Memento pattern

Memento pattern is used to restore state of an object to a previous state. Memento pattern falls under behavioral pattern category.

Design participants

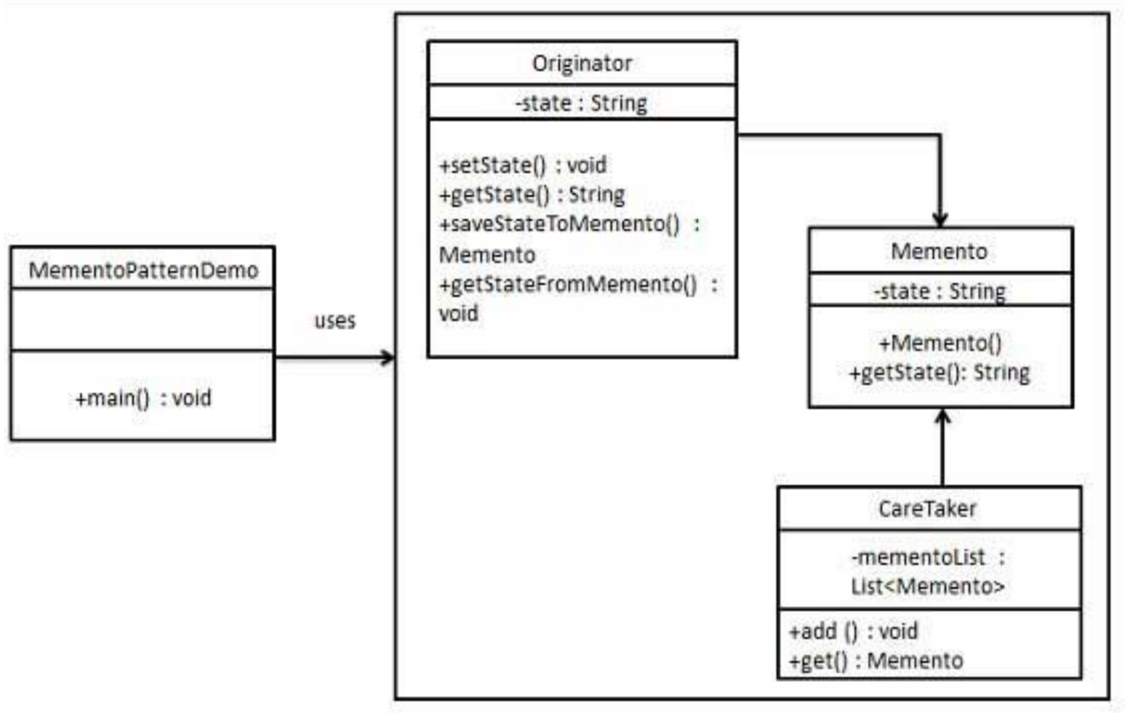
The memento pattern has three participants.

1. **Originator** – is the object that knows how to create and save it's state for future. It provides methods `createMemento()` and `restore(memento)`.
2. **Caretaker** – performs an operation on the Originator while having the possibility to rollback. It keeps track of multiple mementos. Caretaker class refers to the Originator class for saving (`createMemento()`) and restoring (`restore(memento)`) originator's internal state.
3. **Memento** – the lock box that is written and read by the Originator, and shepherded by the Caretaker. In principle, a memento must be in immutable object so that no one can change it's state once created.

Implementation

Memento pattern uses three actor classes. Memento contains state of an object to be restored. Originator creates and stores states in Memento objects and Caretaker object is responsible to restore object state from Memento. We have created classes *Memento*, *Originator* and *CareTaker*.

MementoPatternDemo, our demo class, will use *CareTaker* and *Originator*objects to show restoration of object states.



Step 1

Create Memento class.

Memento.java

```
public class Memento {
   private String state;

   public Memento(String state){
      this.state = state;
   }

   public String getState(){
      return state;
   }
}
```

Step 2

Create Originator class

Originator.java

```
public class Originator {
   private String state;

   public void setState(String state){
      this.state = state;
   }

   public String getState(){
      return state;
   }

   public Memento saveStateToMemento(){
      return new Memento(state);
   }

 public void getStateFromMemento(Memento memento){
      state = memento.getState();
   }
}
```

Step 3

Create CareTaker class

CareTaker.java

```
import java.util.ArrayList;
import java.util.List;

public class CareTaker {
   private List<Memento> mementoList = new ArrayList<Memento>();

   public void add(Memento state){
      mementoList.add(state);
   }

   public Memento get(int index){
      return mementoList.get(index);
   }
}
```
Step 4

Use *CareTaker* and *Originator* objects.

MementoPatternDemo.java

```
public class MementoPatternDemo {
   public static void main(String[] args) {

      Originator originator = new Originator();
      CareTaker careTaker = new CareTaker();

      originator.setState("State #1");
      originator.setState("State #2");
      careTaker.add(originator.saveStateToMemento());

      originator.setState("State #3");
      careTaker.add(originator.saveStateToMemento());

      originator.setState("State #4");
      System.out.println("Current State: " + originator.getState());


      originator.getStateFromMemento(careTaker.get(0));
      System.out.println("First saved State: " + originator.getState());
      originator.getStateFromMemento(careTaker.get(1));
      System.out.println("Second saved State: " + originator.getState());
   }
}
```

Verify the output.

```
Current State: State #4
First saved State: State #2
Second saved State: State #3
```

In this example, we are creating memento for an *Article* object which has three basic attributes – id, title and content. *ArticleMemento* class is used as memento for *Article* objects.

Article.java

```
public class Article //Originator
{
    private long id;
    private String title;
    private String content;

    public Article(long id, String title) {
        super();
        this.id = id;
        this.title = title;
    }

    //Setters and getters

    public ArticleMemento createMemento()
    {
        ArticleMemento m = new ArticleMemento(id, title, content);
        return m;
    }

    public void restore(ArticleMemento m) {
        this.id = m.getId();
        this.title = m.getTitle();
        this.content = m.getContent();
    }

    @Override
    public String toString() {
        return "Article [id=" + id + ", title=" + title + ", content=" + content + "]";
    }
}
```

ArticleMemento.java

```
public final class ArticleMemento
{
    private final long id;
    private final String title;
    private final String content;

    public ArticleMemento(long id, String title, String content) {
        this.id = id;
        this.title = title;
        this.content = content;
    }

    public long getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }

    public String getContent() {
        return content;
    }
}
```

The Main class is acting as Caretaker which creates and restores the memento objects.

Main.java

```
public class Main
{
    public static void main(String[] args)
    {
        Article article = new Article(1, "My Article");
        article.setContent("ABC");      //original content
        System.out.println(article);
        ArticleMemento memento = article.createMemento();   //created immutable
        //memento
        article.setContent("123");      //changed content
        System.out.println(article);

        article.restore(memento);       //UNDO change
        System.out.println(article);    //original content
    }
}
```

Program Output.

Console

```
Article [id=1, title=My Article, content=ABC]
Article [id=1, title=My Article, content=123]
Article [id=1, title=My Article, content=ABC]
```