Creational Patterns

Abstract Factory

Definition: interface to create families of related or dependent objects without specify their concrete classes
Aspects that Vary: Families of product objects
Participants: AbstractFactory, ConcreteFactory, AbstractProduct, ConcreteProduct, Client

Builder

Definition: separate the construction of a complex object from its representation
Aspects that Vary: How a composite object gets created
Participants: Builder, ConcreteBuilder, Director, Product

Factory Method

Definition: define an interface for creating an object and let subclasses decide which class to instantiate
Aspects that Vary: Subclass of object that is instantiated
Participants: Creator, ConcreteCreator, Product, ConcreteProduct

Prototype

Definition: specify the kinds of objects to create using a prototypical instance
Aspects that Vary: Class of object that is instantiated
Participants: Prototype, ConcretePrototype, Client

Singleton

Definition: ensure a class only has one instance
Aspects that Vary: The single instance of a class
Participants: Singleton

Structural Patterns

Adapter

Definition: convert the interface of a class into another interface clients expect
Aspects that Vary: Interface to an object
Participants: Target, Adapter, Adaptee, Client

Bridge

Definition: decouple an abstraction from its implementation so they can vary independently
Aspects that Vary: Implementation of an object
Participants: Abstraction, RefinedAbstraction, Implementor, ConcreteImplementor

Composite

Definition: lets clients treat individual objects and compositions of objects uniformly
Aspects that Vary: Structure and composition of an object
Participants: Component, Leaf, Composite, Client

Decorator

Definition: attach additional responsibilities to an object dynamically
Aspects that Vary: Responsibilities of an object without subclassing
Participants: Component, ConcreteComponent, Decorator, ConcreteDecorator

Facade

Definition: provide a unified interface to a set of interfaces in a subsystem
Aspects that Vary: Interface to a subsystem
Participants: Facade, Subsystem classes

Flyweight

Definition: use sharing to support large numbers of objects efficiently
Aspects that Vary: Storage costs of objects
Participants: Flyweight, ConcreteFlyweight, NonSharedConcreteFlyweight, FlyweightFactory, Client

Proxy

Definition: an object functioning as an interface to something else
Aspects that Vary: How an object is accessed
Participants: Subject, RealSubject, Proxy

Behavioral Patterns

Chain of Responsibility

Definition: command pass through processing objects until one handle it
Aspects that Vary: Object that can process a request
Participants: Handler, ConcreteHandler, Client

Command

Definition: encapsulate a request as an object to perform an action or trigger an event at a later time
Aspects that Vary: When and how a request is processed
Participants: Command, ConcreteCommand, Client, Invoker, Receiver

Interpreter

Definition: define a representation for the grammar of a given language along with an interpreter that uses the representation to interpret sentences in the language
Aspects that Vary: Grammar and interpretation of a language
Participants: AbstractExpression, TerminalExpression, NonterminalExpression, Context, Client

Iterator

Definition: provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation
Aspects that Vary: How the elements of an aggregate are accessed
Participants: Iterator, ConcreteIterator, Aggregate, ConcreteAggregate

Mediator

Definition: define an object that encapsulates how a set of objects interact
Aspects that Vary: How and which objects interact with each other
Participants: Mediator, ConcreteMediator, Colleague, ConcreteColleague

Memento

Definition: expose the private internal state of an object so it can be restored later
Aspects that Vary: What private information is stored outside an object
Participants: Memento, Originator, Caretaker

Observer

Definition: the subject maintains a list of its observers and notifies them automatically when something changes by calling one of their methods
Aspects that Vary: How the dependent objects stay up to date
Participants: Subject, Observer, ConcreteSubject, ConcreteObserver

State

Definition: allow an object to alter its behavior when its internal state changes
Aspects that Vary: States of an object
Participants: Context, State, ConcreteState

Strategy

Definition: allow selecting an algorithm at runtime and which family of algorithms to use
Aspects that Vary: An algorithm
Participants: Strategy, ConcreteStrategy, Context

Template Method

Definition: method in an abstract class that defines the skeleton of an operation
Aspects that Vary: Steps of an algorithm
Participants: AbstractClass, ConcreteClass

Visitor

Definition: represent an operation to be performed on the elements of an object structure
Aspects that Vary: Operations that can be applied to objects without changing their classes
Participants: Visitor, ConcreteVisitor, Element, ConcreteElement, ObjectStructure