Design Patterns - Hallmarks of Good Architecture

SOLID Principles of object-oriented programming.

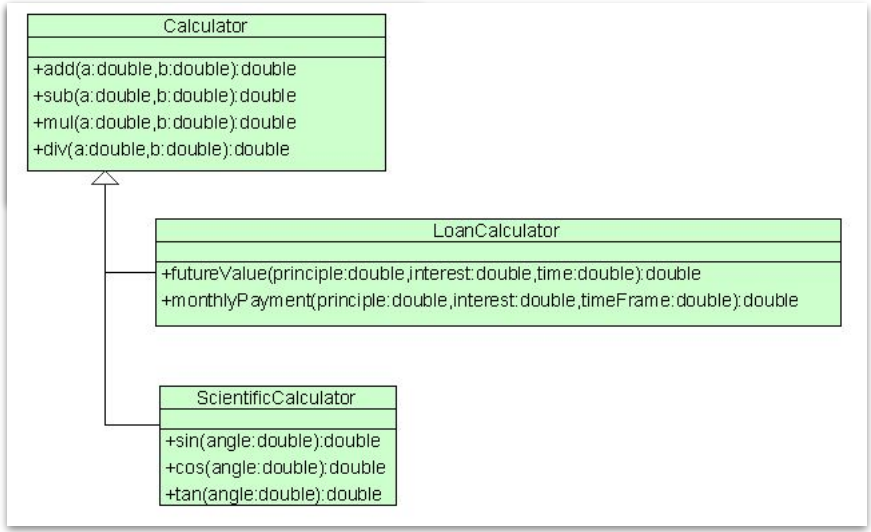
1. **The single-responsibility principle:** *"There should never be more than one reason for a class to change."* Each class should have only one central responsibility.
2. **The open–closed principle:** *"Software entities ... should be open for extension, but closed for modification."*
3. **The Liskov substitution principle:** *"Functions that use pointers, or references to base classes, must be able to use objects of derived classes without knowing it."*
4. **The interface segregation principle:** *"Clients should not be forced to depend upon interfaces that they do not use."*
5. **The dependency inversion principle:** *"Depend upon abstractions, [not] concretions."*

Design Patterns - Hallmarks of Good Architecture

1. **The single-responsibility principle:** *"There should never be more than one reason for a class to change."* Each class should have only one central responsibility.
    a. Persistence
    b. Pre/Post Conditions - Validation
    c. Notification
    d. Logging
    e. Formatting
    f. Parsing (JSON, XML, CSV, …)
    g. Error Handling

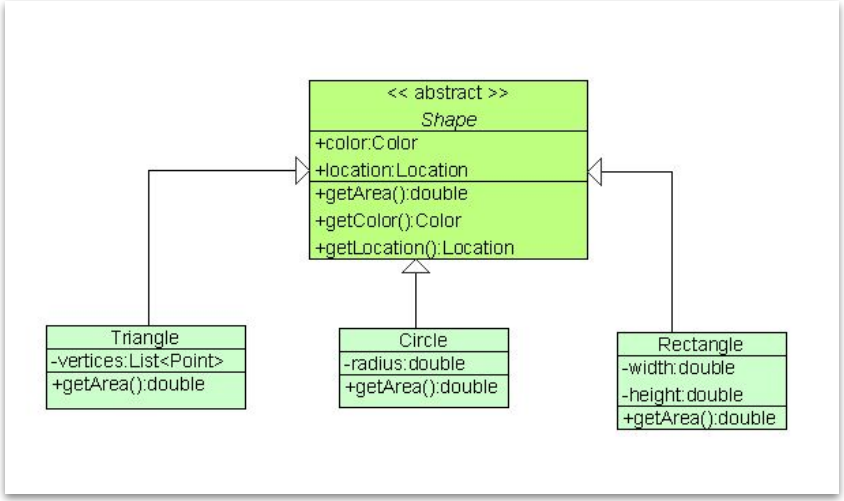
Design Patterns - Hallmarks of Good Architecture

2. **The open–closed principle:** *"Software entities ... should be open for extension, but closed for modification."*



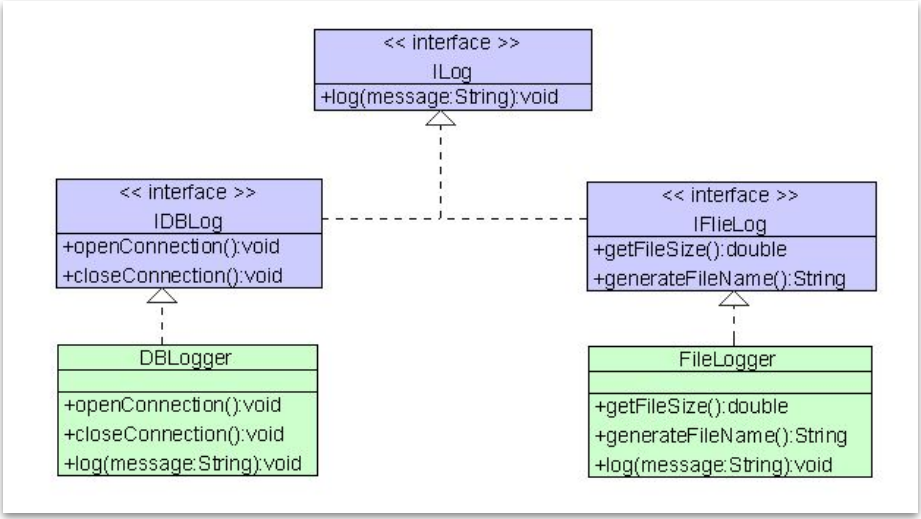
Design Patterns - Hallmarks of Good Architecture

3. **The Liskov substitution principle:** *"Functions that use pointers, or references to base classes, must be able to use objects of derived classes without knowing it."*
    a. If class A is a subtype of class B, then we should be able to replace B with A without interrupting the behavior of the program.



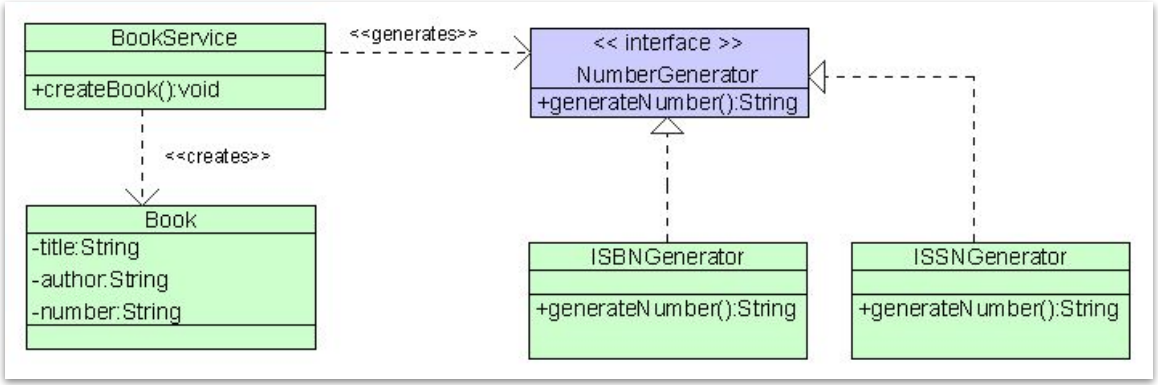
Design Patterns - Hallmarks of Good Architecture

4. **The interface segregation principle:** *"Clients should not be forced to depend upon interfaces that they do not use."*
    a. Declaring methods in an interface that the caller doesn't need pollutes the interface and leads to a “bulky” or “fat” interface.



Design Patterns - Hallmarks of Good Architecture

5. **The dependency inversion principle:** *"Depend upon abstractions, [not] concretions."*



Design Patterns - Hallmarks of Good Architecture

To summarize, we will do the following four things in this course:

1. We will learn about the most fundamental design patterns in the industry,
2. We will study them in a proper architectural context.
3. We will see how they cover the **SOLID** and good architecture principles.
4. We will use them practically in a project.