DESIGN PATTERN

COSTRUISCI LA TUA CODEBASE DI SUCCESSO!

Bacaro Tech

CODE AND FUN

CIAO
CI PRESENTIAMO

Bacaro Tech
CODE AND FUN

Giorgio Basile
FE developer

Michele Scarpa
Full stack developer

VI PRESENTIAMO
BACAROTECH

Bacaro
Tech
CODE AND FUN

BacaroTech è un'inizitiva che ha il compito di ricreare quell’atmosfera gioiosa e di gruppo, tipica del bacaro veneziano, nel mondo dell’informatica, attraverso la divulgazione sui social, eventi e workshop

Portiamo avanti questa realtà dal 2023 e nel corso di questo tempo siamo riusciti a mettere a terra una community di appassionati che ammonta a quasi 3000 sviluppatori

NEL MENTRE
SIAMO ANCHE AUMENTATI

Bacaro Tech
CODE AND FUN

Danilo

Antonio

Vittorio

Moreno

Lorenzo

Davide

GRAZIE MILLE
PER IL SUPPORTO

Bacaro Tech

CODE AND FUN

BacaroTech

BacaroTech

Code and Fun

La tua **community di sviluppatori** dove si parla di programmazione a 360°: strutture dati, algoritmi, carriera tech e molto altro!

Canale Whatsapp

GitHub

Instagram

Youtube

TikTok

LinkedIn

Discord

Condividi questa pagina!

Link della repo di questo LinkTree

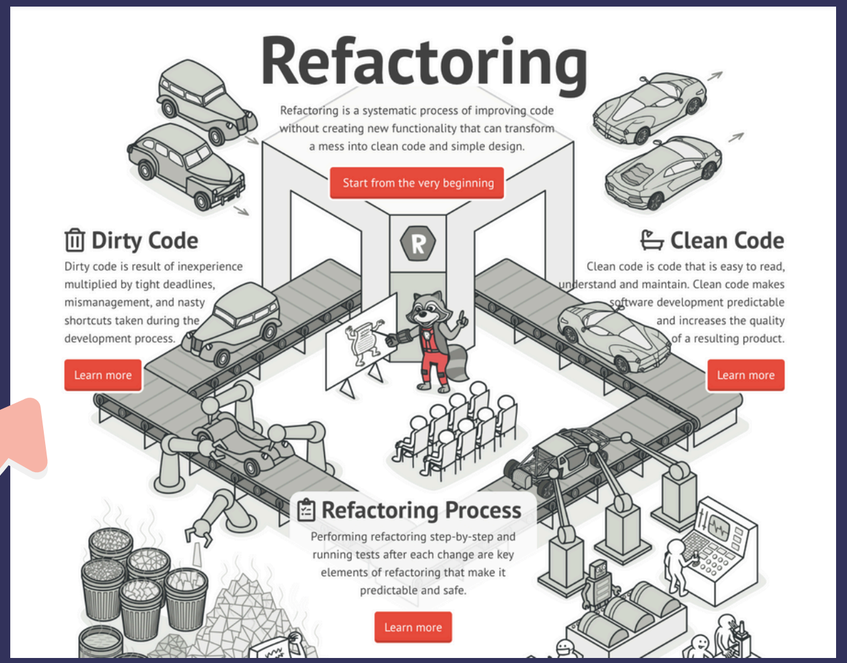
Buon codice devs!

ORA INIZIAMO CON I DESIGN PATTERN!!

ISPIRAZIONE
DI QUESTO TALK

Bacaro Tech
CODE AND FUN

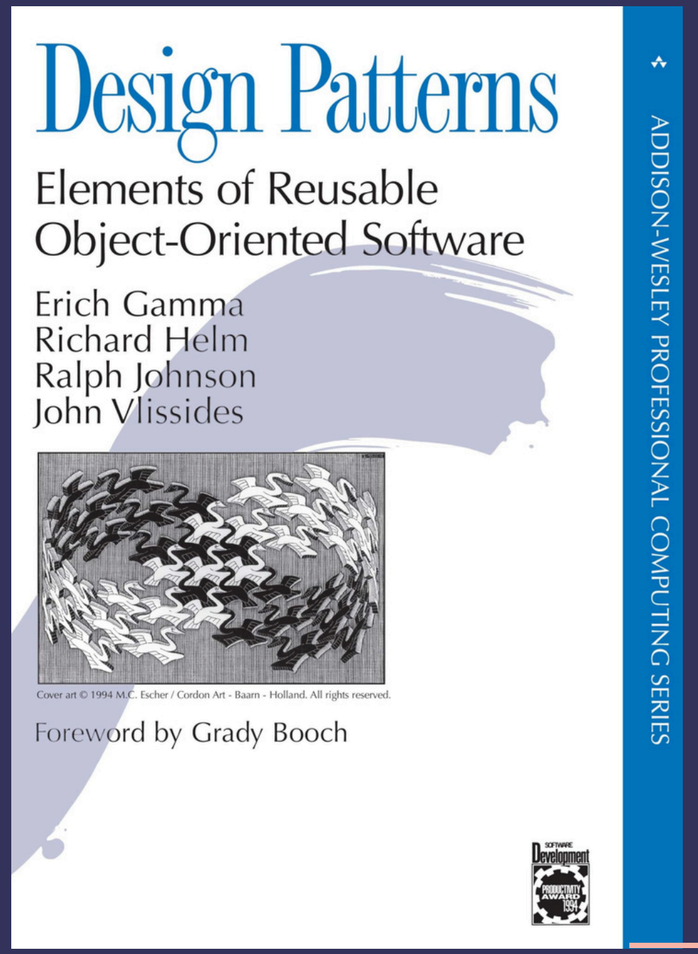
refactoring.guru



DESIGN PATTERN
CHE COSA SONO?

Bacaro Tech
CODE AND FUN

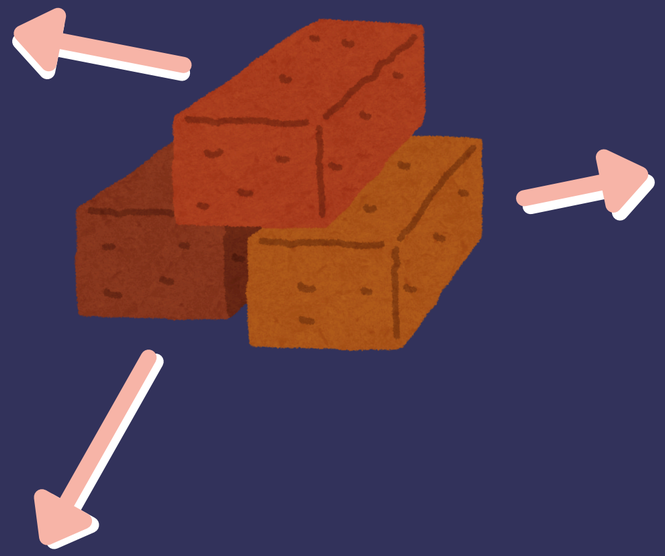
I **design pattern** sono soluzioni tipiche a problemi comuni nella progettazione software. Essi sono come un template che puoi personalizzare per risolvere problemi di progettazione del codice.



DESIGN PATTERN
3 FAMIGLIE

Bacaro Tech
CODE AND FUN

Creational patterns
Meccanismi di creazione di oggetti, che aumentano la flessibilità e il riutilizzo del codice.



Structural patterns
Spiegano come assemblare oggetti e classi in strutture più grandi, mantenendole flessibili ed efficienti.

Behavioral patterns
Riguardano gli algoritmi e l'assegnazione delle responsabilità tra oggetti.

CREATIONAL PATTERNS
CHI NE FA PARTE?

Bacaro Tech
CODE AND FUN

Creational Design Patterns

Factory Method	Abstract Factory	Singleton
Prototype	Builder	

STRUCTURAL PATTERNS
CHI NE FA PARTE?

Bacaro Tech
CODE AND FUN

Structural Design Patterns

Adapter	Proxy	Bridge	Facade
Composite	Decorator		Flyweight

BEHAVIORAL PATTERNS
CHI NE FA PARTE?

Bacaro Tech
CODE AND FUN

Behavioral Design Patterns

Chain Responsibility		Strategy	Visitor
Mediator	State	Iterator	Observer
Command	Memento	Template Method	

DESING PATTERNS
CHE FATICA!!

Bacaro Tech

CODE AND FUN

Il numero totale di desin patter corrisponde a 22, e farli tutti oggi sarebbe impossibile, oltre a essere poco produttivo

DESING PATTERNS
COMPROMESSO

Bacaro Tech

CODE AND FUN

Un compromesso utile è concentrarsi, in questa sede, su **10 dei 22 design pattern più noti.** Si tratta di quelli più diffusi e utili nella “programmazione quotidiana”.

Perchè? In modo tale da avere una solida base partendo dai pattern che ci troveremo più spesso a implementare e lasciarvi a voi esplorare questo modo in completa autonomia.

DESING PATTERNS

TOP 10 DESING PATTERN

Bacaro Tech

CODE AND FUN

1. Singleton
2. Factory Method
3. Builder
4. Adapter
5. Decorator
6. Facade
7. Observer
8. Strategy
9. Command
10. State

Creational Patterns 🔧

Structural Patterns 🧩

Behavioral Patterns 🧠

SINGLETON
CHE COS’E’?

Bacaro Tech
CODE AND FUN

Definizione:
Garantisce che una classe abbia una sola istanza e fornisce un punto di accesso globale a essa.

Implementazione:

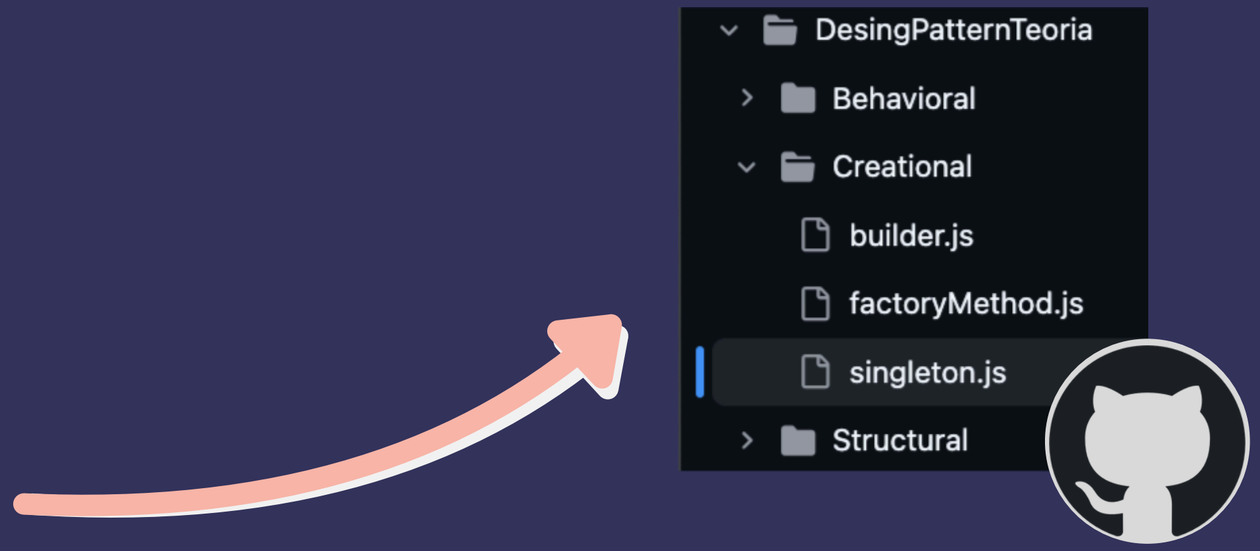
1. Costruttore privato → impedisce la creazione di nuove istanze dall’esterno.
2. Metodo statico → restituisce sempre la stessa istanza della classe.

Esempi d’uso:
Logger, Connessioni al database, Configurazioni globali

SINGLETON
VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



FACTORY METHOD 🔧

CHE COS'E'?

Bacaro Tech
CODE AND FUN

Definizione:
Fornisce un'interfaccia per creare oggetti in una superclasse, ma lascia alle sottoclassi la decisione su quale tipo di oggetto istanziare.

Implementazione:

- Metodo factory → definito nella classe base e sovrascritto dalle sottoclassi.
- Creazione delegata → il codice client usa il metodo factory senza conoscere i dettagli di costruzione.

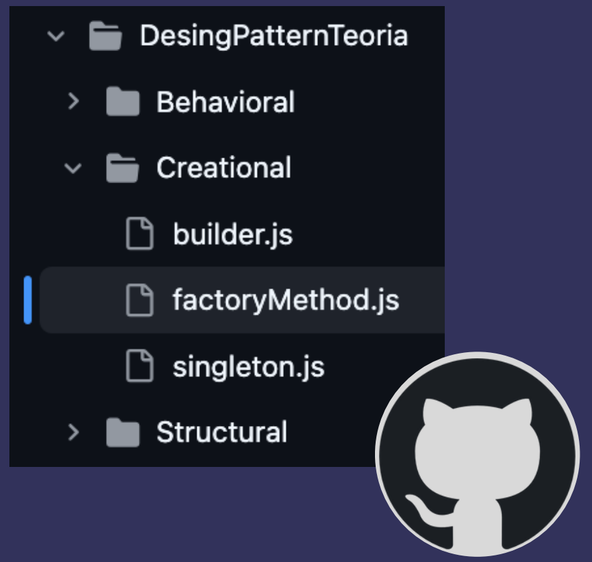
Esempi d'uso:
Creazione di documenti vari, Parser per formati differenti

FACTORY METHOD

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



BUILDER
CHE COS'E'?

Bacaro Tech
CODE AND FUN

Definizione:
Permette di costruire oggetti complessi passo dopo passo, separando la costruzione dalla rappresentazione finale.

Implementazione:

- Builder → classe dedicata che costruisce l'oggetto in modo modulare.
- Metodi fluenti e build() → permettono di configurare l'oggetto con chiamate concatenate e infine costruirlo

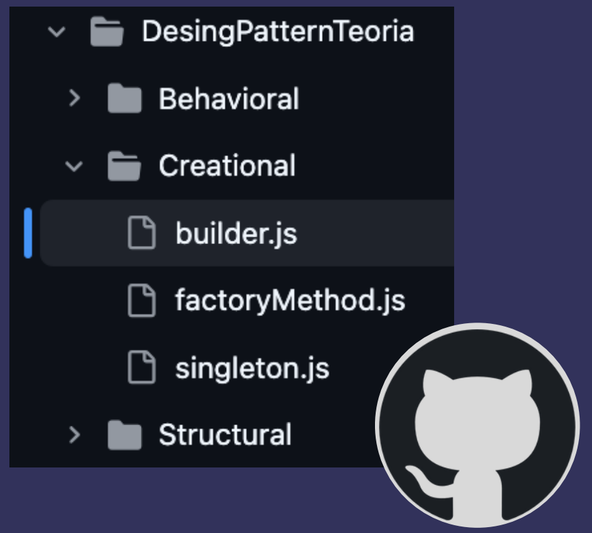
Esempi d'uso:
Costruzione di oggetti con molti parametri opzionali,
Creazione di configurazioni di sistema flessibili

BUILDER

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



ADAPTER
CHE COS'E'?

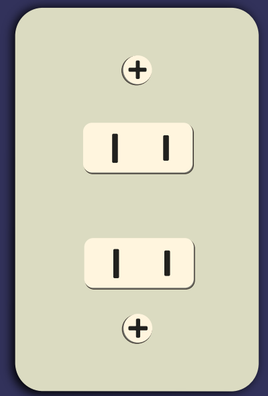
Bacaro Tech
CODE AND FUN

Definizione:
Permette a classi con interfacce incompatibili di collaborare, facendo da "ponte" che traduce le chiamate tra interfacce diverse.

Implementazione:

- Adapter → classe che implementa l'interfaccia target e incapsula l'oggetto esistente (adaptee).

Esempi d'uso:
Integrazione di librerie di terze parti, migrazione API o uniformazione di formati dati diversi (es. XML a JSON).

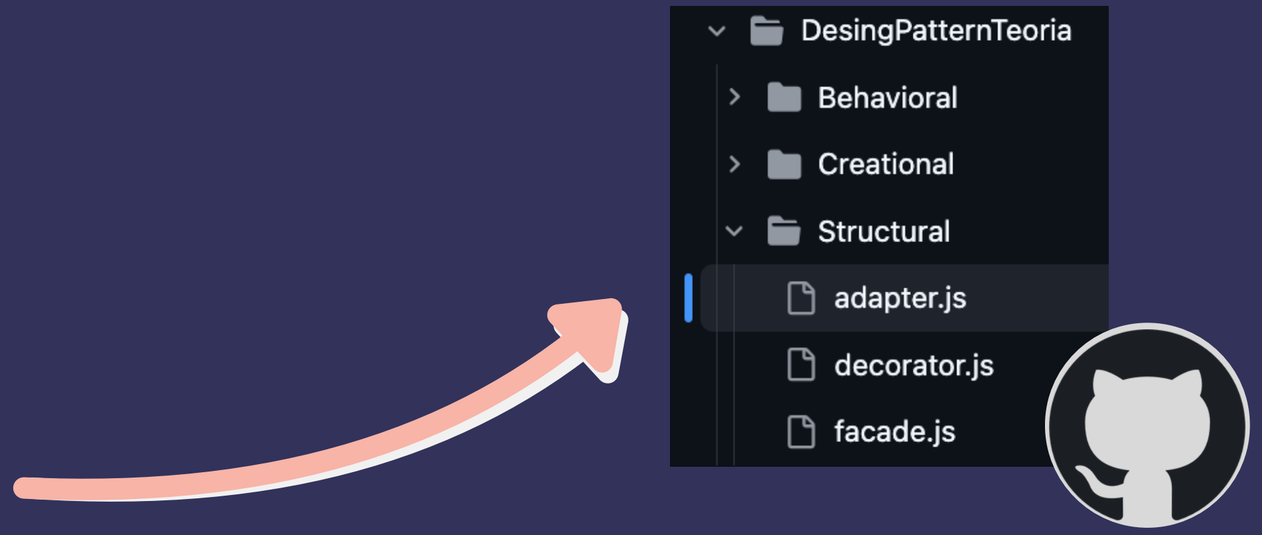


ADAPTER

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



DECORATOR
CHE COS'E'?

Bacaro Tech
CODE AND FUN

Definizione:
Permette di aggiungere dinamicamente funzionalità a un oggetto senza modificarne la classe, "impacchettando" l'oggetto originale in un wrapper

Implementazione:

- Componente → base da wrappare
- Estensione dinamica → componente che ha il wrapping e operazioni aggiuntive

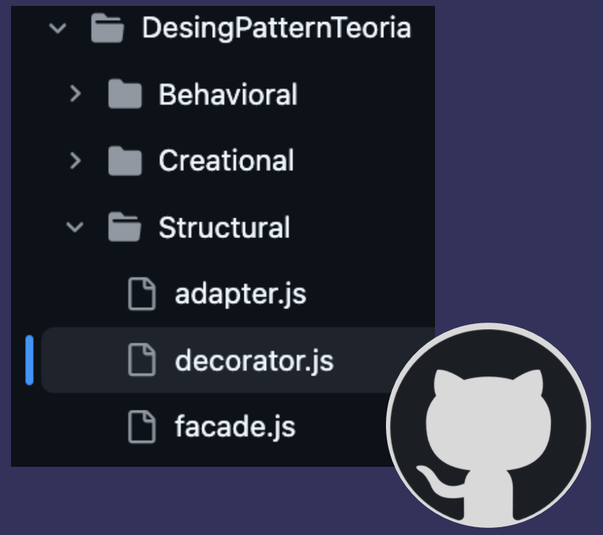
Esempi d'uso:
Estendere componenti UI con stili o funzionalità aggiuntive senza creare molte sottoclassi specifiche.

DECORATOR

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



FACADE
CHE COS'E'?

Bacaro Tech
CODE AND FUN

Definizione:
Fornisce un'interfaccia semplificata per accedere a un insieme complesso di classi o sottosistemi.

Implementazione:

- Classe Facade → espone metodi semplici che incapsulano chiamate complesse a più componenti interni.
- Delegazione → il Facade coordina le operazioni tra i sottosistemi.

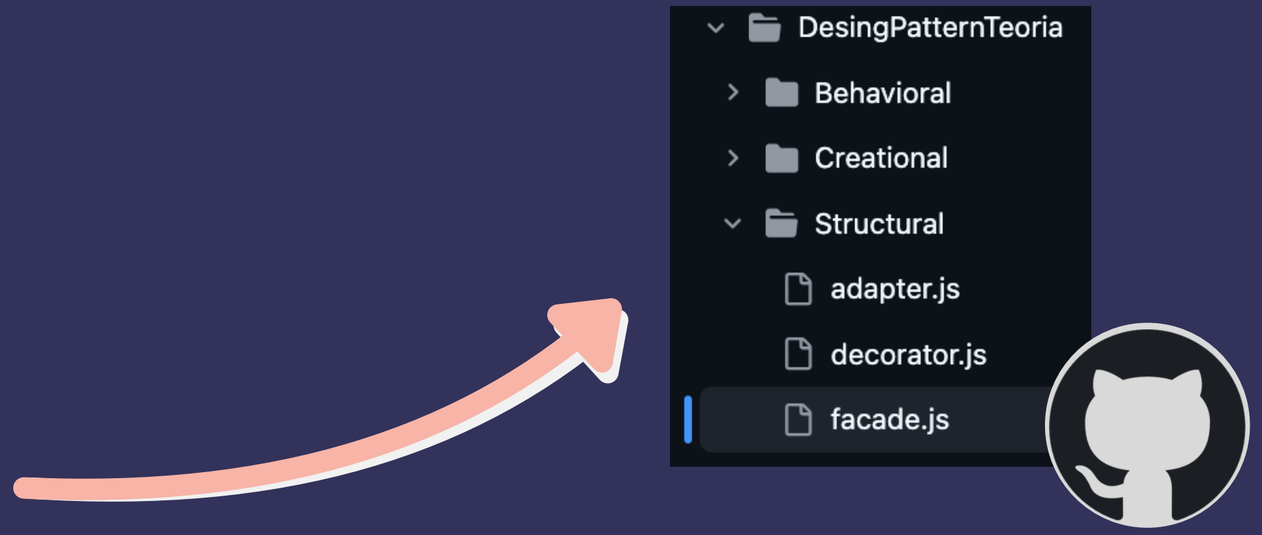
Esempi d'uso:
Un'unica API per gestire funzioni di più librerie

FACADE

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



OBSERVER
CHE COS'E'?

Bacaro Tech
CODE AND FUN

Definizione:
Permette a un oggetto (Subject) di notificare automaticamente più oggetti (Observer) quando il suo stato cambia.

Implementazione:

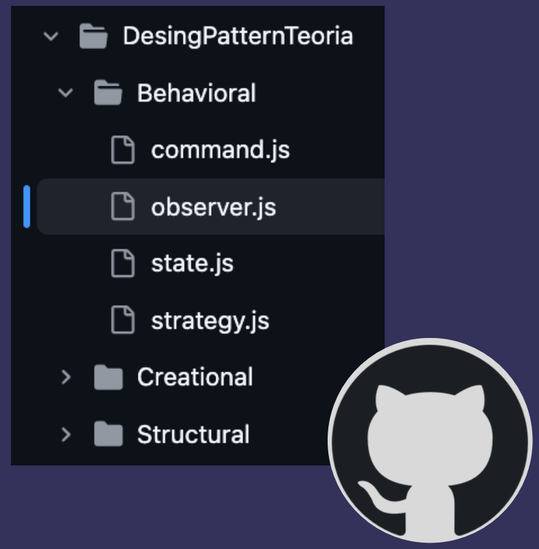
- Subject → mantiene la lista degli observer e gestisce la registrazione/rimozione.
- Observer → implementa un metodo di aggiornamento chiamato dal Subject.

Esempi d'uso:
Sistemi di eventi, notifiche in tempo reale

OBSERVER
VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



STRATEGY
CHE COS'E'?

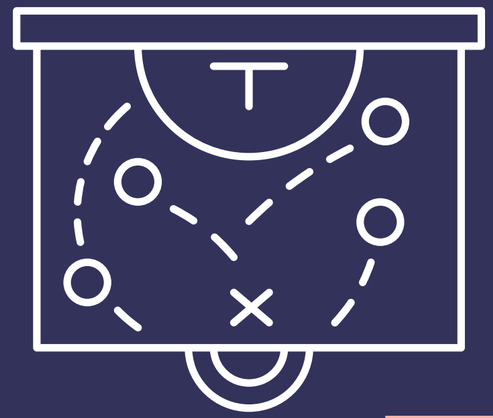
Bacaro Tech
CODE AND FUN

Definizione:
Permette di definire una famiglia di algoritmi, incapsularli e renderli intercambiabili a runtime senza modificare il contesto che li utilizza.

Implementazione:

- Interfaccia comune → definisce il metodo dell'algoritmo.
- Strategie concrete → implementano l'algoritmo
- Contesto → utilizza una strategia dinamica

Esempi d'uso:
Sistemi di pagamento con metodi diversi, ordinamento con algoritmi differenti, compressione di file con formati vari.

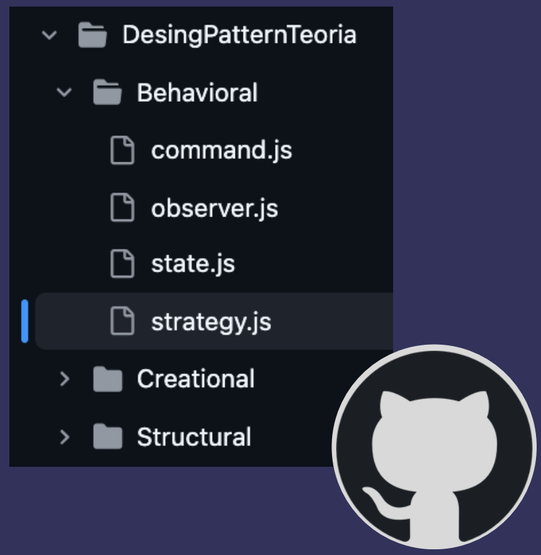


STRATEGY

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



COMMAND
CHE COS’E’?

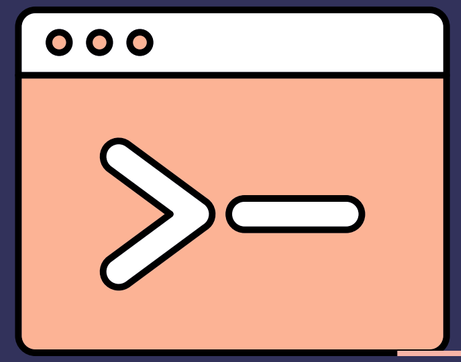
Bacaro Tech
CODE AND FUN

Definizione
Incapsula una richiesta come oggetto, consentendo di parametrizzare i client con operazioni diverse.

Implementazione:

- Interfaccia Command → definisce il metodo execute().
- Comandi concreti → implementano l’azione da eseguire.
- Invoker → chiama il comando senza conoscere i dettagli dell’operazione.
- Receiver → l’oggetto che esegue realmente l’azione.

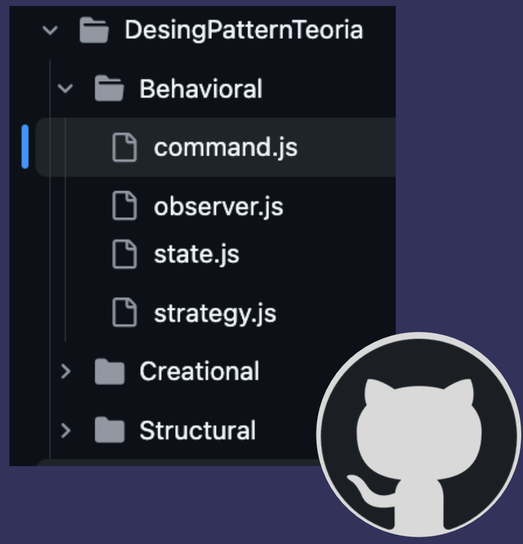
Esempi d’uso:
Gestione di operazioni undo/redo in un editor di testo, esecuzione di macro, code di task asincroni.



COMMAND
VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



STATE
CHE COS'E'?

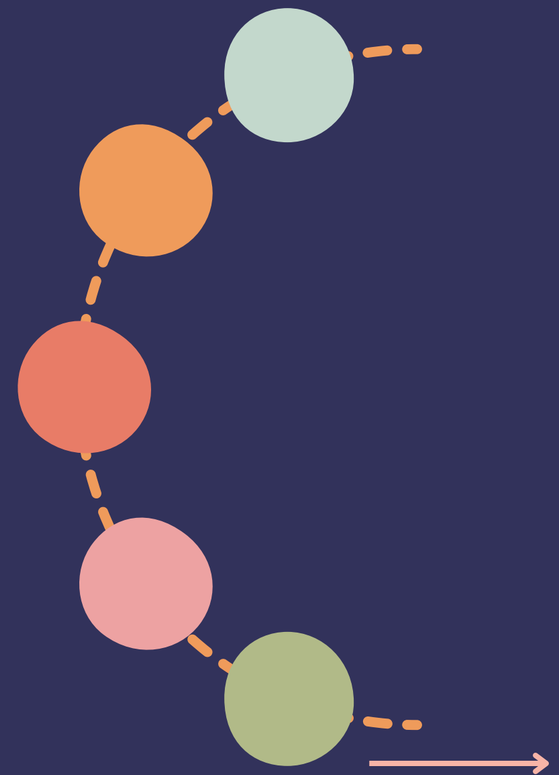
Bacaro Tech
CODE AND FUN

Definizione
Permette a un oggetto di modificare il proprio comportamento quando cambia il suo stato interno

Implementazione:

- Interfaccia State → definisce cosa accomuna gli stati
- Stati concreti → implementano il comportamento vero
- Contesto → mantiene un riferimento allo stato corrente e delega a esso le operazioni.

Esempi d'uso:
Macchine a stati finiti come un distributore automatico, gestione di sessioni utente o workflow con fasi diverse.

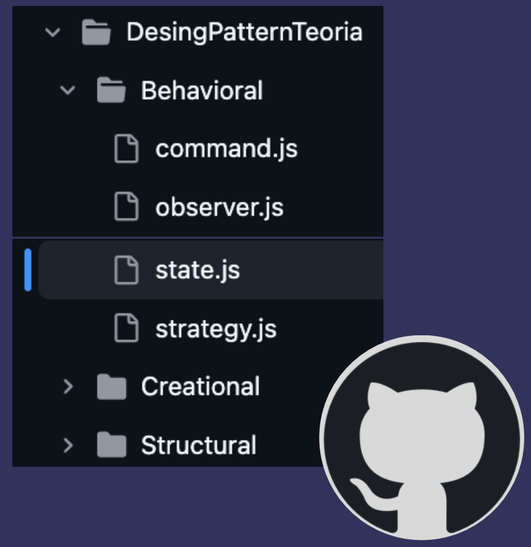


STATE

VEDIAMO IL CODICE!

Bacaro Tech
CODE AND FUN

Bacaro-tech-gym



DESIGN PATTERN
CONCLUSIONI

Bacaro Tech
CODE AND FUN

Abbiamo esplorato solo una parte dell’universo dei design pattern, eppure già questi esempi ci rivelano quanto siano **strumenti potenti e versatili** per risolvere problemi ricorrenti nello sviluppo software.

Vantaggi generali:
✅ Migliorano la leggibilità del codice
✅ Aumentano la manutenibilità
✅ Favoriscono il riutilizzo, evitando di reinventare la ruota ogni volta.
✅ Creano un linguaggio comune tra sviluppatori, migliorando la collaborazione e la codebase.

Bacaro
Tech

CODE AND FUN

VI RINGRAZIA TUTTI PER AVER PARTECIPATO!

ASPETTIAMO VOSTRE DOMANDE

BACAROTECH

SEGUITECI!

Bacaro Tech

CODE AND FUN

BacaroTech

BacaroTech

Code and Fun

La tua **community di sviluppatori** dove si parla di programmazione a 360°: strutture dati, algoritmi, carriera tech e molto altro!

- Canale Whatsapp
- GitHub
- Instagram
- Youtube
- TikTok
- LinkedIn
- Discord

Condividi questa pagina!

Link della repo di questo LinkTree

Buon codice devs!