



Design Patterns in Java syllabus

1. Introduction
 - 1.1. Why Design Patterns?
 - 1.2. Java Code Example with and without Design Patterns: Game of Live
 - 1.3. Types of Patterns: Design, Analysis, and Process Patterns.
 - 1.4. Anti-Patterns.
 - 1.5. Brief Introduction to UML Class Diagrams.
2. Object Orientation
 - 2.1. Object Oriented Design Principles
 - 2.2. "SOLID" Principles
3. Creational Patterns
 - 3.1. Factory
 - 3.2. Abstract Factory
 - 3.3. Singleton
 - 3.4. Builder
 - 3.5. Prototype
 - 3.6. Object Pool
4. Structural Patterns
 - 4.1. Adapter
 - 4.2. Bridge
 - 4.3. Decorator
 - 4.4. Façade
 - 4.5. Flyweight
 - 4.6. Proxy
 - 4.7. Dynamic Linkage
 - 4.8. Cache Management
5. Partitioning Patterns
 - 5.1. Layered Initialization
 - 5.2. Filter
 - 5.3. Composite
6. Behavioral Patterns
 - 6.1. Chain of Responsibility
 - 6.2. Command
 - 6.3. Interpreter
 - 6.4. Iterator
 - 6.5. Mediator
 - 6.6. Snapshot
 - 6.7. Memento



Design Patterns in Java syllabus

- 6.8. Observer
- 6.9. State
- 6.10. Strategy
- 6.11. Template Method
- 6.12. Visitor
- 6.13. Null Object

- 7. Structural Patterns
 - 7.1. Adapter
 - 7.2. Bridge

- 8. Concurrency Patterns
 - 8.1. Single Threaded Execution
 - 8.2. Guarded Suspension
 - 8.3. Scheduler
 - 8.4. Thread Pool
 - 8.5. Fork/Join

- 9. JEE Patterns
 - 9.1. MVC
 - 9.2. Business Delegate
 - 9.3. Composite Entity
 - 9.4. Data Access Object
 - 9.5. Front Controller
 - 9.6. Intercepting Filter
 - 9.7. Service Locator
 - 9.8. Transfer Object

- 10. Java Anti Patterns
 - 10.1. Design Anti Patterns
 - 10.2. Tactical Code Anti Patterns