March 1, 2018

Jacopo Carravieri

corso di tecniche speciali di programmazione a.a. 2017-2018

UNIVERSITÀ DEGLI STUDI DI MILANO

Memento pattern

implementazione in aspectj

Indice

[Il pattern ‘Memento’ 2](#_Toc507859972)

[Implementazione 2](#_Toc507859973)

[Struttura di riferimento 2](#_Toc507859974)

[Classi e interfacce 2](#_Toc507859975)

[Aspetti 2](#_Toc507859976)

[Class diagram 2](#_Toc507859977)

[Sequence diagrams 2](#_Toc507859978)

[Utilizzo 2](#_Toc507859979)

[Esempio 1: Editor di testi con supporto per le tab 3](#_Toc507859980)

[Il programma 3](#_Toc507859981)

[Class diagram 3](#_Toc507859982)

[Sequence diagrams 3](#_Toc507859983)

[Esempio 2: Disegno di diagrammi (dal libro) 4](#_Toc507859984)

[Il programma 4](#_Toc507859985)

[Class diagram 4](#_Toc507859986)

[Sequence diagrams 4](#_Toc507859987)

[Esempio 3: History ad albero 5](#_Toc507859988)

[Estensione delle funzionalità 6](#_Toc507859989)

[Supporto alle chiamate ricorsive 6](#_Toc507859990)

# Introduzione

Nel contesto del corso di Tecniche Speciali di Programmazione, tenuto dal prof. Walter Cazzola presso l’Università degli Studi di Milano, è stato richiesto di produrre un’implementazione il più possibile generica del pattern **Memento**. Tale progetto era da svolgere per mezzo del framework per l’AOP **AspectJ**, facendo diventare in questo modo il pattern un concetto trasversale.

La presente documentazione fornisce una panoramica sulle scelte implementative, le difficoltà riscontrate e le relative soluzioni; ed illustra il senso ed il funzionamento dei tre esempi forniti al fine di provare il reale funzionamento del progetto.

## Implementazione

Al fine di decidere concretamente come implementare il pattern si è inizialmente esaminata la struttura di riferimento del pattern stesso:

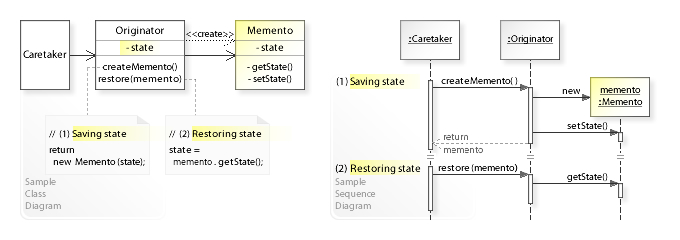


Figura 1 Diagramma delle classi e Diagramma di sequenza del pattern Memento  
https://en.wikipedia.org/wiki/Memento\_pattern

Nella sua forma di base il pattern è organizzato nei seguenti elementi:

* **Originator**: si tratta dell’oggetto dotato di uno stato interno che deve essere salvato ed eventualmente ripristinato per mezzo degli oggetti Memento. Si può notare dal diagramma di sequenza come si occupi egli stesso, dopo l’invocazione di *createMemento*, di esternalizzare il suo stato. In questo modo il suo funzionamento interno non deve essere conosciuto dalle altre classi; in altre parole lo stato è **opaco** e la sua implementazione può variare nel tempo.
* **Memento**: è la classe le cui istanze rappresentano lo stato dell’originator in un determinato istante di tempo. È conservata ed eventualmente utilizzata in forma pubblica dal Caretaker, ma come accennato nel punto precedente i suoi membri privati sono accessibili solamente all’originator.
* **Caretaker**:

### 

### Struttura di riferimento

### Classi e interfacce

### Aspetti

### Class diagram

### Sequence diagrams

## Utilizzo

# Esempio 1: Editor di testi con supporto per le tab

## Il programma

## Class diagram

## Sequence diagrams

# Esempio 2: Disegno di diagrammi (dal libro)

## Il programma

## Class diagram

## Sequence diagrams

# Esempio 3: History ad albero

# Estensione delle funzionalità

## Supporto alle chiamate ricorsive