



MEMENTO

Design Pattern





INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

Report

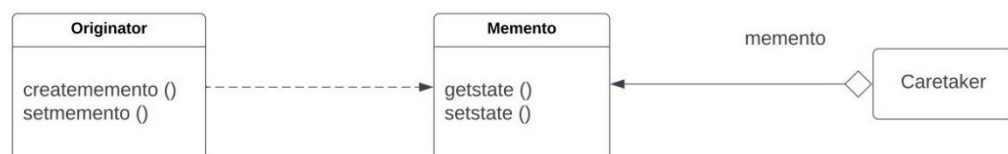
Name: Tamoor Ahmad
Registration No: 4125-FBAS/BSSE/F20
Course Code: SE322
Course Title: Software Design and Architecture

Submitted to: Dr. Muhammad Nasir

Database transactions are operations on the database that occur in an atomic, consistent, durable, and isolated fashion. A transaction can contain multiple operations on the database; each operation can succeed or fail, however a transaction guarantees that if all operations succeed, the transaction will commit and would be final. And if any operation fails, then the transaction would fail, and all operations would roll-back and leave the database as if nothing has happened.

This mechanism of rolling back uses the memento design pattern. Consider an object representing a database table, a transaction manager object which is responsible of performing transactions must perform operations on the table object while having the ability to undo the operation if it fails, or if any operation on any other table object fails. To be able to rollback, the transaction manager object would ask the table object for a memento before performing an operation and thus in case of failure, the memento object would be used to restore the table to its previous state.

Class Diagram:



Source Code:

```
using System;
namespace Memento.Structural
{
    public class Program
    {
        public static void Main(string[] args)
        {
        }
```

```

        Originator obj = new Originator();
        obj.State = "On";

        Caretaker c = new Caretaker();
        c.Memento = obj.CreateMemento();

        obj.State = "Off";

        obj.SetMemento(c.Memento);

        Console.ReadKey();
    }
}

public class Originator
{
    string state;
    public string State
    {
        get { return state; }
        set
        {
            state = value;

            Console.WriteLine("State = " + state);
        }
    }

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

    public void SetMemento(Memento memento)
    {
        Console.WriteLine("Rolling to back ");
        State = memento.State;
    }
}

public class Memento
{
    string state;

    public Memento(string state)
    {
        this.state = state;
    }
    public string State
    {
        get { return state; }
    }
}

public class Caretaker
{
    Memento memento;
    public Memento Memento
    {

```

```
        set { memento = value; }  
        get { return memento; }  
    }  
}
```

Output:

Select C:\Users\ITS\source\repos\Memento\Memento\bin\Debug\Memento.exe

```
State = On  
State = Off  
Rolling to back  
State = On
```

