Façade design pattern

Facade is a structural design pattern that provides a simplified interface to a library, a framework, or any other complex set of classes. The facade pattern conceals the system's complexity and offers the client an interface via which they can access the system. Given that it adds an interface to an existing system to conceal its complexity, this form of design pattern falls under the structural pattern category.

This pattern makes use of a single class that offers clients the streamlined methods they need while delegating calls to methods of pre-existing system classes.

Implementation

We are going to create a Shape interface and concrete classes implementing the Shape interface. A facade class ShapeMaker is defined as a next step. ShapeMaker class uses the concrete classes to delegate user calls to these classes. FacadePatternDemo, the demo class, will use ShapeMaker class to show the results.

As the name suggests, it means the face of the building. The people walking past the road can only see the glass face of the building. They do not know anything about it, the wiring, the pipes, and other complexities. It hides all the complexities of the building and displays a friendly face.

Advantage of Facade Pattern

- It shields the clients from the complexities of the sub-system components.
- It promotes loose coupling between subsystems and its clients.

Usage of Facade Pattern:

It is used:

- When you want to provide simple interface to a complex sub-system.
- When several dependencies exist between clients and the implementation classes of an abstraction.