DESIGN PATTERNS

Corso ENG: DP

§ Design patterns GoF

§ Cenni ai design principles

Presentazione: Oggi

Breve illustrazione della struttura del corso

- § Cenni ai design principles
- § Illustrazione di **un** dp esemplificativo (chain of responsibility)

Presentazione: Next step

TBD per incontro/i successivo/i

- Lab interattivo su altri DP da scegliere in base a feedback e interesse
- Varie ed eventuali a richiesta

Design principles

§ DRY (Don’t Repeat Yourself)

§ KISS (Keep it short and simple)

§ YAGNI (You ain’t gonna need it)

§ S.O.L.I.D. (acronimo per 5 principles)

Design principles - SOLID

§ Single Responsibility

§ Open Closed Principle

§ Liskov

§ Interface Segregation

§ Dependency Inversion

Design pattern - Introduzione / 1

§ Soluzioni a problemi comuni e ricorrenti

§ Frutto di esperienza e processo induttivo

§ Standardizzazione

§ Generalità / astrazione

§ Primo anno citato di storia: 1977 (-> induzione + esperienza)

Design pattern - Introduzione / 2

Un pattern descrive il nucleo di una soluzione relativa,
un problema che compare frequentemente in un dato contesto

Christopher Alexander (architetto)

Design pattern - Introduzione / 3

Il modello della soluzione deve essere strutturato in modo tale che "si possa usare tale soluzione un milione di volte, senza mai farlo allo stesso modo"

Christopher Alexander (architetto)

Macro categorie di pattern

Le principali categorie di pattern sw sono:

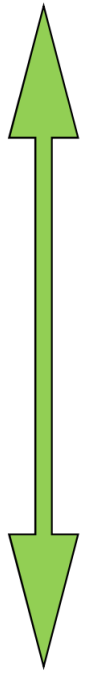
- Architetturali (es. client server, MVC)
- Progettuali ("via di mezzo", i "nostri")
- Idiomi (basso livello, legati al linguaggio, es. il Singleton)

Classificazione dei pattern

Creazionali	Strutturali	Comportamentali
(costruttore)	*(attributi)*	*(metodi)*
Factory method	Adapter	Interpreter
Abstract Factory	Bridge	Template Method
Builder	Composite	Chain of Responsability
Prototype	Decorator	Command
Singleton	Facade	Iterator
	Flyweight	Mediator
	Proxy	Memento
		Observer
		State
		Strategy
		Visitor

Frequenza d'uso dei pattern

bassa



alta

3.Interpreter	1
6.Memento	1
4.Prototype	2
6.Flyweight	2
5.Mediator	2
11.Visitor	2
3.Builder	3
2.Bridge	3
1.Chain of Responsibility	3
1.Factory Method	4
2.Abstract Factory	4
1.Adapter	4
4.Decorator	4
2.Command	4
8.State	4
9.Strategy	4
5.Singleton	5
3.Composite	5
5.Façade	5
7.Proxy	5
4.Iterator	5
7.Observer	5
10.Template Method	5

Leitmotiv

Nei pattern (creazionali in primis, ma tutti) si notano due fondamentali linee guida:

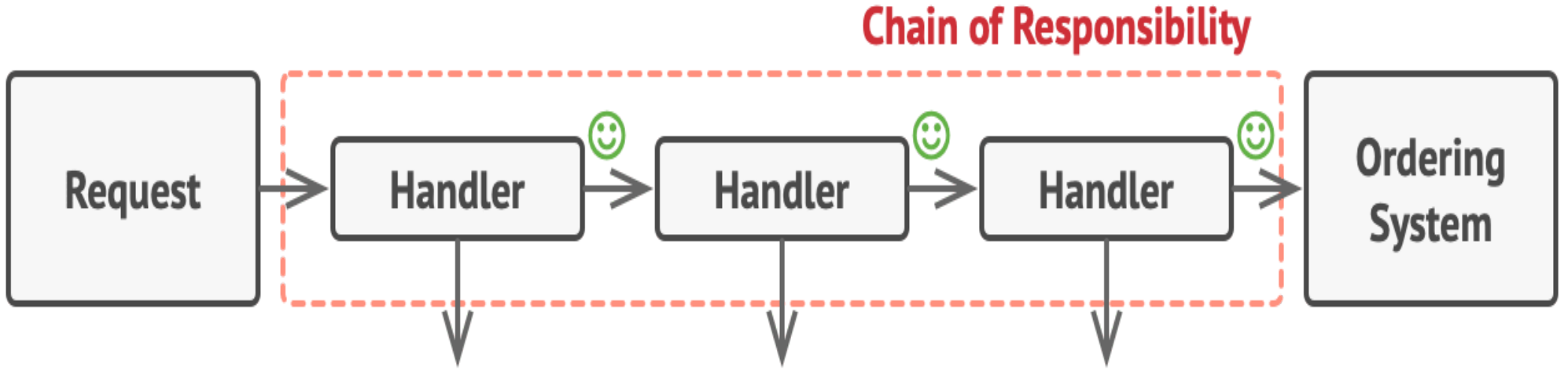
- Chiamate polimorfiche di metodi
- Utilizzo degli oggetti attraverso le loro interfacce

Pattern scelto

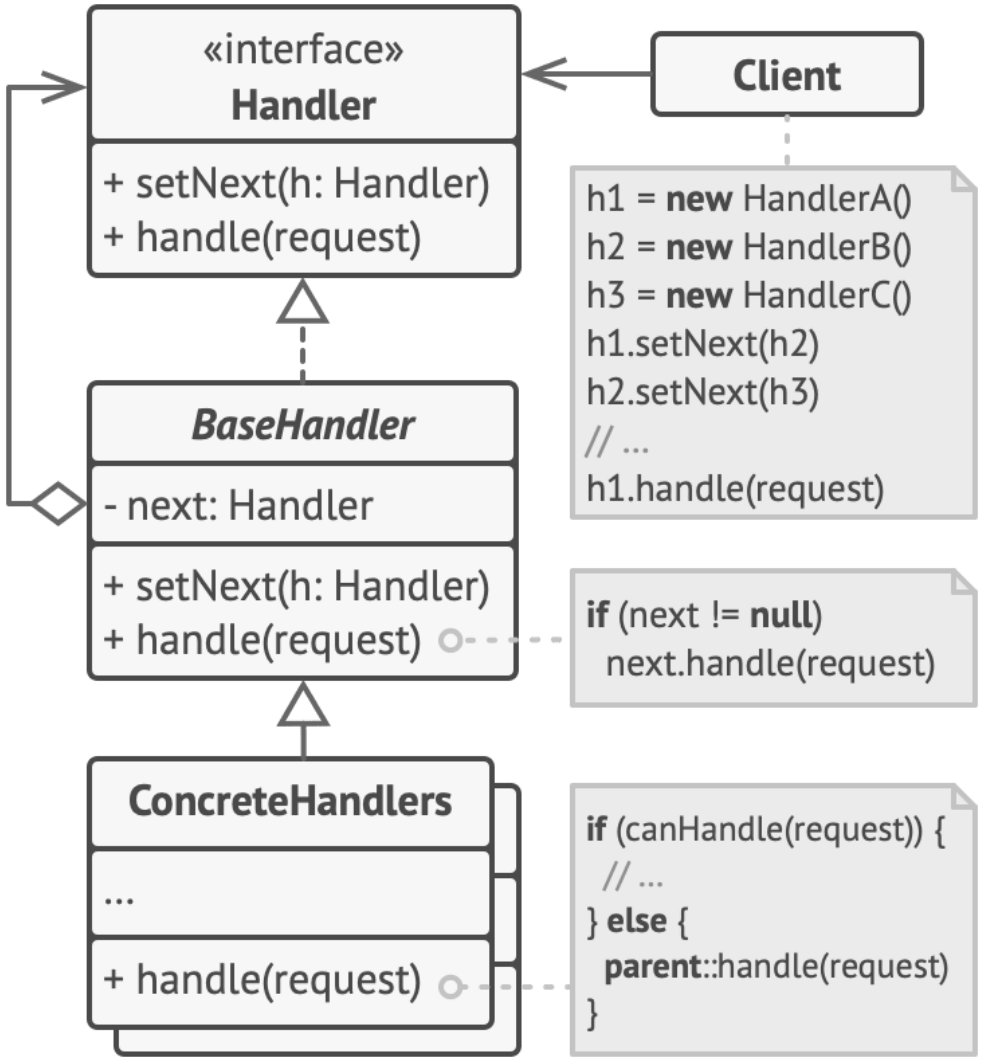
Chain of responsibility

E' un pattern comportamentale (*"behavioural"*)

Lo scopo del Chain of Responiibility è quello di fornire un serie di handler di una stessa richiesta in sequenza,non sapendo in anticipo quale evaderà la richiesta.



Chain Of responsibility / Struttura



Chain Of responsibility / Esempio

Elevamento a potenza

- Per piccoli numeri e esponente intero, si procede per moltiplicazioni ripetute
- Per numeri intermedi si usa usa math.pow
- Per grandi numeri si effettua un arrotondamento intero con l'operatore `**`

Chain Of responsibility / Filosofia

L'utilizzo (la scelta) del chain of responsibility fa sì che la scelta di gestire la richiesta oppure di rimandarla allo step succesivo sta nello step stesso (a differenza ad esempio dello strategy dove la selezione della strategie è "esterna" rispetto agli step stessi)