|  |
| --- |
| **MementoSerializable** |
| **Line** | **Code** |
| **1****2****3****4****5****6****7****8****9****10****11****12****13****14****15****16****17****18****19****20****21****22****23****24****25****26****27****28****29****30****31****32****33****34****35****36****37****38****39****40****41****42****43****44****45****46****47****48****49****50****51****52****53****54****55****56****57****58****59****60****61****62****63****64****65****66****67****68****69****70****71****72****73****74****75****76****77****78****79****80****81****82****83****84****85****86****87****88** | **using** System;**using** System.IO;**using** System.Runtime.Serialization.Formatters.Binary;**using** System.Collections.Generic;**using** System.Collections;**namespace** MementoSerializable { **class** Program { //Client **static void** Main(**string**[] args) { //References to the mementos Caretaker[] c = **new** Caretaker[10]; Originator originator = **new** Originator(); **int** move = 0; //Iterator for the moves Simulator simulator = **new** Simulator(); **foreach** (**string** command **in** simulator) { //Check for undo **if** ((command[0] == '\*') && (move > 0)) originator.GetMemento(c[move - 1].Memento); **else** originator.Operation(command); move++; c[move] = **new** Caretaker(); c[move].Memento = originator.SetMemento(); } Console.ReadKey(); } [Serializable()] **class** Originator { List<**string**> state = **new** List<**string**>(); **public void** Operation(**string** s) { state.Add(s); **foreach** (**string** line **in** state) Console.WriteLine(line); Console.WriteLine("==========================================="); } //The reference to the memento is passed back to the client **public** Memento SetMemento() { Memento memento = **new** Memento(); **return** memento.Save(state); } **public void** GetMemento(Memento memento) { state = (List<**string**>)memento.Restore(); } } [Serializable()] //Serializes by deep copy to memory and back **class** Memento { MemoryStream stream = **new** MemoryStream(); BinaryFormatter formatter = **new** BinaryFormatter(); **public** Memento Save(**object** o) { formatter.Serialize(stream, o); **return this**; } **public object** Restore() { stream.Seek(0, SeekOrigin.Begin); **object** o = formatter.Deserialize(stream); stream.Close(); **return** o; } } **class** Caretaker { **public** Memento Memento { get; set; } } **class** Simulator : IEnumerable { **string**[] lines ={ "The curfew tolls the knell of parting day", "The lowing herd winds slowly o'er the lea", "Uh hum uh hum", "\*UNDO", "The plowman homeward plods his weary way", "And leaves the world to darkness and to me."}; **public** IEnumerator GetEnumerator() { **foreach** (**string** element **in** lines) **yield return** element; } } }} |