

Most Used Design Patterns Cheat Sheet

Creational Patterns

Used to construct objects

Structural Patterns

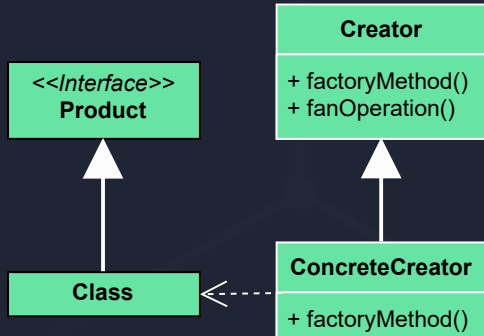
Used to form large object structures

Behavioral Patterns

Used to manage algorithms and relationships

Factory Method

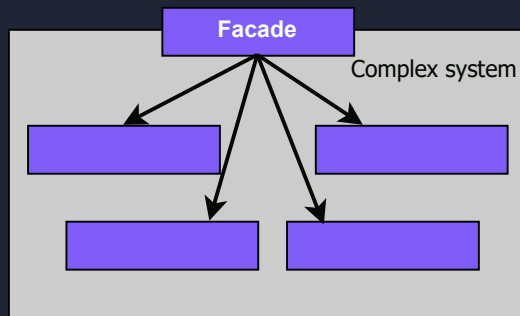
Use when you want to delegate object creation to subclasses.



Example: create GUI component

Facade

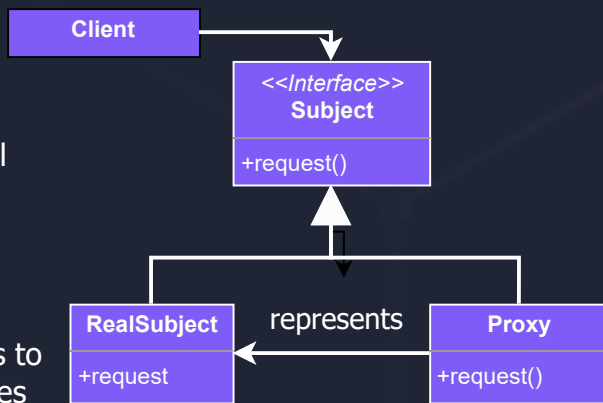
Use when you want to provide a simplified interface to a complex subsystem



Example: Providing a simple interface to a complex subsystem

Proxy

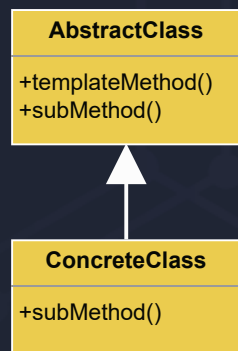
Use for object access control



Example: Controlling access to sensitive resources

Template Method

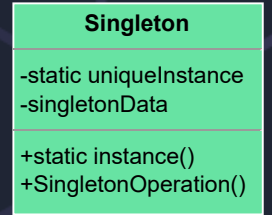
Use when you want to break down an algorithm into a series of steps



Example: Common behavior should be located in one class

Singleton

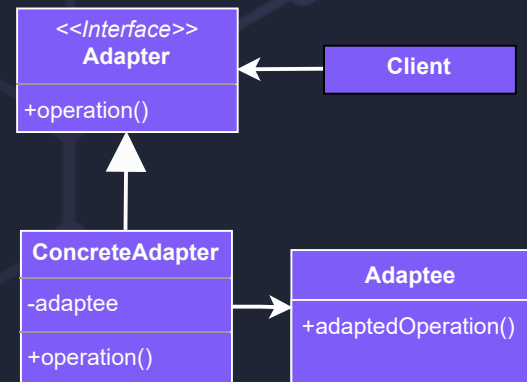
Use when you want to have one instances of a class.



Example: logging, db connections.

Adapter

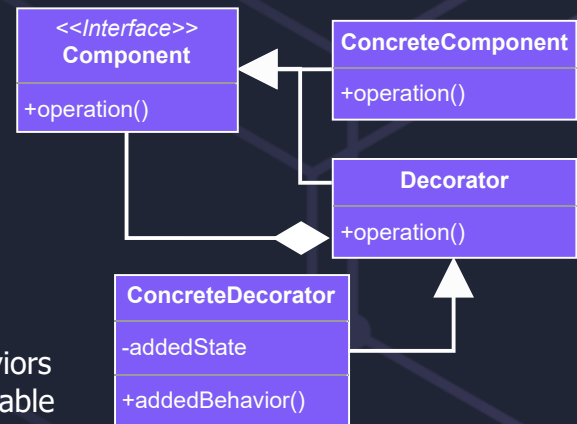
Use when you need to convert an interface to another interface



Example: make incompatible classes work together

Decorator

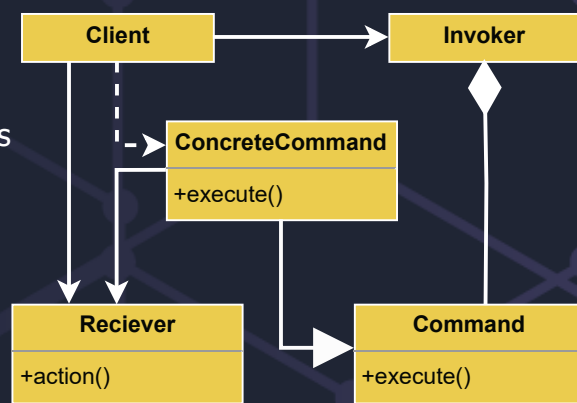
Use when you need to wrap objects to modify their behaviors



Example: make object behaviors dynamically modifiable

Command

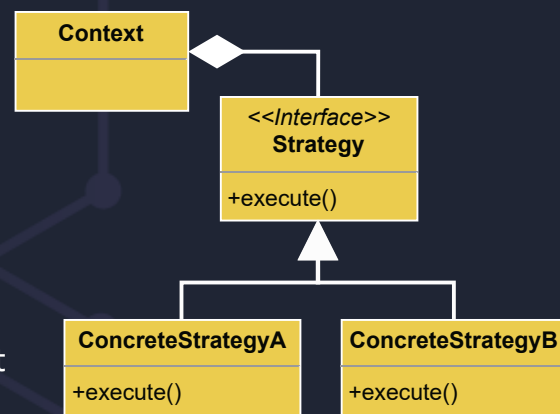
Use for encapsulating requests with parameters



Example: Implementing operations

Strategy

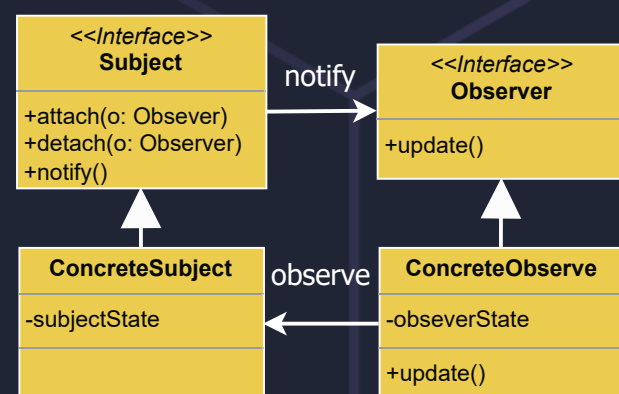
Use for interchangeable algorithms that can be swaped at runtime



Example: Implement different sorting algorithms

Observer

Use for automatic updates
of dependand objects



Example: Implement subscribers