

# Façade Pattern

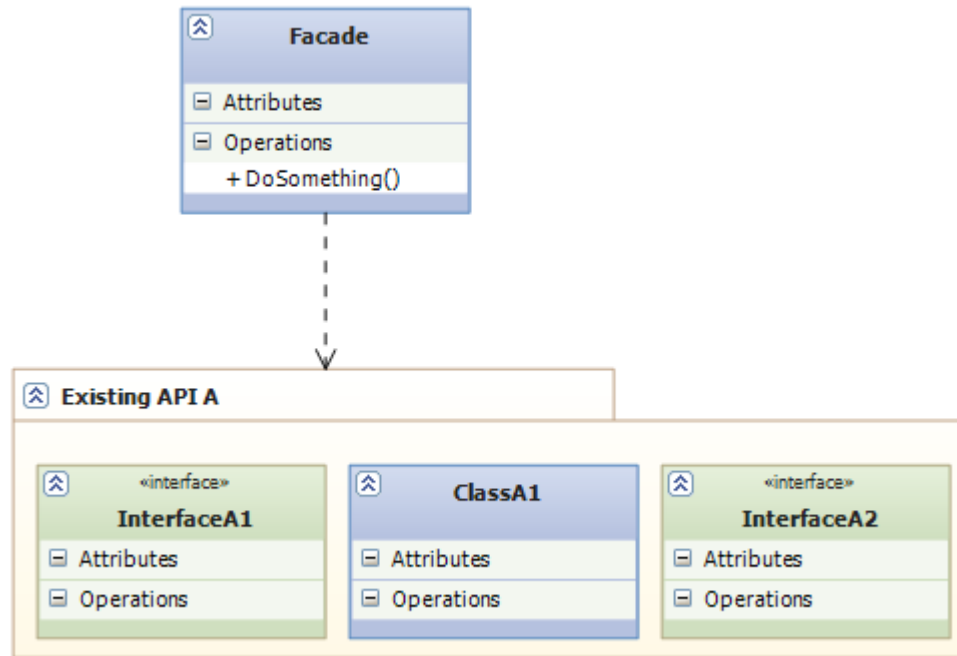
Design Patterns



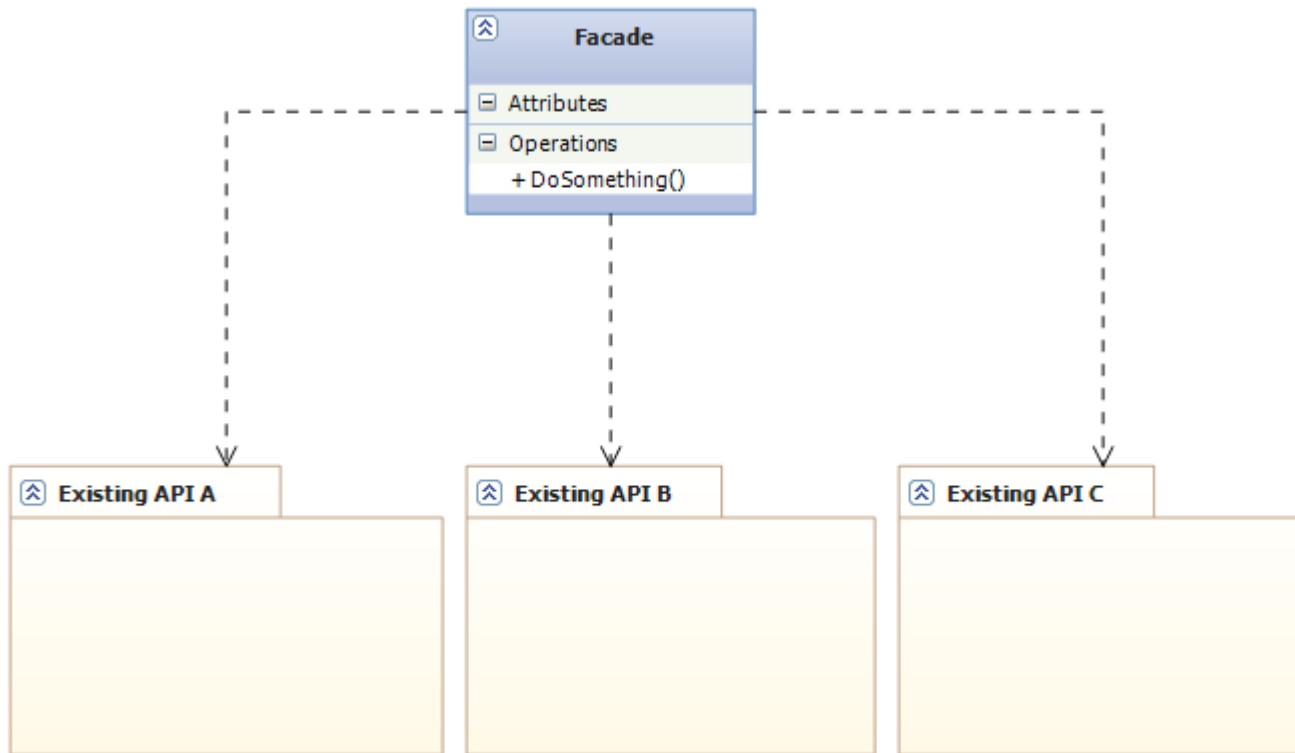
# Intent

- **Provide a simple, purpose-built interface to a larger, more complex body of code or interfaces**
- **Make a complex body of code simpler to consume and use**
- **Expose a set of complex object interactions with a single interface**
- **Wrap a poorly designed API in a better designed one**

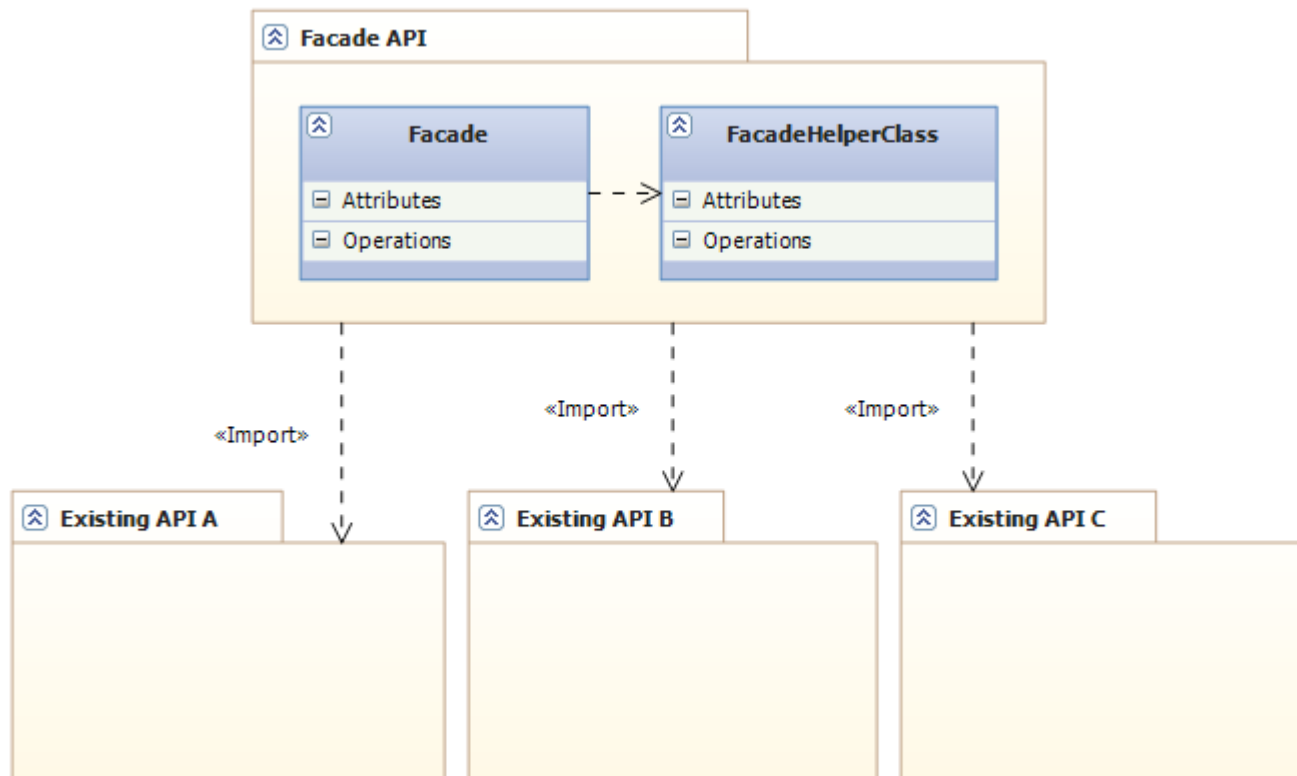
# Structure



# Structure



# Structure



# Consequences

- **The use of the existing API is simplified**
- **The façade will need to be updated (or a new one created) to expose more functionality from the sub-system**
- **The full capability of the underlying APIs will often not be available through the façade**

# Known Uses

## Common Façade Implementations

- Data Access
- File and Stream IO

## Demo

- Pluralsight.com search with Lucene.NET

# Related Patterns

- **Adapter**
- **Flyweight**
- **Mediator**



# Summary

- **Use the Façade pattern to**
  - Wrap complex or multiple APIs
  - Cover up poorly written code to make it easier to consume  
(Shove your dirty laundry under your bed)
- **Many developers do this without knowing it had a name**