
# SDET Course

Design Patterns -Facade

# 3 Types of Design Patterns

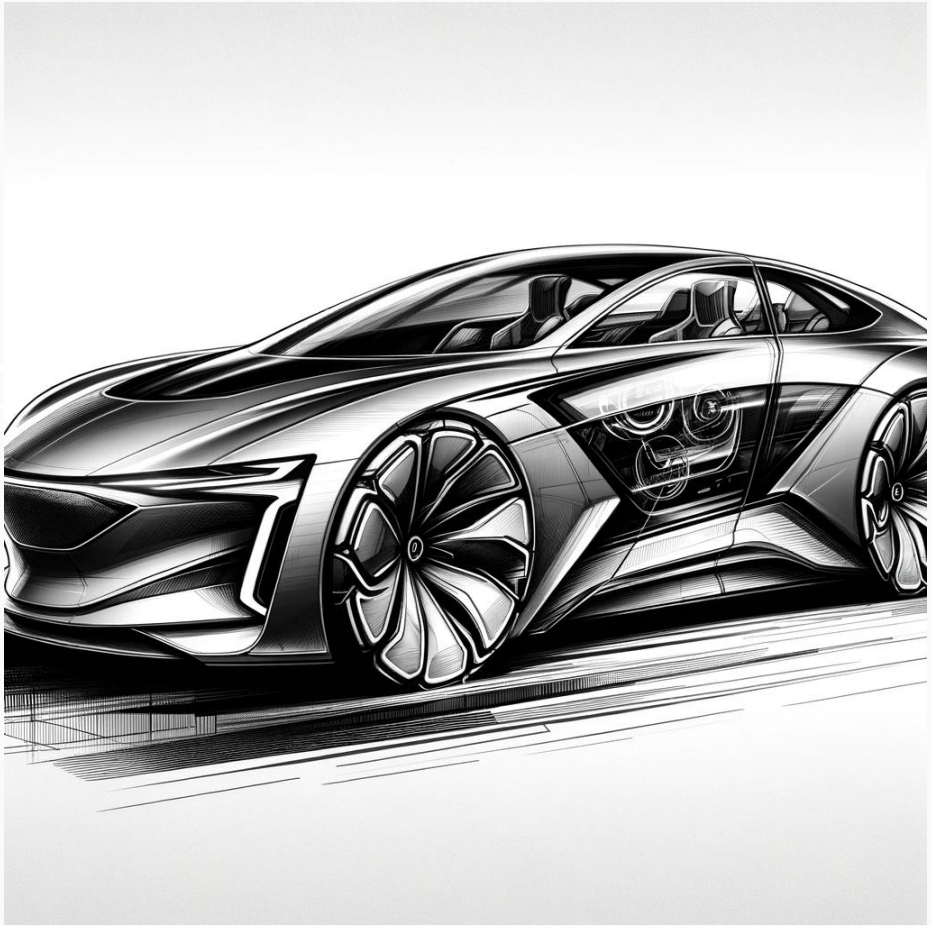
- Creational
  - Singleton
  - Builder
  - Prototype
  - Factory Method
  - Abstract Factory
- Structural
  - Adapter
  - Composite
  - Proxy
  - Flyweight
  - Bridge
  - **Facade**
  - Decorator
- Behavioral
  - Strategy
  - Observer
  - Command
  - Memento
  - State
  - Template Method
  - Mediator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

# Agenda

- Description
- Diagram
- Code sample (Java)
- Use cases

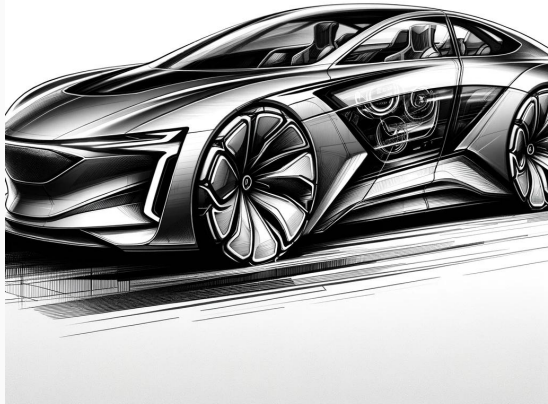
VERISOFT

Descrip
on

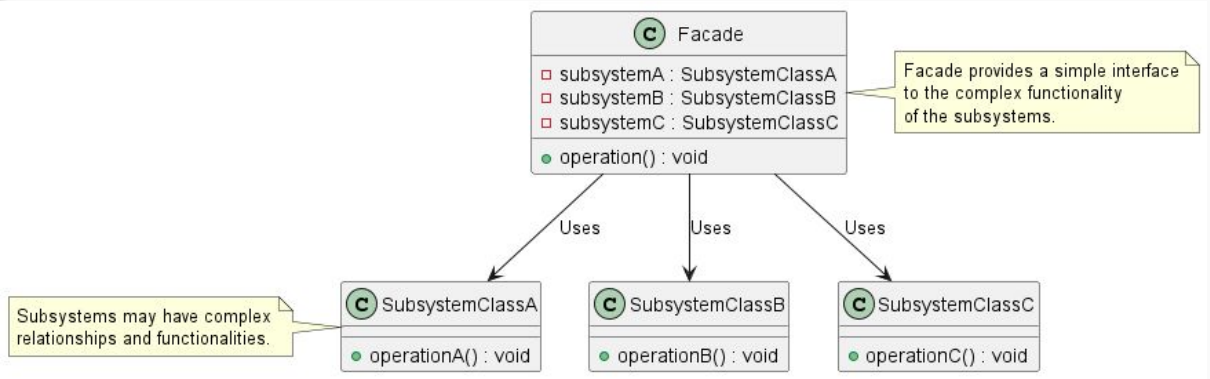


# Description

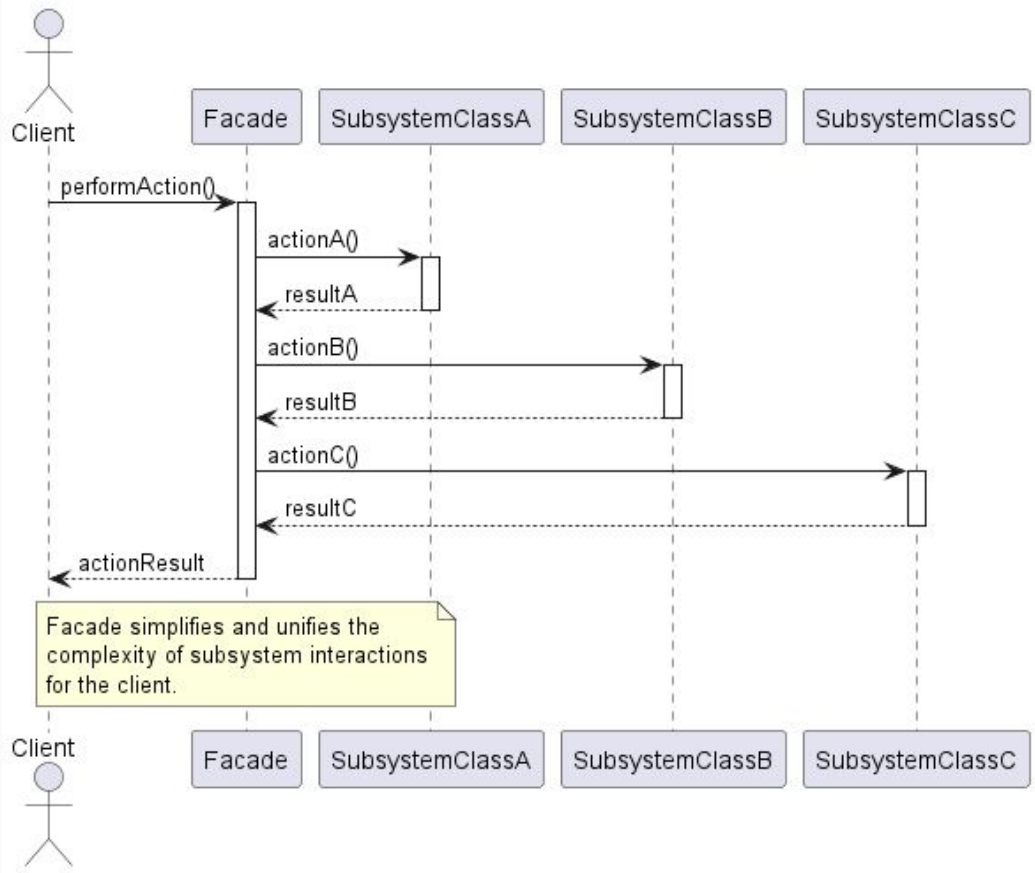
The Facade Design Pattern is a structural design pattern that provides a simplified interface to a complex system of classes, library, or framework. By doing so, it hides the complexities of the system and makes it easier for the client to interact with it. This pattern involves creating a facade class that serves as a single point of access to different parts of a system. The facade class forwards requests from the client to the appropriate components of the system without exposing the underlying details. This reduces dependencies of outside code on the inner workings of the system, leading to better decoupling and easier maintenance. The Facade Pattern is particularly useful when dealing with complex systems with multiple interdependent classes or when there are multiple entry points to a subsystem that needs to be simplified.



# Class Diagram



# Sequence Diagram



# Code Sample

# Use cases

VERISOFT

- General
  - Wrapping API
  - Logging interaction (Slf4J)
  - Legacy Systems

- In Test Automation
  - Search Page Object (Google / Yahoo)
  - Different resolutions - responsive web sites


# Happy Coding