

BEHAVIORAL

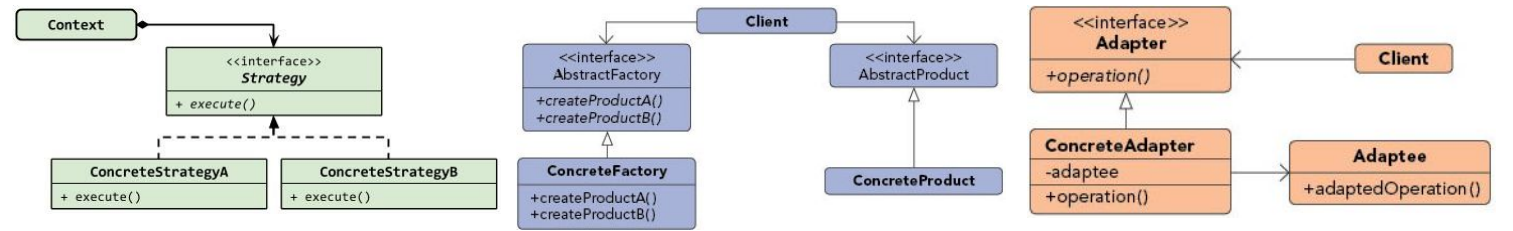
CREATIONAL

STRUCTURAL

STRATEGY

ABSTRACT FACTORY

ADAPTER



"Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from clients that use it."

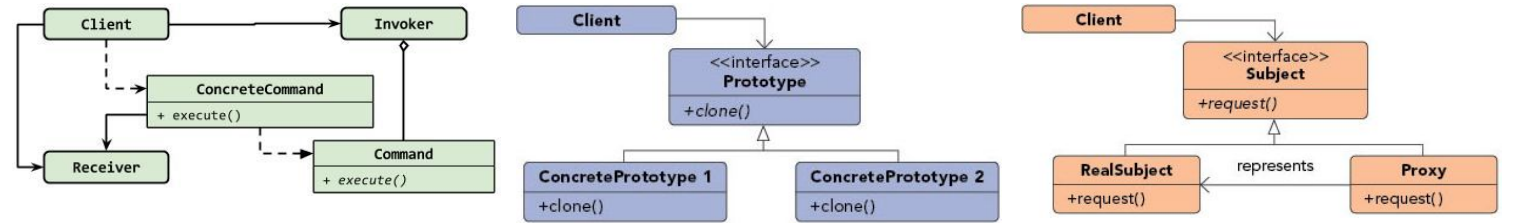
"Provide an interface for creating families of related or dependent objects without specifying their concrete classes."

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

COMMAND

PROTOTYPE

PROXY



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

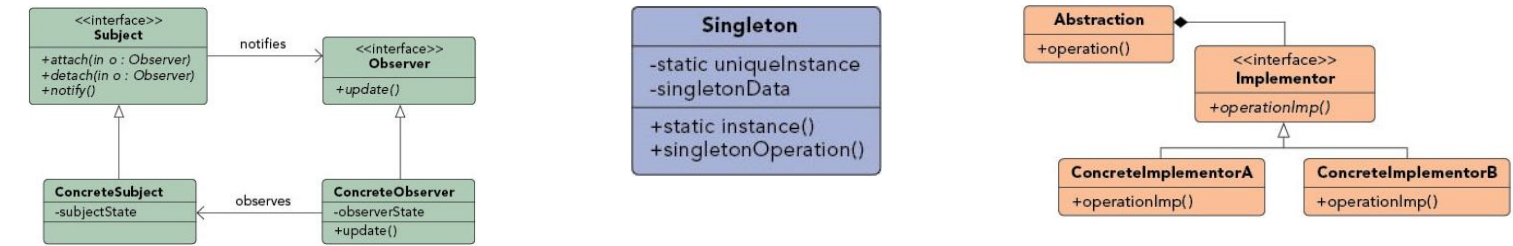
"Create objects based upon a template of an existing object through cloning."

"Provide a surrogate or placeholder for another object to control access to it."

OBSERVER

SINGLETON

BRIDGE



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

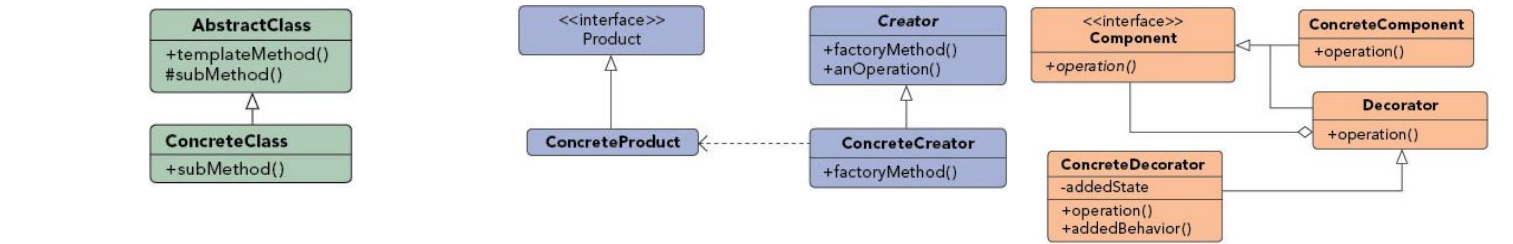
"Ensure a class only has one instance, and provide a global point of access to it."

"Decouple an abstraction from its implementation so that the two can vary independently."

TEMPLATE METHOD

FACTORY METHOD

DECORATOR



"Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure."

"Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses."

"Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality."

OBJECT POOL MANAGER

FACADE

"Uses a set of initialized objects kept ready to use - a pool - rather than allocating and destroying them on demand."

"Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use."