Façade ID

Student:

Izabela Kuźniar

Teacher:

Andrea Corradini

Course:

Software Design Patterns

# Name and category

Façade is a structural pattern.

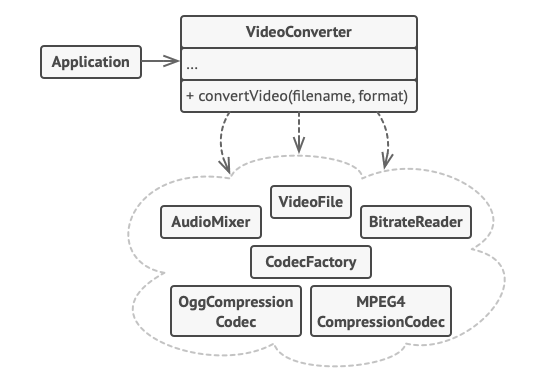
# Intent:

Façade provides a simplified interface to a complex set of classes.

# Motivation:

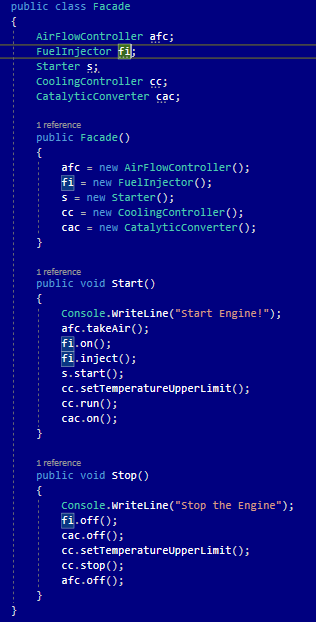
Façade is one of the few patterns that instead of decomposing the system into small pieces, combines it to provide a unified interface.

# Structure as a UML class diagram



# Implementation:

To create a Façade is to unify all the objects of other classes in one.



# Consequences:

Benefits:

* Isolation of the code from a complexity of the subsystem

Drawbacks:

* It can end up being a coupled object to all classes of an application.

# Known uses

* Class Class<T> is a Façade
* Class URL is a Façade

# Related patterns

1. Adapter: makes the existing interface usable, while Façade defines a new interface,
2. Abstract Factory: can serve as an alternative to Façade when there is a need to hide the way the subsystem objects are created from the client.
3. Flyweight shows how to make lots of object, whereas Façade shows how to make a single object that represents an entire subsystem.
4. Mediator and Façade have similar jobs: they try to organize collaboration between lots of tightly coupled classes.
   1. Mediator centralizes communication between components of the system. The components only know about the mediator object and don’t communicate directly.
   2. Façade defines a simplified interface to a subsystem of objects, but it does not introduce any new functionality. The subsystem itself is unaware of the façade. Objects within the subsystem can communicate directly.
5. Singleton: Façade can be transformed into Singleton.
6. Proxy: Façade is similar to this pattern.