

Design pattern 3

Introduzione

Obiettivi:

- Applicare propriamente i pattern del modello GoF

Questa sezione rappresenta il naturale conseguimento del documento *Design Pattern 2*, in cui sono elencati ulteriori pattern *comportamentali*, *creazionali* e *strutturali* del catalogo *GoF*.

Singleton pattern

Problema

Come garantire che sia creata una sola istanza di una classe e fare in modo che sia condivisibile dagli elementi del modello?

Singleton è un pattern creazionale adoperato per istanziare oggetti, in maniera tale che siano rispettati i principi di qualità del software e cercando di manipolare al meglio le dipendenze tra le classi. La volontà di implementare un meccanismo simile, avviene qualora differenti elementi del dominio richiedano lo stesso processo esecutivo, da cui saranno restituiti certi *behaviors* oppure dati in *output*.

Per cui l'intento promuove la creazione di una singola istanza della classe software analizzata, affinché essa possa essere condivisa tra le entità del sistema elaborato. Tendenzialmente è dichiarata come una variabile *privata*, in quanto lo stesso *costruttore* è *protetto* oppure *privato* ai fini del pattern presentato, a cui si associa una funzione *pubblica* che incapsula al suo interno l'intero codice di inizializzazione e provvede all'accesso all'istanza in questione.

Soluzione

Creare una *singola* istanza della classe software analizzata e provvedere ad un *unico punto di accesso*.

Generalmente l'utilizzo di *Singleton* avviene solo se soddisfatti tre criteri comportamentali, suddivisi in:

- Sviluppo progettuale secondo il principio *lazy initialization*, ossia posticipare la creazione di un oggetto fino al momento in cui non sia realmente necessario
- Non sia possibile attribuire ad alcun elemento del modello l'*appartenenza* dell'istanza creata
- Nonostante sia sviluppato un metodo pubblico per rendere accessibile l'oggetto della classe software, non deve essere garantita la possibilità che la funzione sia richiamabile a livello globale