

Design Patterns

By Miles Coney

Creational Patterns

Singleton

Builder

Factory



Singleton

- Design pattern that ensures a given class can only have one object.
- Constructor is made private to restrict access of objects
- Example: Often used in logging, config settings and caches



Builder

- A way to construct complex objects on a step by step basis.
- Code used to construct an object is removed from its class and instead placed in a separate builder object.
- Example: Utilized when creation of objects are complex. Ex. Creation of a license or two video game characters with similar and differing attributes



Factory

- Class creation that is designed to function as a creator for objects that will be utilized in other classes
- Examples include creation of People, Automobiles, Animals



Structural Patterns

Decorator

Adapter

Proxy

Facade



Decorator

- Provides additional responsibilities to an object, modifiable at runtime
- The object is wrapped in a decorator to add attributes
- Example: Netflix app having a trial user option



Adapter

- Used to bridge together incompatible interfaces
- Adapter wraps a class with a new interface so that it becomes compatible with user's interface
- Example: An app that needs to convert mph to kpmh for a user using app n US and UK.



Proxy

- Provides a placeholder for another object in order to control access to it
- “Thing calls the thing you want rather than just calling the thing”
- Example: Remote proxy, virtual proxy, protection proxy



Facade

- Complexities of the subsystem are hidden and interaction is done via a Facade class or interface is interacted with instead which invokes all the other complex pieces(classes).
- Example: Utilizing an application or computer that has a bunch of different subsystems required to make it work, but interaction is done through a simple interface to operate



Behavioral Patterns

Observer


Strategy

Template

Command



Observer

- One to many dependency between objects.
 - When the one objects state has changed, the other objects are notified and updated accordingly
 - Example: Email subscriptions where you register and are then sent notified on subsequent changes
 - An application that notifies you of updates
- 

Strategy

- Allows for both behavior and algorithm to be changed at runtime
- Utilization of multiple objects
- Example: Strategizing placing multiple buildings in a city area



Template

- Defines the skeleton of the algorithm
- Based on inheritance
- Example: Mass housing construction
 - Foundation
 - Framing
 - Plumbing Walls



Command

- Allows for both behavior and algorithm to be changed at runtime
- Utilization of multiple objects
- Example: Strategizing placing multiple buildings in a city area

