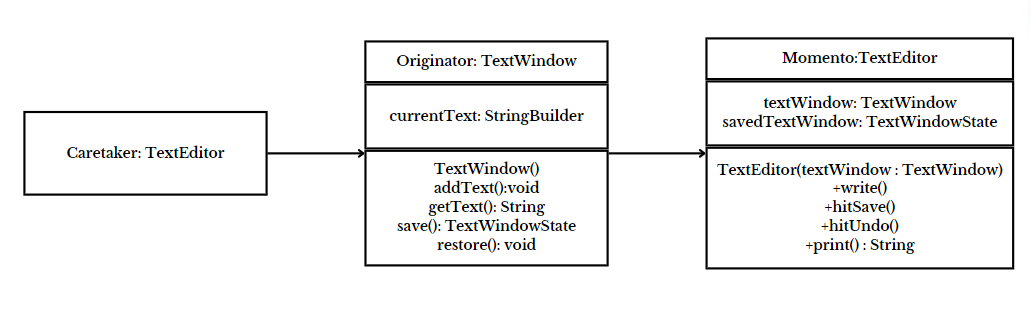
Here's a brief overview of how the Memento pattern typically works:

1. Originator: This is the object whose state needs to be saved. It creates a memento object containing a snapshot of its current state.
2. Memento: This is an immutable object that stores the state of the originator. It provides methods for getting the state and possibly applying it back to the originator.

3. Caretaker: This is responsible for keeping track of mementos. It can store multiple mementos, usually in a stack or some other data structure. It doesn't modify or examine the contents of the memento.

By using this pattern, you can ensure that the encapsulation of the originator is not compromised while still providing the ability to save and restore its state. It promotes separation of concerns by keeping the responsibility of managing state separate from the originator itself.

* **Program :** Implement memento design pattern for word example.
* **UML Diagram :**



* **Code :**

class TextWindow

{

private StringBuilder currentText;

public TextWindow()

{

this.currentText = new StringBuilder();

}

public void addText(String text)

{

currentText.append(text);

}

public String getText()

{

return currentText.toString();

}

public TextWindowState save()

{

return new TextWindowState(currentText.toString());

}

public void restore(TextWindowState save)

{

currentText = new StringBuilder(save.getText());

}

}

class TextWindowState

{

private String text;

public TextWindowState(String text)

{

this.text = text;

}

public String getText()

{

return text;

}

}

class TextEditor

{

private TextWindow textWindow;

private TextWindowState savedTextWindow;

public TextEditor(TextWindow textWindow)

{

this.textWindow = textWindow;

}

public void write(String text)

{

textWindow.addText(text);

}

public void hitSave()

{

savedTextWindow = textWindow.save();

}

public void hitUndo()

{

textWindow.restore(savedTextWindow);

}

public String print()

{

return textWindow.getText();

}

}

class Main

{

public static void main(String[] args)

{

TextEditor textEditor = new TextEditor(new TextWindow());

textEditor.write("The Memento Design Pattern \n");

textEditor.write("Initial State: Welcome \n");

textEditor.hitSave();

textEditor.write("Add some more text to the document\n");

System.out.println(textEditor.print());

textEditor.hitUndo();

System.out.println(textEditor.print());

}

}

* **Output :**

