MOHAMMAD ARIZ HAIDER

MEMENTO PATTERN

UML DIAGRAM: MEMENTO

Originator
+ DemoList:type = CList
+ setState(DemoList CList) +saveStateToMemento() +getStateFromMemento()

Memento
State: CList
setList(state: CList) getState(): CList

Caret
//keeps track of t
mem
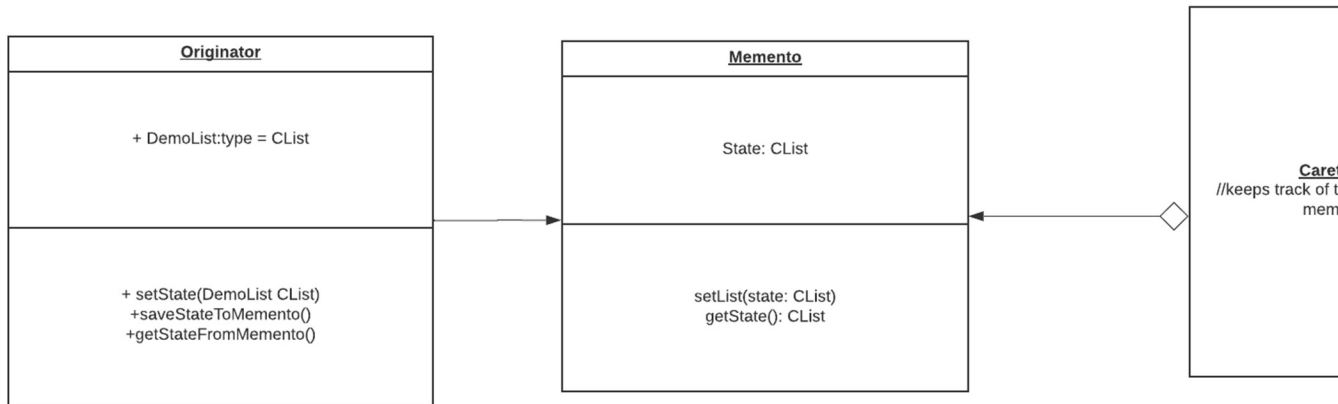
REVELANT CODE TO MEMENTO PATTERN

Originator

```java
package Memento;
public class Originator //Creates the save time
{
        DemoList CList; //CList = Car List

        public DemoList getState()
        {
                return CList;
        }

        public void setState(DemoList  CList) // saves a sate of the current list
        {
                this.CList = CList;
        }

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

        public void getStateFromMemento(Memento memento)// loads  or revert back saved
state
        {
                CList = memento.getState();
        }

}
```

CareTaker

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

public class CareTaker //adds more save states as list is changed
{
        private     <Memento> CarList = new ArrayList<Memento>();
        private     <Memento> CopyCarLList = (     )(((ArrayList)CarList).clone());

        public void add(Memento mem)
        {
            CarList.add(mem); //
            System.out.println("List of Cars:" + CarList + "\n");
        }

        public Memento get(int index)
        {
            return CarList.get(index);
        }
}
```

Memento

```java
package Memento;
public class Memento //save state
{
        private DemoList CList;

        public Memento (DemoList CList)
        {
                super();
                this.CList = CList;
        }
        public DemoList getState()
        {
                return CList;
        }
        public void setList(Memento memento)
        {
                CList = memento.getState();
        }

        public String toString()
        {
                return "Add the Cars to the List: " + CList;
        }
}
```

DemoList

```java
package Memento;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class DemoList //List to be added from somewhere
{
    private      <Car> Cars = new ArrayList<Car>();

    private String[] RNGMakeList = {"Alpha", "Beta", "Gamma", "Delta", "Epsilon",
"Sigma"};
    private String[] RNGModelList = {"SEDAN", "COUPE", "SPORTS CAR", "STATION
WAGON", "HATCHBACK", "CONVERTIBLE", "SPORT-UTILITY VEHICLE", "PICKUP TRUCK"};
    private short[] RNGYearList = {1989, 1994, 1999, 2005, 2006, 2009, 2011, 2015,
2020, 2022};
    private double[] RNGPriceList = {12300.99, 56300.99, 32500.00, 43600.50,
78900.00, 32200.00};

    public DemoList()
    {
        RNGList();
    }

    public      <Car> getCars()
    {
        return Cars;
    }

    public void setCars(     <Car> cars)
    {
        Cars = cars;
    }

    public DemoCar RNGCar()
    {

        String RNGMake = RNGMakeList[new Random().nextInt(RNGMakeList.length)];
        String RNGModel = RNGModelList[new
Random().nextInt(RNGModelList.length)];
        short RNGYear = RNGYearList[new Random().nextInt(RNGYearList.length)];
        double RNGPrice = RNGPriceList[new
Random().nextInt(RNGPriceList.length)];
        return new DemoCar(RNGMake, RNGModel, RNGYear, RNGPrice);
    }

    private void RNGList()
    {
        int index = 25;

        for(int i = 0; i<index; i++)
        {
            Cars.add(RNGCar());
        }
    }
```

UNIT TESTS

```java
1  package Memento;
2
3⊕ import static org.junit.Assert.*;
13
14 public class MementoTest
15 {
16     DemoList CList;
17     Originator Org = new Originator ();
18     CareTaker CareT = new CareTaker();
19
20⊖    @BeforeClass // runs at very starting of iteration
21     public static void setUpBeforeClass() throws Exception
22     {
23         System.out.println("check if a state can be saved and loaded");
24     }
25
26
27
28⊖    @Test
29     public void saveSate() //FIrst Test : saves a state
30     {
31         Org.setState(CList); //adds a default state
32     }
33
34⊖    @Test
35     public void test2() //Checks if it can go back to last saved state
36     {
37         CareT.add(Org.saveStateToMemento());
38     }
39
40
41 }
42
```

COMPONENT TEST

A list of cars exist for the user to buy (gets
removed from the list) and sell, (gets added to the

car list). The List is first a default state (untouched) which exists in the Originator, it then is saved as a Memento (Save File), and the CareTaker keeps track of how many different save files are added or removed (keeps track of states). If a car is bought by accident and the list must be return to previous state, the CareTaker takes the saved state in the Memento which has the index no. of state before purchase and restores the list to that state.