Design Patterns-Elements of Reusable Object Orientated Software. Authors: Gang of Four

The design patterns that they describe are solutions to general problems that developers face during software development.

Design Patterns (DP) can be thought of as a general reusable solution or template to a commonly occurring problem.

Patterns typically show interactions and relationships between multiple classes

Benefits

1. Speeds up development
    - First understand problem
    - Then map it to the appropriate DP
2. Reuse of old interfaces
3. Makes current code more reliable

Design patterns in the book

I. Creational (5/23)

1. Abstract Factory
2. Builder
3. Factory Method

4. Prototype
5. Singleton

II. Structural (7/23)

1. Adapter
2. Bridge
3. Composite
4. Decorator
5. Facade
6. Flyweight
7. Proxy

III. Behavioral (11/23)

1. Chain of responsibility
2. Command
3. Interpreter
4. Iterator
5. Mediator
6. Momento
7. Observer
8. State
9. Strategy
10. Template Method
11. Visitor

Creational Patterns

- Provide means to create objects while hiding creation logic

- Do not instantiate objects directly using “new” operator
- Can be categorized into two categories
    - Class Creation (using inheritance)
    - Object Creation (using delegation)

Structural Patterns

- Concerned with organizing different classes and objects to form large structures and new functionality
- And inheritance used to compose interfaces

Behavioral Patterns

- Concerned with identifying and realizing common communication patterns between objects