

INTENTS:

ADAPTER:

Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.

COMPOSITE:

Compose objects into tree structures to represent part whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.

COMMAND:

Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations.

ITERATOR:

Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

OBSERVER:

Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

ADAPTER PATTERN:

- Explanation: It's like using a special connector to make different things work together, even if they have different shapes.
- Example: Using an adapter to charge your phone in a foreign country with different plug types.

COMPOSITE PATTERN:

- Explanation: Imagine putting Lego pieces together to make something bigger. You can treat the small pieces and the big creation as if they were the same.
- Example: Creating complex shapes by combining simple shapes like circles and rectangles in a drawing program.

COMMAND PATTERN:

- Explanation: Think of a remote control for your TV. Each button press represents a command that tells the TV what to do. You can also keep a list of commands to replay or undo them.
- Example: Pressing a button to turn the TV on or off, and then being able to undo that action.

ITERATOR PATTERN:

- Explanation: It's like reading a book one page at a time, without needing to know how the book is structured on the inside.
- Example: Going through a list of items one by one without worrying about the list's internal structure.

OBSERVER PATTERN:

- Explanation: Imagine you have a bunch of friends, and you want to tell them when you have something exciting to share. They'll automatically know when you're excited and what it is.
- Example: Notifying your friends when you post a new photo on social media, and they get a notification without checking your profile.

SENERIO

1. You have a legacy system that expects data in a specific format, but you want to use a new third-party library that provides data in a different format. You need to create an adapter to bridge the gap between the old and new data formats.
2. You are building a graphic design application that allows users to create complex shapes, including compound shapes that consist of multiple simple shapes. You want to be able to treat both simple and compound shapes uniformly in your application.
3. You have a collection of items, and you want to provide a way to traverse the collection without exposing its internal structure. You also want to be able to iterate over the items using different traversal methods.
4. You are building a weather monitoring application, and you want various displays to automatically update when the weather conditions change. The displays should be loosely coupled to the weather data, so they can be added or removed without affecting other parts of the system.
5. You are designing a remote control for a smart home system. The remote control should be able to execute various actions on different devices (e.g., turning on/off

lights, adjusting thermostat, etc.) without having to know the specifics of each device's operation.

ANSWERS:

1. ADAPTER PATTERN
2. COMPOSITE PATTERN
3. ITERATOR PATTERN
4. OBSERVER PATTERN
5. COMMAND PATTERN