

Resumen de Clase

Patrones

Son soluciones a problemas tipicas que ya estan probados y son utilizados en la industria.

SOLID

SRP

Una clase solo debe tener una razon para cambiar.

OCP

Abierta a la extension, cerrado a la modificación

Proteger el estado de una variable (no hacerla global)

Anti-Patrones

Son basura que se deben evitar para poder tener codigo de calidad.

Blob

Un codigo gigante que es dificil de mantener y entender, con poca documentacion

Lava Flow

Codigo que se mantiene si es necesario.

Codigo que no se toca por miedo a romper algo.

Soluciones: —> refactor de codigo.

Golden Hammer

Cuando se depende demasiado de una tecnología.

Sobreutilización de una herramienta

Se evita no generando sesgo para una tecnología en especial

Spaghetti Code

Código ilegible y complejo, que después da miedo tocarlo.

Métodos con gran scroll son típicamente este tipo de AP

Copy and Paste

Utilizar código que no entiendo, pero que funciona, extraído de quien sabe donde.

Tester Driven Development

Requerimientos del área de Test que rompe todo

Patrones creacionales

Design Patterns (Gang of Four)

3 tipos de patrones

- Creacionales
- Estructurales
- De comportamiento

Patrón creacional:

Se abordan diferentes formas de la creación de objetos, de estos se desprenden varias formas.

Singleton

Se garantiza una sola instancia de la clase, de modo de hacerla global.

Contras:

- Viola SRP, genera instancias
- Puede enmascarar un mal diseño.

Factory Method

Basicamente se utiliza para encapsular la creación de objetos.

Transfiere la responsabilidad de creación de objetos.

Builder

Permite construir el objeto de forma compleja paso a paso.

Intenta desacoplar el código del cliente.

Prototype

Similar al Builder, se usa para clonar objetos

2 tipos: superficial y profunda.

Patrones estructurales

Facilitan el diseño al identificar relaciones entre entidades.

Explican como ensamblar objetos pequeños en otros más grandes.

Adapter

Utilización de Interfaces para poder comprender un objeto a otro.

Transforma el contenido de una entidad para pasarselo a otra entidad, todo a través de una Interfaz

Birdge

permite dividir una clase o grupo de clases relacionadas, separando la logica existente en cada clase. Ej. Forma (cuadrado, circulo) —> Color (Rojo, azul)

La Abstracción (o Interfaz) de una capa...

Decorator

Permite encapsular el objeto inicial en una interfaz abstracta

Ej. Window, scroll h, scroll v, borde.

Facade

Proporciona una interfaz unificada a un conjunto de interfaces en un subsistema.

Proxy

Cambia la posición de un marcador para proteger el contenido original de él.

Crea una nueva clase de proxy, que permite hacer cosas antes o después con los objetos.

- Inicialización diferida
- Control de acceso
- Solicitudes de registro
- Referencia Inteligente

DAO (Data Access Object)

Abstracción de la persistencia de datos y se considera más cercano al repo de almacenamiento

Repository

Mecanismo para encapsular el comportamiento de almacenamiento, recuperación y búsqueda , que emula una colección de objetos.

Chain of Responsibility

Genera una abstracción con un pipeline que se conectan entre si. Cada elemento tiene una referencia a su padre y a su siguiente. Cada handler se fija si le toca realizar el trabajo o no.

Command

encapsula una solicitud de acción, para que se lleve a cabo en un objeto con los parametro de la acción.

Tiene un metodo Execute y una clase Invoker

Observer

se conoce como Publisher-Subscriber, una dependencia uno a muchos entre objetos.

Cuando un objeto cambia, el objeto dependiente se entera del cambio ocurrido.

Actores:

Sujeto(Subject)

Observadores (Observer)

Mediator

Comunicacion N a N, pero en un solo punto-

las clases se comunican con una unica clase que se comunica con el resto en su nombre.

Memento

Para tener una copia del estado de un objeto.

Partes

- Originator
- Caretaker
- Snapshot

State

Cambiar el comportamiento cuando el estado cambia.

Strategy

Va a ejecutar una estrategia en particular acorde el contexto

Ejemplo: empresas de paqueteria, segun el peso del producto (distintos % de precios por Kg)

Partes:

- Strategy
- Concrete Strategies
- Context

Visitor

Permite separar algoritmos de los objetos sobre los que operan. Se utiliza para operar sin cambiar estado del objeto operado.

Partes:

- Visitor
- Concrete visitor
- Element
- Concrete element