

# Facade

## Objective

Provide a unified interface for a set of interfaces of a subsystem, facilitating the use of that subsystem.

## Function

Facilitate the addition of functionality to a class dynamically.

## Structure

Facade provides convenient access to a particular part of the functionality of the subsystem. You know where to address the customer's request and how to operate all moving parts.

The structure that meets this pattern is shown in Figure 1

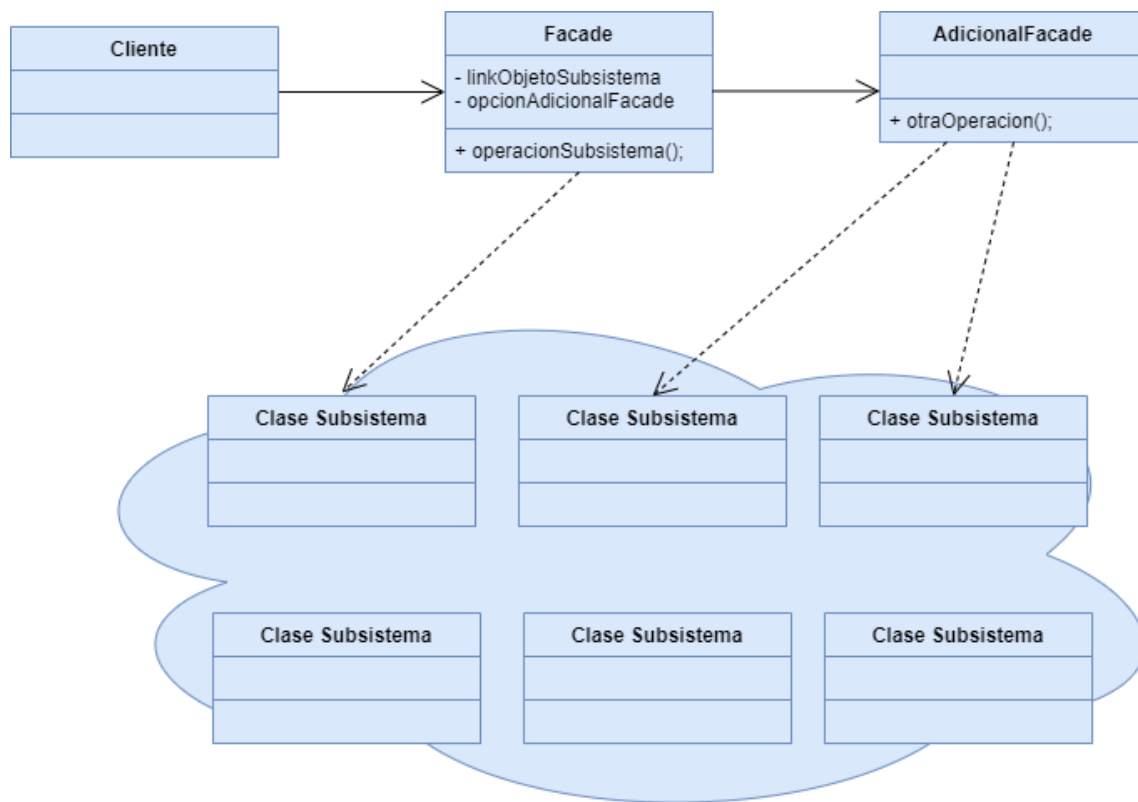


Figure 1: UML Diagram Facade Pattern

# Applications

The use of the Facade pattern is recommended when:

- There are too many dependencies between clients and the implementation of classes, facilitating portability.
- The aim is to define an entry point for each subsystem, to simplify the interaction between subsystems through their respective implementations of the Facade pattern.
- Avoid the increasing complexity of the subsystems due to their evolution process, by implementing a simple interface.

## Design Patterns Collaborators

- The Facade and Abstract Factory patterns can work together to provide an interface for creating subsystem objects independently.
- The Facade and Mediator patterns are similar in that they abstract the of existing classes, considering that the purpose of Mediator is to summarize the communication between fellow objects.

## Scope of action

Applied at the object level.

## Problem

The implementation of a solution that simplifies the use of a complex subsystem is carried out by means of an encapsulation process, which in many occasions turns out not to be the best way.

## Solution

The Facade pattern introduces an object that provides a single interface simplified by minimizing communication and dependencies between subsystems. The object performs a complex task since it must translate the interface of each subsystem to a common language to allow interaction between them.

## Diagram or Implementation

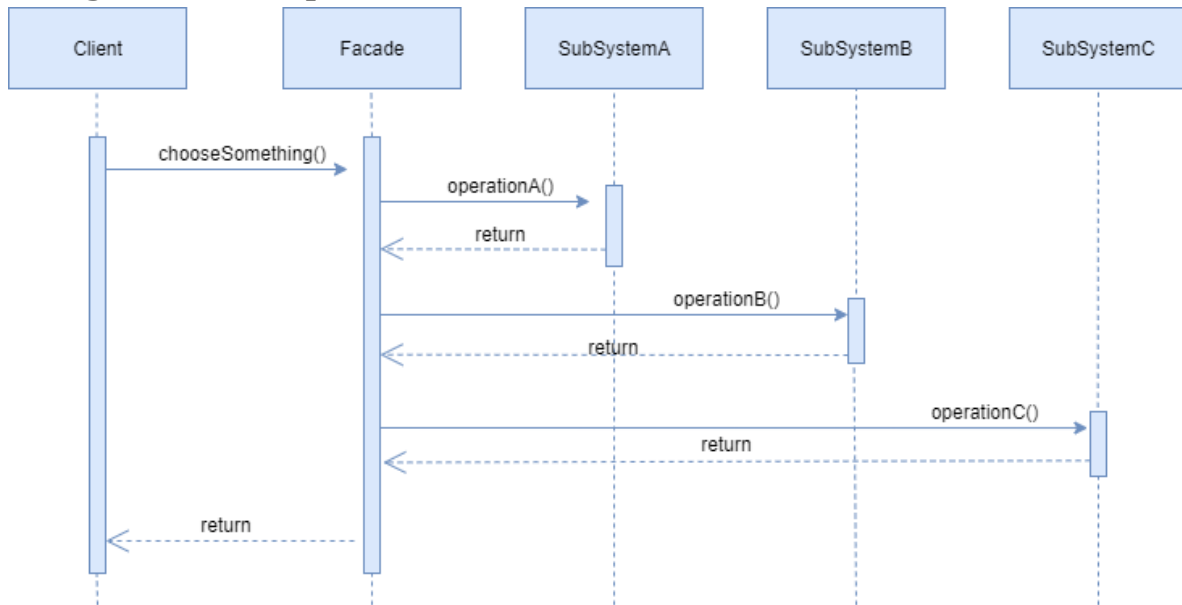


Figure 2: UML Diagram Facade Pattern

Figure 2 explains the behaviour of the Facade pattern using a sequence diagram.

- The client class invokes a Facade class transaction.
- The Facade class communicates with the SubsystemA component to perform a operation.
- The Facade class communicates with the SubsystemB component to perform a operation.
- The Facade class communicates with the SubsystemC component to perform a operation.
- The Facade class responds to the client class with the result of the operation.