



Ente Ospedaliero Cantonale

# Design pattern ....

Non sempre le cose vecchie sono inutili ....

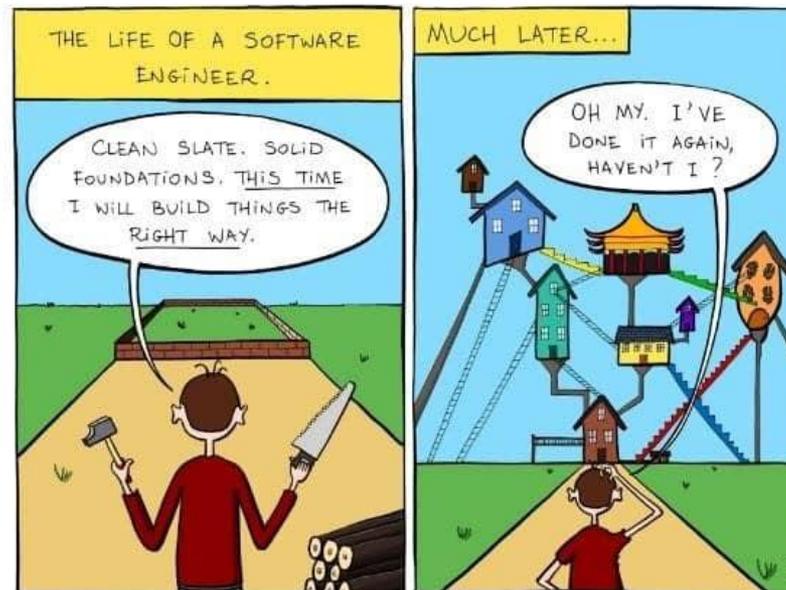
*F.Mora*

*24 marzo 2022*

eoc

*«Projects fails despite the latest technology for lack of ordinary solutions»*

Ralph Jonson



# Cosa sono i design patterns?

La parola pattern sta a significare un disegno, una trama che si ripete ancora ed ancora.

Nel caso di sviluppo software, un design pattern è una **soluzione** generale e riutilizzabile per un comune e **ricorrente problema**.

# Cenni storici

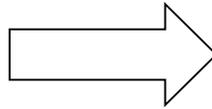
Trattati da Christopher Alexander nel 1990 ma non riconosciutogli all'unanimità.

Maggiore influenza l'ha avuta la pubblicazione del "gang of four book" del 1994.

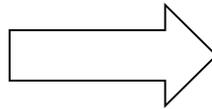


# Ma noi dovremmo usare principi vecchi come ....?

Anni '90



Oggi





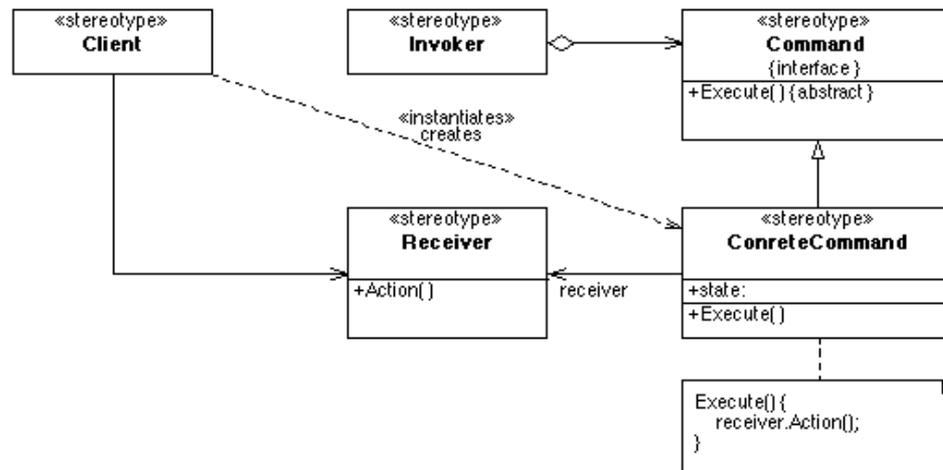
# Ma perché usarli?

- Possono aiutare ad accelerare il processo di sviluppo del software.
- L'utilizzo dei DP aiuta a prevenire problemi.
- Migliorano la leggibilità del codice.
- Facilitano la comunicazione tra software developers.

# Command pattern

(behavioral pattern)

Il Command pattern è uno dei design pattern che permette di isolare la porzione di codice che effettua un'azione (eventualmente molto complessa) dal codice che ne richiede l'esecuzione; l'azione è incapsulata nell'oggetto Command (wiki).

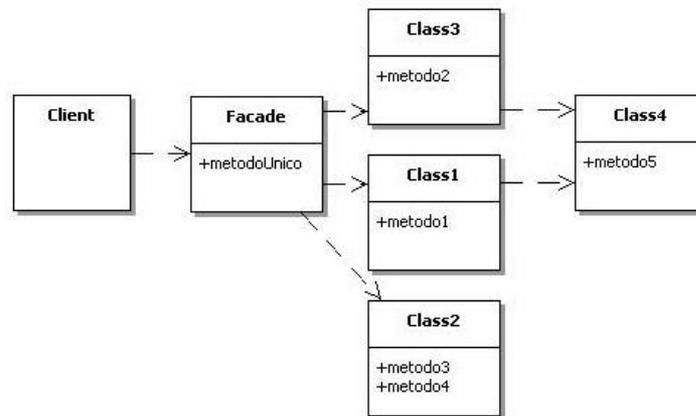


# Fachade pattern

(structural pattern)

Permette, attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce complesse e molto diverse tra loro, nonché a blocchi di codice complessi.

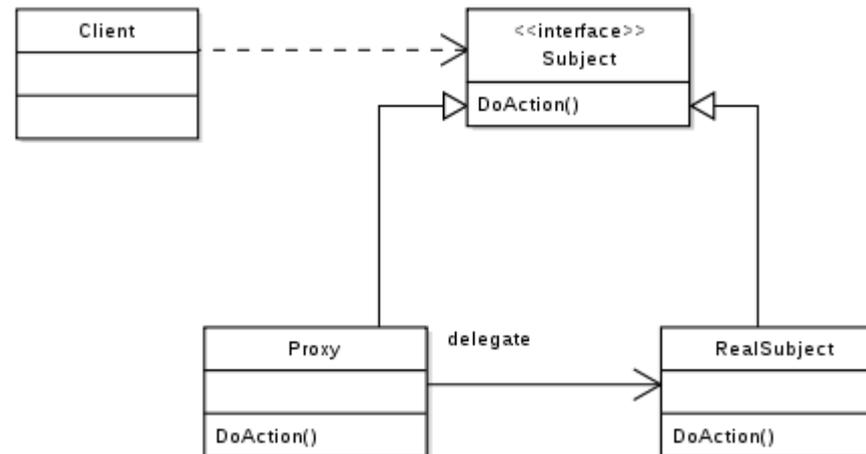
Lo scopo non è scrivere test ma programmare «correttamente».



# Proxy pattern

(structural pattern)

Nella sua forma più generale, un proxy è una classe che funziona come interfaccia per qualcos'altro



# Referenze

- Martin Fowler: Analysis patterns – reusable object modules (1997)
- Frank Buschmann, Regine Meunier, Hans Rhonet, Peter Sommerland, Michael Stall: System of patterns (1997)
- + web (google)



Fulgencio.mora@eoc.ch