

Programming Patterns - Mastery Notes

Creational Patterns (5)

1. Singleton - One instance class. Use in logging/configs.
2. Factory Method - Subclasses decide which class to instantiate.
3. Abstract Factory - Create related objects without specifying classes.
4. Builder - Build complex objects step-by-step.
5. Prototype - Clone objects with/without deep copy.

Structural Patterns (7)

6. Adapter - Converts interface of one class into another.
7. Bridge - Decouple abstraction from implementation.
8. Composite - Tree structures for part-whole hierarchies.
9. Decorator - Add behavior dynamically.
10. Facade - Simplified interface to complex subsystems.
11. Flyweight - Efficient sharing of common data.
12. Proxy - Substitute or placeholder for another object.

Behavioral Patterns (11)

13. Chain of Responsibility - Pass request along chain.
14. Command - Encapsulate request as object.
15. Interpreter - Grammar and parsing logic.
16. Iterator - Sequential access without exposing structure.
17. Mediator - Central communication hub.
18. Memento - Save/restore object state.
19. Observer - Notify dependents on state change.
20. State - Behavior changes with state.

- 21. Strategy - Select algorithm at runtime.
- 22. Template Method - Algorithm skeleton with hooks.
- 23. Visitor - Add operations without modifying classes.

Concurrency Patterns (3)

- 24. Thread Pool - Manage a pool of worker threads.
- 25. Producer-Consumer - Thread-safe communication.
- 26. Read-Write Lock - Optimize concurrent access.

Architectural Patterns (3)

- 27. MVC - Model, View, Controller separation.
- 28. MVVM - Model-View-ViewModel with binding.

8-Week Study Plan

Week 1: Singleton, Factory, Builder

Week 2: Adapter, Bridge, Composite

Week 3: Decorator, Facade, Proxy

Week 4: Command, Observer, Strategy

Week 5: State, Visitor, Mediator

Week 6: Chain, Memento, Thread Pool

Week 7: MVC, MVVM

Week 8: Practice with projects and revise