

Design Patterns

Concetti Introduttivi

a cura di **Angelo Furfaro**
da “Design Patterns”, Gamma et al.
“Patterns in Java”, Grand

Dipartimento di
Ingegneria Informatica, Elettronica, Modellistica e Sistemistica
Università della Calabria, 87036 Rende(CS) - Italy
Email: a.furfaro@unical.it
Web: <http://angelo.furfaro.dimes.unical.it>

Che cosa è un Design Pattern

- Un pattern di progetto costituisce una “soluzione” pronta all’uso o preliminarmente adattabile, per specifici (sotto) problemi applicativi
- Indirizzano questioni rilevanti di riusabilità, modificabilità, estendibilità, disaccoppiamento di moduli software
- La conoscenza dei pattern offre un linguaggio (sintassi e semantica) di comunicazione all’interno di team di progetto
- I pattern si possono classificare come: **creazionali**, **strutturali** o **comportamentali** a seconda della loro sfera di azione. Ad es. un pattern State è di tipo comportamentale.
- Non esiste comunque consenso generalizzato sulla classificazione
- Possono essere espressi con diagrammi UML che coinvolgono classi, interfacce e oggetti, e loro relazioni
- 23 pattern di riferimento sono descritti nel libro “Design Patterns” (Gang-of-Four, E. Gamma, R. Helm, R. Johnson, J. Vlissides, 1995). Tuttavia, altri pattern esistono e si scoprono tuttora. Di ogni pattern si indicherà la provenienza

Benefici derivanti dall'uso dei Design Pattern

- Facilitano il riuso efficace di progetti ed architetture, rendendo accessibili agli sviluppatori tecniche consolidate.
- Aiutano a scegliere tra le alternative progettuali che rendono il sistema riusabile ed a scartare quelle che comprometterebbero la riusabilità.
- Rispondono ad esigenze quali:
 - isolare e incapsulare gli “aspetti che cambiano” in un progetto
 - concretizzare il principio **open/closed** nello sviluppo di componenti software: apertura ai cambiamenti (*openness*) da ottenere con l'aggiunta di nuovi moduli ma lasciando il più possibile invariati gli altri moduli. (*closeness*)
- Possono aiutare a migliorare la documentazione e la manutenzione dei sistemi esistenti.

Struttura di un Pattern

Ogni pattern è caratterizzato da (almeno) quattro elementi essenziali

- **Nome**

È un nome simbolico che descrive sinteticamente un problema di progettazione, le sue soluzioni, e le conseguenze della soluzione scelta.

- **Problema (o intento)**

Descrive quando è utile applicare il pattern. Spiega il problema che risolve ed il suo contesto.

- **Soluzione (concettuale)**

Illustra gli elementi del progetto, le loro relazioni e collaborazioni. Non fornisce un'implementazione ma una descrizione astratta del modo in cui una configurazione di elementi (classi e oggetti) può risolvere il problema di progettazione.

- **Conseguenze**

Sono i risultati ed i vincoli che derivano dall'applicazione del pattern. Le conseguenze sono importanti al fine di poter valutare soluzioni alternative e per comprendere/stimare i costi e i benefici dell'applicazione del pattern.

- Il concetto di pattern (schema progettuale) è stato introdotto originariamente da Christopher Alexander che nel suo libro: C. Alexander ed altri, "A pattern language: towns, buildings, constructions", Oxford University Press, 1977
- Delineò qualcosa come 250 pattern per la progettazione architettonica di case e città Alexander suggerì, per ogni pattern, un breve nome, un contesto d'uso, il problema che si intende risolvere, l'indicazione della soluzione
- Un esempio di pattern di Alexander è il Corridoio corto Il pattern suggerisce che nelle abitazioni civili si devono evitare corridoi più lunghi di 16/17 metri, in quanto essi tendono ad essere piuttosto bui e desolati così da indurre disagio nelle persone.
- La soluzione consiste nel progettare corridoi corti, assistiti da ampie vetrate, con rientranze tipo stanze con mobili, tappeti etc

Classificazione dei Pattern

I Design Pattern sono classificati in base a due criteri:

- **Scopo** (indica ciò che fa il pattern)
 - *Strutturale*
 - *Creazionale*
 - *Comportamentale*
- **Raggio d'azione**
 - *Classi* (Riguardano le relazioni statiche tra classi)
 - *Oggetti* (Riguardano le relazioni tra oggetti le quali possono essere dinamiche)
- La maggior parte dei pattern ha gli oggetti come raggio d'azione

Pattern Creazionali

- I class-pattern di questa categoria delegano parte del processo di creazione di un oggetto alle sottoclassi
- Gli object-pattern lo delegano ad altri oggetti

Classificazione dei Pattern

Pattern Strutturali

- I class-pattern di questa categoria ricorrono all'ereditarietà per comporre le classi
- Gli object-pattern descrivono modi per raggruppare oggetti

Pattern Comportamentali

- I class-pattern di questa categoria usano l'ereditarietà per descrivere algoritmi e flussi di controllo
- Gli object-pattern descrivono il modo in cui gruppi di oggetti cooperano per svolgere un'attività che un singolo oggetto non potrebbe portare a termine da solo