Abstract Factory

- Adapter
- Bridge
- Chain of Responsibility
- Command
- Composite
- Decorator
- Facade
- Factory Method
- Flyweight
- Interpreter
- Iterator
- Mediator
- Memento
- Prototype
- Proxy
- Observer
- Singleton
- State
- Strategy
- Template Method
- Visitor

ConcreteAggregate

ConcreteColleague

ConcreteIterator

ConcreteObserver

ConcreteStrategyA

ConcreteStrategyB

ConcreteSubject

ConcreteVisitor

Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading, Massachusetts: Addison Wesley Longman, Inc.
**Facade**
Type: Structural
What it is: Provide a unified interface to a set of interfaces in a subsystem. Defines a high-level interface that makes the subsystem easier to use.

**Adapter**
Type: Structural
What it is: Convert the interface of a class into another interface clients expect. Lets classes work together that couldn't otherwise because of incompatible interfaces.

**Bridge**
Type: Structural
What it is: Decouple an abstraction from its implementation so that the two can vary independently.

**Abstract Factory**
Type: Creational
What it is: Provides an interface for creating families of related or dependent objects without specifying their concrete class.

**Builder**
Type: Creational
What it is: Separate the construction of a complex object from its representing so that the same construction process can create different representations.

**Factory Method**
Type: Creational
What it is: Define an interface for creating an object, but let subclasses decide which class to instantiate. Lets a class defer instantiation to subclasses.

**Prototype**
Type: Creational
What it is: Specify the kinds of objects to create using a prototypical instance, and create new objects by copying this prototype.

**Singleton**
Type: Creational
What it is: Ensure a class only has one instance and provide a global point of access to it.