

Programmazione 3 e Laboratorio di Programmazione 3

Principi di Design Pattern

Angelo Ciaramella

Introduzione

- OO applicato ai processi di sviluppo software complessi
 - soluzione non semplice
 - passaggio critico tra analisi e progettazione
- Design Pattern
 - scelta di design particolarmente delicate



Introduzione

■ Prima definizione di pattern

■ Christopher Alexander

- Architetto austriaco (Università di Berkeley) iniziò a formalizzare tale concetto sin dagli anni '60
 - *Pattern Language: Towns, Buildings, Construction* (Oxford University Press, 1977)
- pattern - una soluzione architeturale che può risolvere problemi in contesti eterogenei

■ Formalizzazione del concetto

■ Gang of Four (GoF)

- Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides
- 23 pattern
- libro - *Design Patterns: elements of reusable object oriented software* (Addison-Wesley)



Design Pattern

■ Definizione

I Design Pattern rappresentano soluzioni di progettazione generiche applicabili a problemi ricorrenti all'interno di contesti eterogenei

- la **progettazione** è direttamente proporzionale all'**esperienza del progettista**
- un **progettista** esperto risolve i problemi che si presentano utilizzando **soluzioni** che in passato hanno **già dato buoni risultati**



■ Presentazione del pattern

- nome

- il **problema** a cui può essere applicato

- la **soluzione** (non in un caso particolare)

- le **conseguenze**

■ Descrizione del pattern

- nome e classificazione

 - importante per il vocabolario utilizzato nel progetto

- **scopo**

 - breve descrizione di cosa fa il pattern e suo fondamento logico



- **Descrizione del pattern**
 - **Nomi alternativi** (se ve ne sono)
 - molti pattern sono conosciuti con più nomi perché “scoperti” da persone diverse che avevano dato loro nomi diversi
 - **Motivazione**
 - descrizione di uno scenario che illustra un problema di progettazione e la soluzione offerta
 - **Applicabilità**
 - quando può essere applicato il pattern
 - **Struttura** (o modello)
 - rappresentazione grafica delle classi del pattern mediante un linguaggio di notazione (libro dei GOF, OMT di Rumbaugh e i diagrammi di interazione di Booch)



- **Descrizione del pattern**
 - **Partecipanti**
 - le classi/oggetti con le proprie responsabilità
 - **Collaborazioni**
 - come collaborano i partecipanti per poter assumersi le responsabilità
 - **Conseguenze**
 - pro e contro dell'applicazione del pattern
 - **Implementazione**
 - come si può implementare il pattern
 - **Codice d'esempio**
 - frammenti di codice che aiutano nella comprensione del pattern (GoF, C++ e SmallTalk)
 - **Pattern correlati**
 - relazioni con altri pattern
 - **Utilizzi noti**
 - esempi di utilizzo reale del pattern in sistemi esistenti



Categorie GoF

- **Categorie (scopo del pattern)**
 - **Pattern Creazionali**
 - propongono soluzioni per **creare oggetti**
 - **Pattern Strutturali**
 - propongono soluzioni per la **composizione strutturale di classi e oggetti**
 - **Pattern Comportamentali**
 - propongono soluzioni per gestire il modo in cui vengono **suddivise le responsabilità delle classi e degli oggetti**



Categorie GoF

Scopo		Creazionale	Strutturale	Comportamentale
Raggio di azione	Class	Factory Method	Adapter (class)	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter (object) Bridge Composite Decorator Facade Flyweight Proxy	Chain of responsibility Command Iterator Mediator Memento Observer State Strategy

Categorie dei pattern GoF



