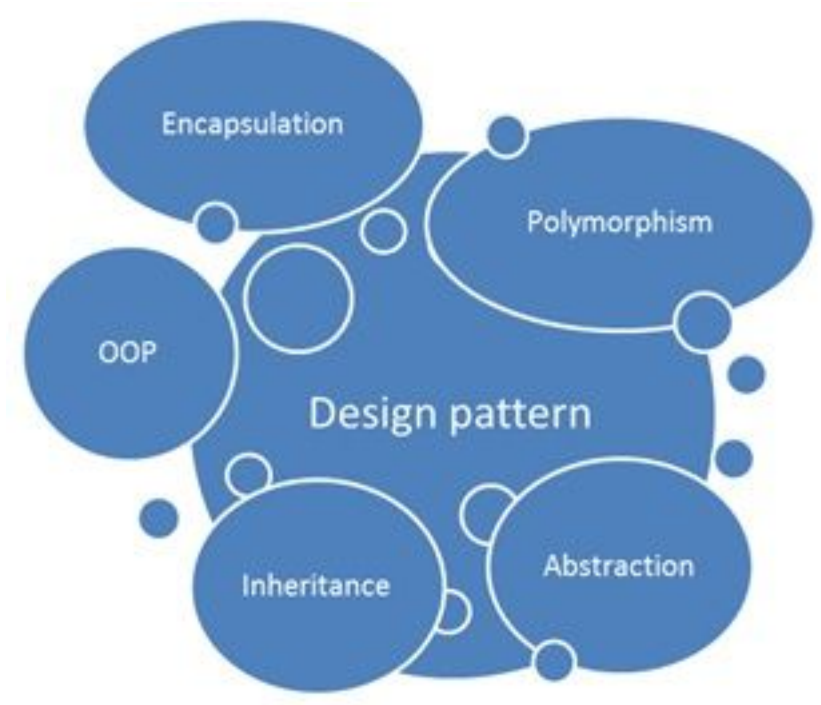
GAME MAKING

Lesson 04

NGUYỄN VĂN CƯỜNG

CONTENT

Concept



- ❖ What is the design pattern?
  - ➢ The core of solution for a common problem in software design.
  - ➢ Description of communicating objects and class.
  - ➢ Can be transformed directly to source code.

Concept

- Why we use design pattern?
  - Speed up the development process.
  - Provide proven development paradigms.
  - Reuse and extend programs.
  - Improve code readability for coders and architects.
  - Allow developers to communicate using well-known, well understood names for software interactions.

Type of design pattern

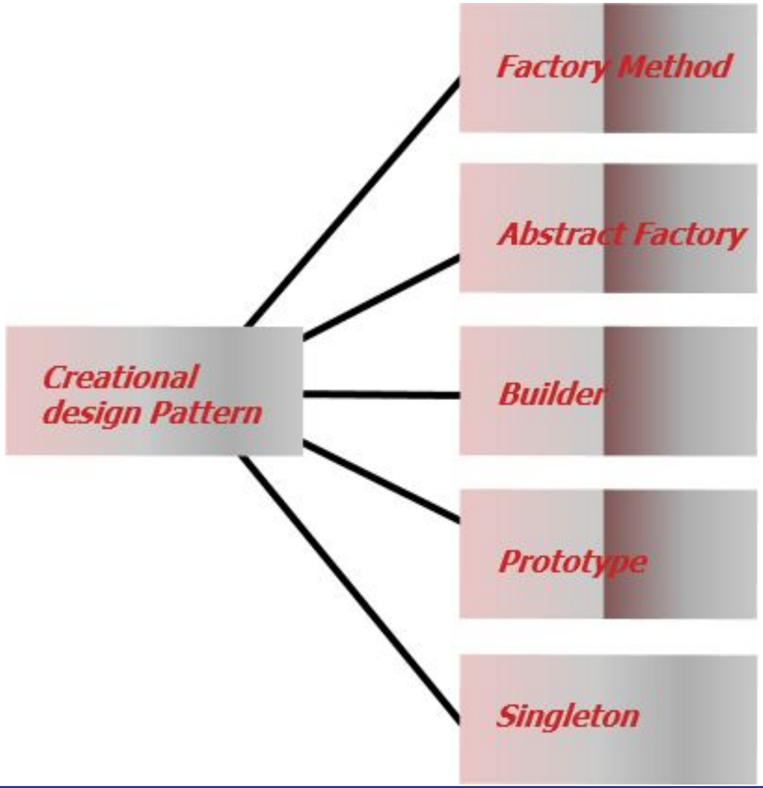
Creational

Structural

Behavioral

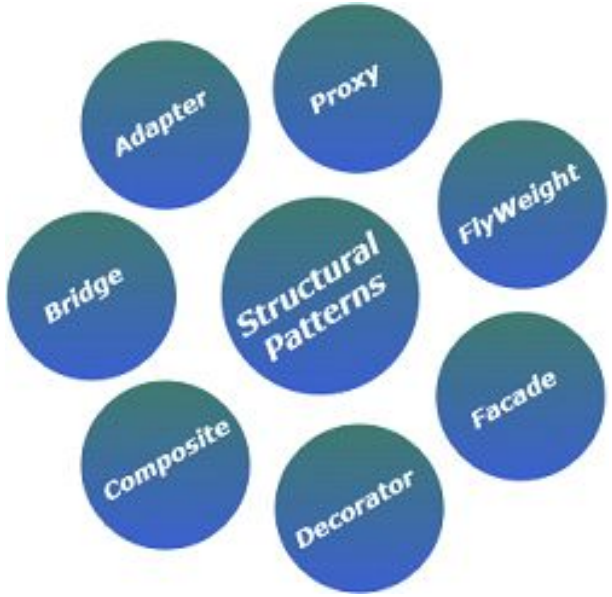
.....

Creational design pattern



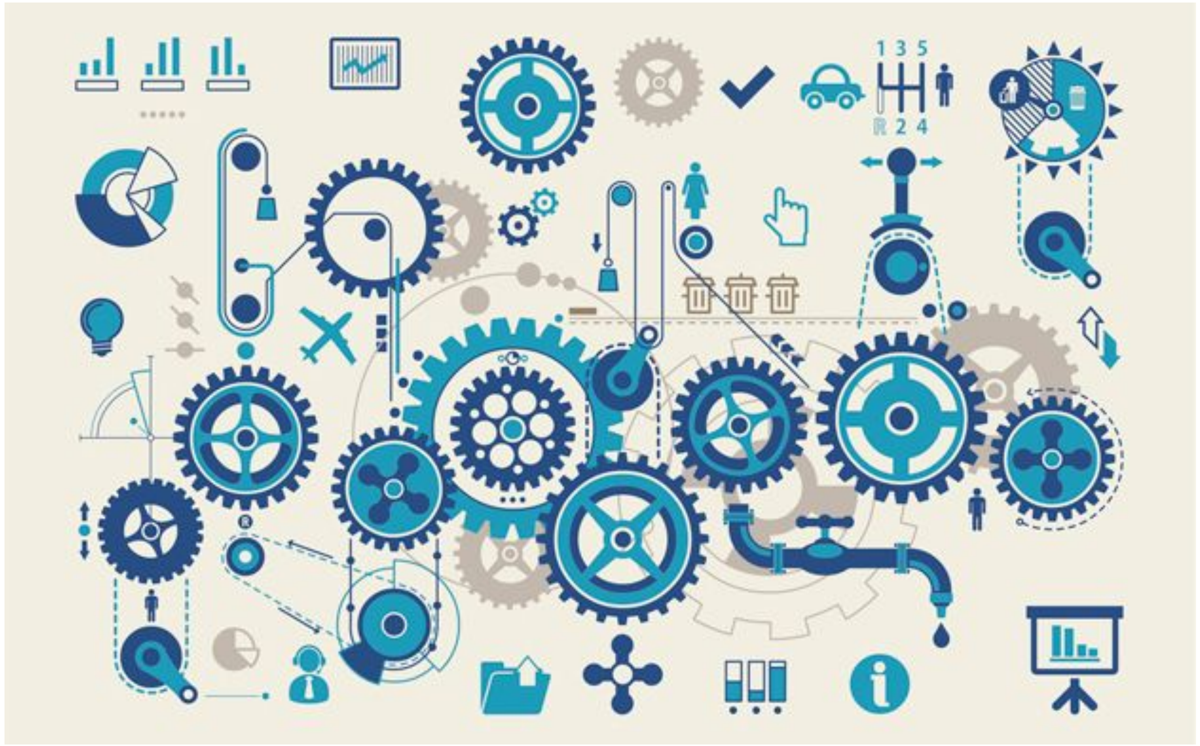
- Create objects for you, rather than having you instantiate objects directly.
- Gives your program more flexibility in deciding which objects need to be created for a given case.

Structural design pattern



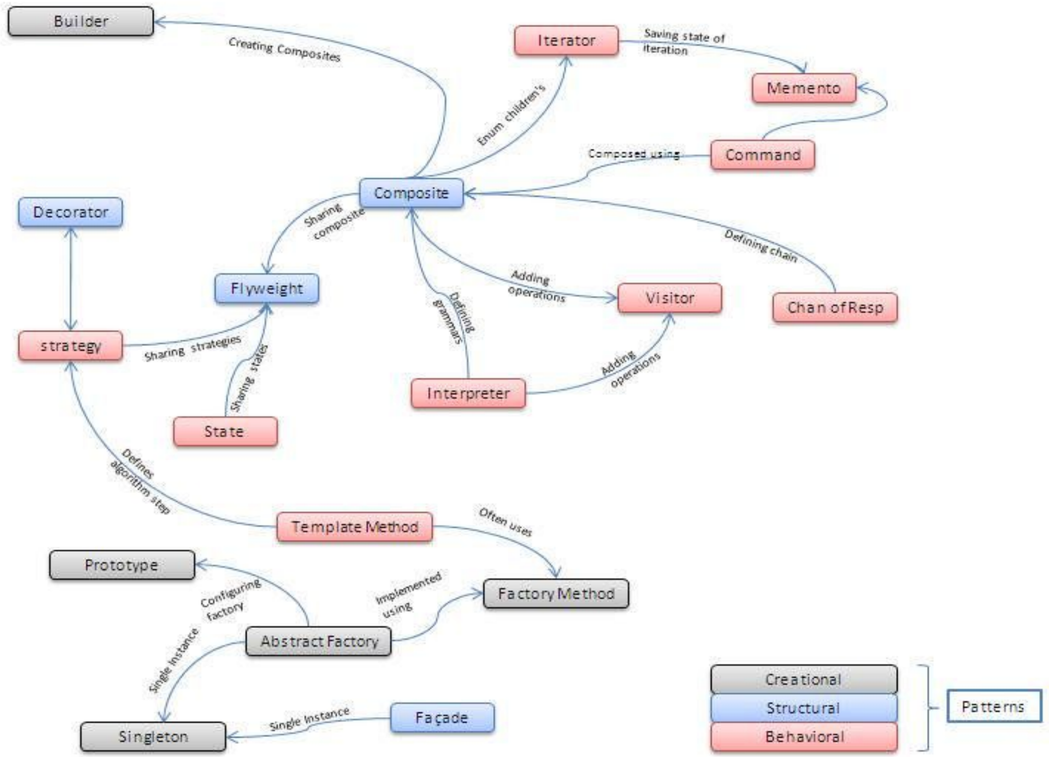
- ❖ Deal with the composition of classes or objects.
- ❖ Compose interfaces and define ways to compose objects to obtain new functionality.

Behavioral design pattern



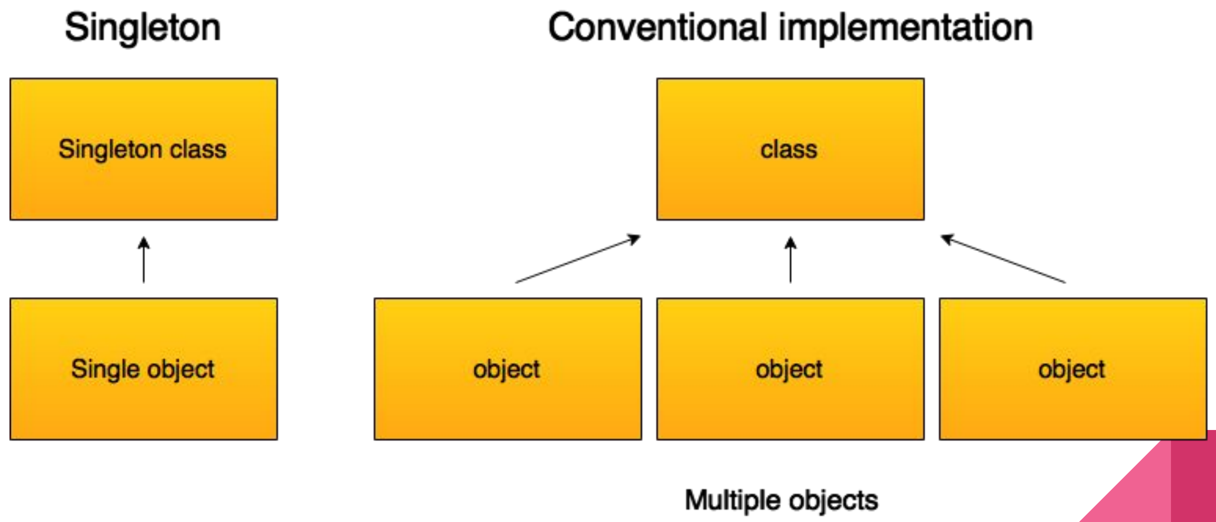
❖ Characterize the ways in which classes or objects interact and distribute responsibility.

Relationship between design pattern



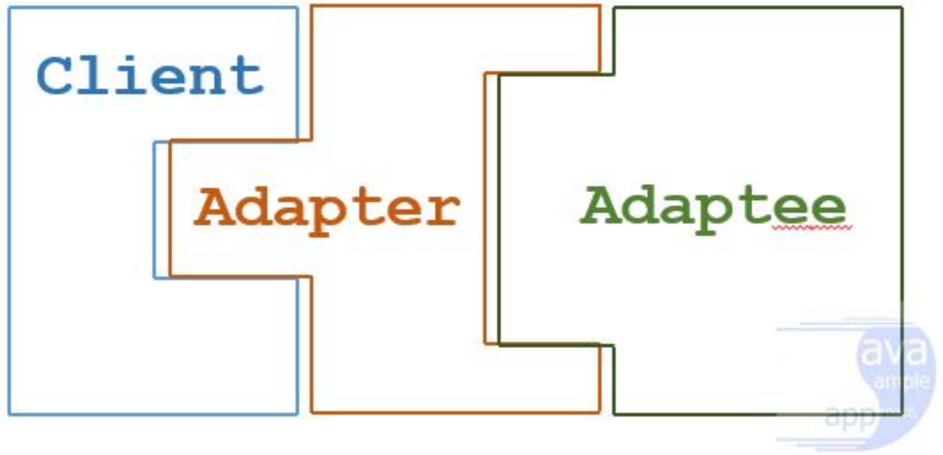
Example

- Singleton
  - Ensure a class only has one instance, and provide a global point of access to it.



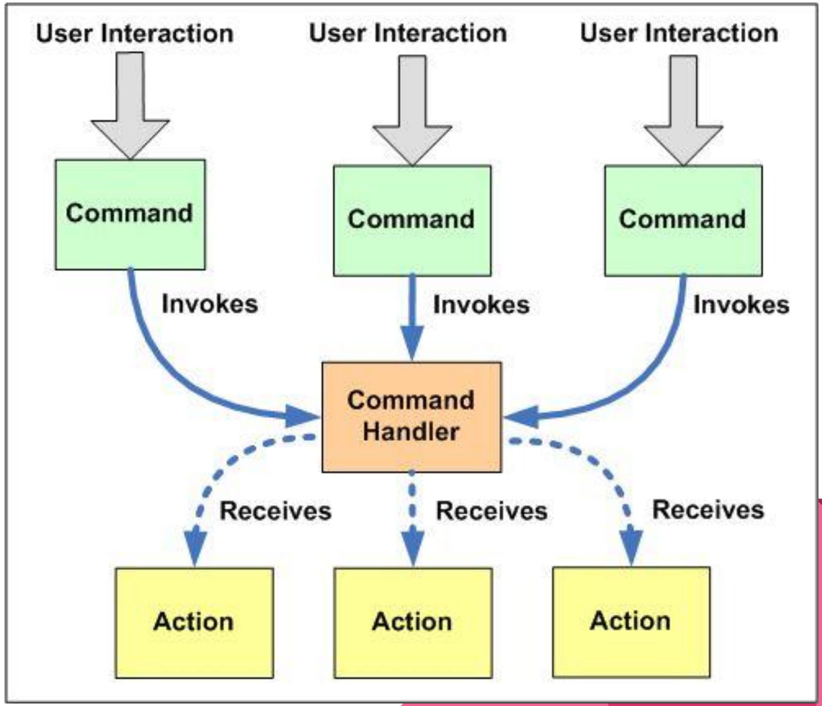
Example

- 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.



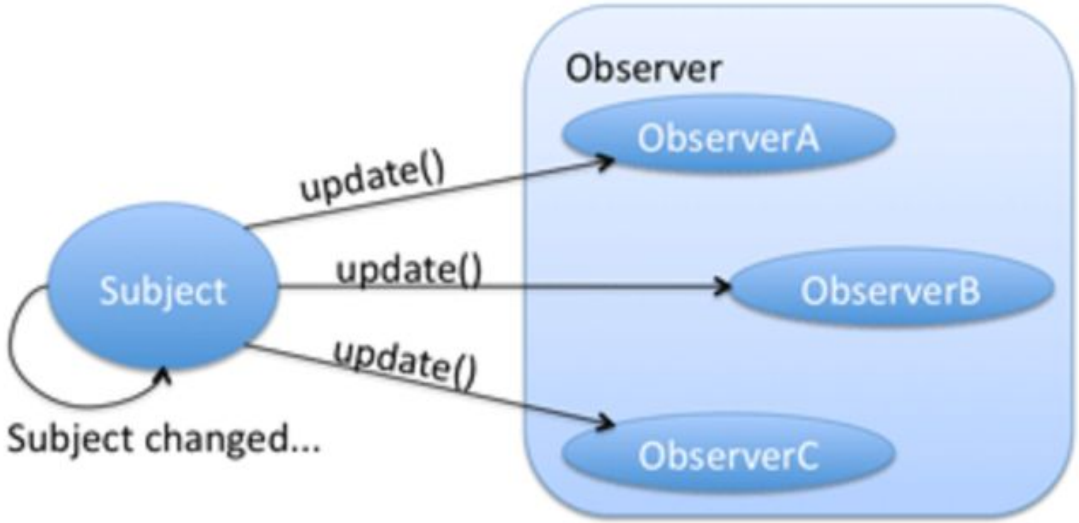
Example

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

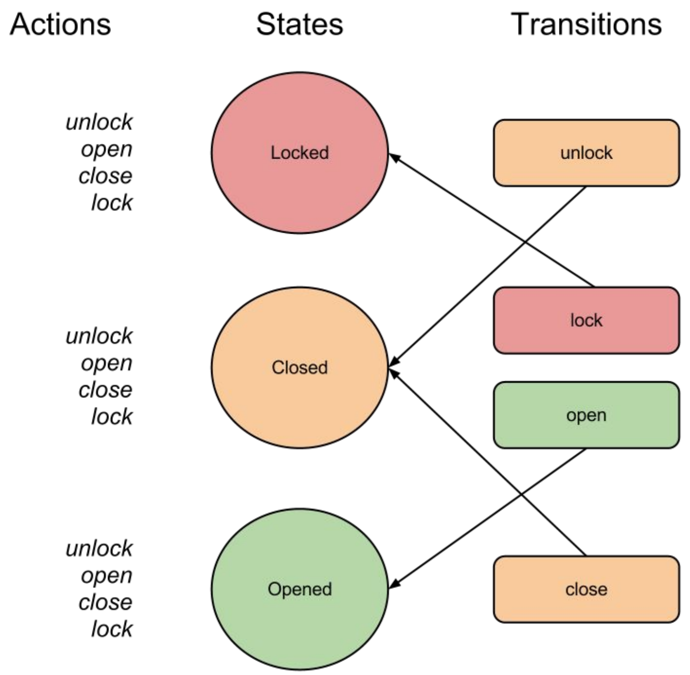


Example

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



Example



- State
  - Allow an object to alter its behavior when its internal state changes. The object will appear to change its class

Assignment

- Using state design to write a program handle a Door.

Q&A

-

Reference

- https://gameprogrammingpatterns.com/contents.html
- https://sourcemaking.com/design_patterns
-