Ibtissam
Oulahchoum

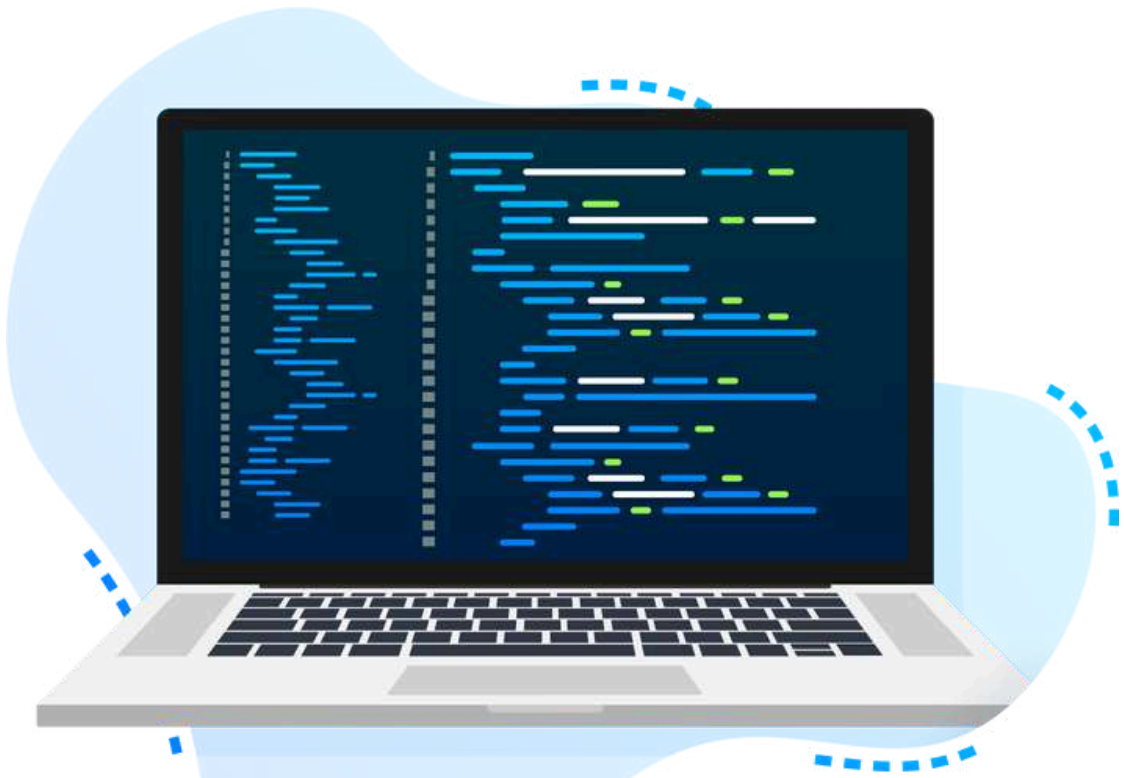
Types of Software Design Patterns

English and Moroccan Darija

Types of Design Patterns

Besmellah, ghadi nbdaw b 3 types principal dial design patterns:

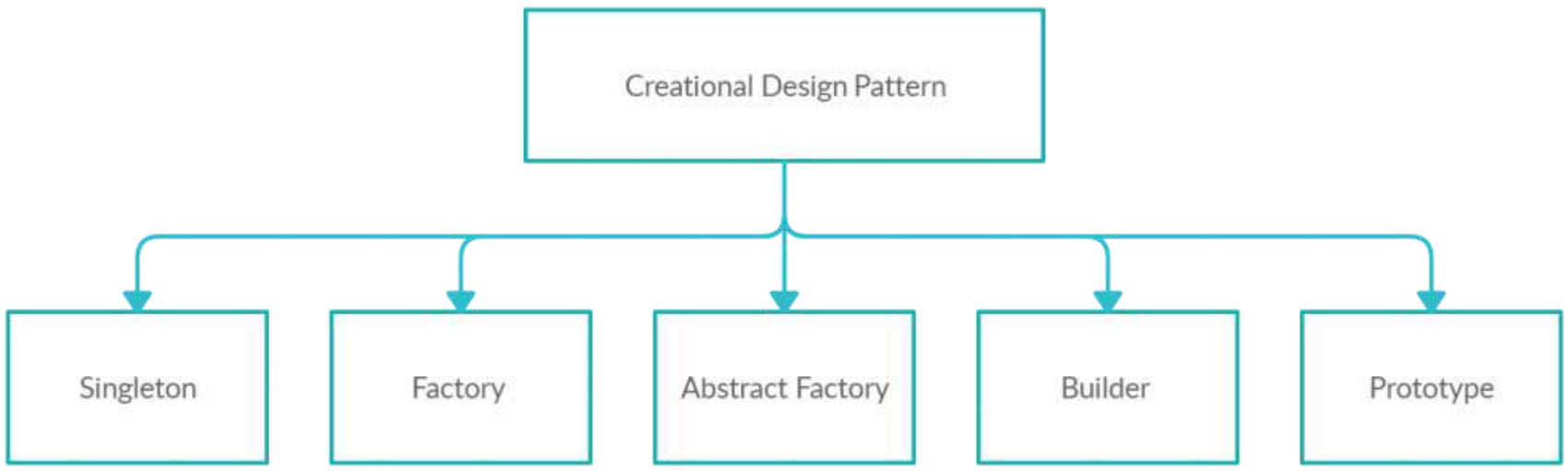
- **Creational** Design Pattern
- **Structural** Design Pattern
- **Behavioral** Design Pattern



Creational Design Pattern

Creational Design Patterns focus on the process of object creation or problems related to object creation. They help in making a system independent of how its objects are created, composed and represented.

Hadi kthdar 3la kifach kan cryiw objects o classes f code dyalna.



Types of Creational Design Pattern

Singleton: One class, one instance, and universal access. It’s the ultimate way to ensure there’s just one global go-to for your needs.

Factory Method: Need an object but don’t want to know its exact type? The Factory Method is your friend—it builds objects without exposing their creation details.

Abstract Factory: Imagine a factory of factories! It’s like a Factory Method leveled up, creating families of related objects without specifying their actual classes.

Builder: Think of it as a step-by-step recipe for creating intricate objects. You focus on the process, and the Builder ensures you get the desired result every time.

Prototype Method Why reinvent the wheel? Clone an existing object to save time and avoid complex creations. It’s efficient and cost-effective.

Anwa3 dial Creational Design Pattern

Singleton:

Hada design pattern li kay5alik t5dam b object wahd unique f program kamlo Hada concept dialo ana baghi ndir t3rif l instance mn classe kaytst3ml ela nita9 l app kaml bhal fash kadir json object o t7to f file f javascript o f oop static classe o t3rf wsto matalan database connection blabla , functions kifash tconecta mea database bla bla so itst3ml instance whda f app kamla

Factory Method:

Imagine 3ndk shi fabrique , nta ma3ndkch darouri t3ref kifach kay5dm dakchi li kayt9ad fiha.
Mtalan Bhal f chi restaurant, nta katcommandi pizza, machi darouri t3ref kifach kat9ad o lmowasafat dial lmokwinat . factory classe hwa ay3yt ela function had function smytha assemble(); o ms2oulytha hya tkwn o trkb o tkhlt ga3 dok lmokwinat d pizza b call whdaa . Khoulassa hwa 3ibara 3la wahd l pattern katkhdmo bach ay haja kat3awd ki kat9ad ola bach kat9ad yb9a 3ndk f code ka classe ola ka object

Anwa3 dial Creational Design Pattern

Abstract Factory: Hadi bhal Factory walakin advanced, bhal chi group dial fabrikat. Matalan, 3ndk group dial meubles (tables, chaises...). Kola style 3ndo l'fabrique dyalo.

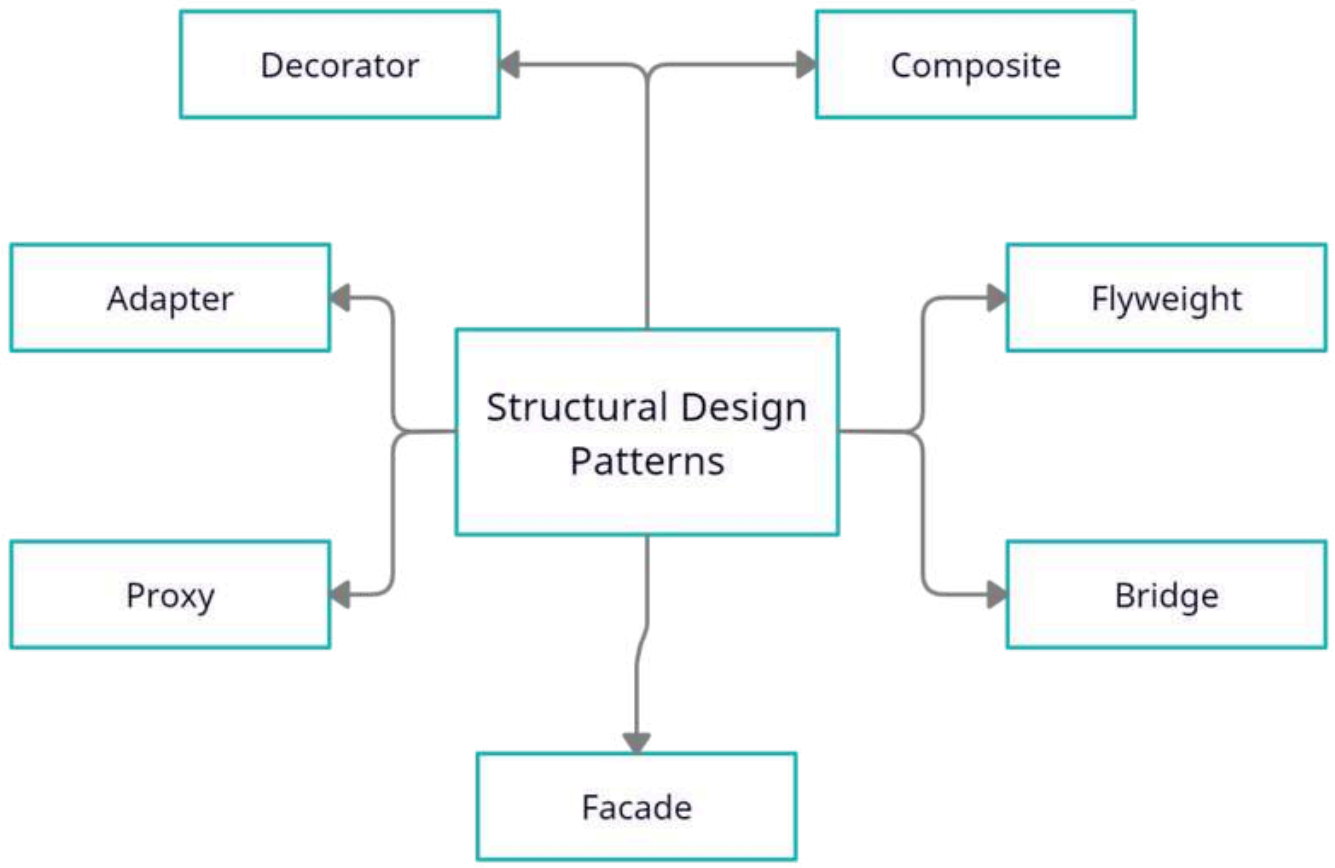
Builder: Bhal recipe dial tyab . Katchof steps b steps bach dir chi haja . Bhal creation dial pizza: zid sauce, zid fromage, zid zitoune...
Katji f nfs class o nta endk assemble() li hwa kysawb lik pizza o sf ms hya kat3yt l builder o hwa li kygol f step lwla 3jn pizza b3da , dirha f fran , wjd la sauce , blab la mhm step by step

Prototype Method :Hada bhal copied w paste. 3wad ma t crea chi object men zero, kancopyiw object li kayn w kan3awdo nmodifyiw 3lih.

Structural Design Patterns

Structural Design Patterns solves problems related to how classes and objects are composed/assembled to form larger structures which are efficient and flexible in nature. Structural class patterns use inheritance to compose interfaces or implementations.

☆ Hadi kayhdro 3la kifach kan organiziw code dyalna.



Types of Structural Design Patterns

Adapter: When two interfaces don’t speak the same language, the Adapter steps in as a translator, making them compatible and ready to work together.

Bridge: Separate abstraction from implementation. The Bridge lets you develop them independently, so the client only deals with the abstract layer, not the underlying details.

Composite: Treat a group of objects like a single object. Composite structures items into tree-like hierarchies, perfect for representing part-whole relationships.

Decorator: Need to add some flair? Decorator dynamically adds features to objects without touching others. Think of it as accessorizing without overhauling.

Types of Structural Design Patterns

Facade: Simplify complexity! Facade offers a single, unified interface to a subsystem, making its functionality easier and more intuitive to use.

Flyweight: Too many objects cluttering your app? Flyweight minimizes the count by sharing common parts of similar objects, optimizing memory and performance.

Proxy: A stand-in that acts on behalf of another. Proxy patterns handle access control, communication, or additional logic without changing the actual object.

Anwa3 dial Structural Design Patterns

Adapter: Bhal adaptateur dial charging. 3ndk telephone iPhone w charger Android, kaddir adaptateur bach y5dmo m3a b3diyathom.

Bridge: Kay3tik façon bach tfs5 implementation 3la abstraction. Bhal remote control: 3ndk télécommande w television, kola wahd kay5dam bohdo.

Composite: Kayt3aml m3a group dial objects bhal object wahd. Bhal dossier f PC: 9der tdir files bohdom wla dossier fih files.

Decorator: T9dr tZid features l object bla mat5arb9o. Bhal pizza, t9der tzid toppings bla matbdl pizza originale.

Types of Structural Design Patterns

Facade: Kay3tik interface bsita l system m3a9d. Bhal telecommande: button wahd kay5dm 3la operations m3a9din.

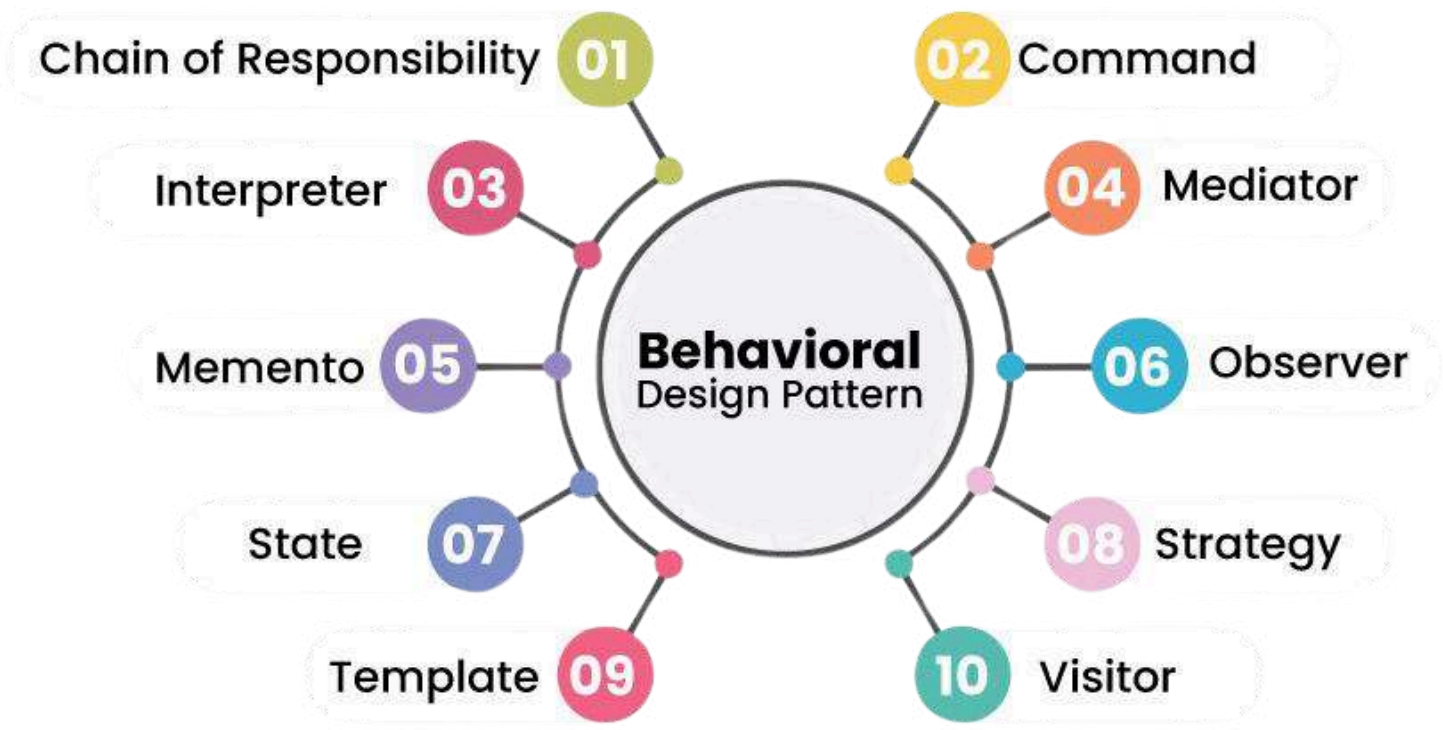
Flyweight: Kaysta3ml memory b efficient. Bhal jeu video: 3wad mat5zen kola ‘char’ dial text, kat5zen ghir wahd model w katsta3mlo several times.

Proxy: Bhal security guard. 9bal manwslo l object real, kanmchiw 3la proxy li kay controlli access.

Behavioral Design Patterns

Behavioral Design Patterns are concerned with algorithms and the assignment of responsibilities between objects. Behavioral patterns describe not just patterns of objects or classes but also the patterns of communication between them. These patterns characterize complex control flow that's difficult to follow at run-time.

Hadi kathdr 3la kifach classes kay communicaw m3a b3diyathom



Types of Behavioral Design Patterns:

Chain of Responsibility Design Pattern: This pattern allows a request to be passed through a chain of objects, where each object can decide whether to handle the request or pass it to the next in the chain. It promotes loose coupling by avoiding direct connections between the sender and receiver.

Command Design Pattern : The Command pattern encapsulates a request as an independent object, storing all necessary information about the request. This object can be passed around, stored, or executed later, enabling features like undo, redo, or deferred execution.

Interpreter Design Pattern: This pattern defines a grammar for a language and provides an interpreter to process sentences in that language. It’s often used for parsing and evaluating expressions.

Types of Behavioral Design Patterns:

Mediator Design Pattern : The Mediator pattern introduces a central object (mediator) to manage communication between other objects, reducing their direct dependencies and simplifying their interactions.

Memento Design Pattern : This pattern captures an object's state, allowing it to be restored to a previous state later. It is commonly used for creating checkpoints in applications, such as undo functionality in text editors or games.

Observer Design Pattern : The Observer pattern establishes a one-to-many relationship between objects. When one object (subject) changes, all its dependents (observers) are automatically notified and updated. It's useful for event-driven systems.

Types of Behavioral Design Patterns:

State Design Pattern : This pattern enables an object to alter its behavior when its internal state changes. Instead of using if-else or switch statements, the behavior is encapsulated in state-specific classes, making the code more flexible and easier to extend.

Strategy Design Pattern : The Strategy pattern allows you to choose an object's behavior at runtime by encapsulating a family of algorithms into separate classes that share a common interface. This makes it easy to switch between different algorithms.

Template Method Design Pattern : This pattern defines the skeleton of an algorithm in a base class, with specific steps implemented by subclasses. The base class maintains the structure, while subclasses customize the behavior for specific needs.

Anwa3 dial Behavioral Design Patterns:

Chain of Responsibility Design Pattern: Bhal service client: katbda m3a agent normal, ila mal9ach lik chi 7el kay passik l supervisor, w hakda...

Command Design Pattern : Bhal waiter f restaurant. Kay5d orders mn clients w kaywslhom l couzina Orders y9dro yt5zno wla ytlagaw.

Interpreter Design Pattern: Kayfhm w kaytarjm chi language. Bhal Google Translate.
Example akhour Matalan, ila 3ndek calculator bsit li kayfehem "PLUS 5 3" - kay9raha w kay3ref bli khasso ydir 5+3. L'Interpreter kay9sem text l ajza2 ("PLUS", "5", "3"), kay3ref wash "PLUS" wla "MOINS", w kay executi l'operation .

Anwa3 dial Behavioral Design Patterns:

Mediator Design Pattern : Had pattern kaynf3 f ay situation fin 3ndek multiple objects khashom ycommuniquiw bin b3diyathom b tri9a loose coupled, bhal: système réservation, traffic control, message brokers, etc.
Matlan L mediator pattern howa bhal l'garçon f restaurant: kay wqef bin client w cuisine bach y coordini communication binahom. 3iwed ma nkhelliw components (client w cuisine) y communiwiw directement, kay passiwha 3la mediator (garçon) li kay n'organizi l communication

Memento Design Pattern : Kay5zen state dial object bach n9dro nraj3o lih men b3d. Bhal CTRL+Z

Anwa3 dial Behavioral Design Patterns:

Observer Design Pattern : Matalan radio fih news .. subject howa li lmodi3 o l observer hwa li baghi ysme3 news f bnsba lia anmshi l subject andir lih list observer listeners y3ni li bgha ysm3ha aydir add me as a listener pls 7tni fdik list o ila jat shi new news sift ldik list
Newsletter aktr example 9rib bach tfhmou

State Design Pattern : Behavior kaytbdl 3la hsab state. Bhal bouilloire: ki tkoun ON katsa5n lma, ki tkoun OFF katwqf.

Strategy Design Pattern : Kay5alik tbdl behavior f runtime. Bhal GPS: t9der tbdl (walking, driving, bus) f ay wa9t.

Template Method Design Pattern : Kay3tik squelette dial algorithm, w subclasses kay implementiw details. Bhal recipe: kola wahd 3ndo tari9a dyalo bach y3ml cake.

Anwa3 dial Behavioral Design Patterns:

Visitor Design Pattern : Kay5alik tzid operations l objects bla matbdl code dyalhom. Bhal inspector: kaydor 3la kola department w kaydir inspection

Ibtissam Oulahchoum