

Observer Pattern

Indhold

Observer Pattern	1
Definition	1
Motivation:	1
Components:	1
Types of Observer Patterns.....	2
Pros and Cons:	2
SOLID Principles:	2
Comparison:	3
Diagrams	3

Definition:

Behavioural Design Pattern: Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

Motivation:

Used to maintain consistency across related objects without making them tightly coupled.

Components:

- **Subject (Publisher)**
 - Maintains a list of observers.
 - Provides methods to attach, detach, and notify observers.
- **Observer (Subscriber)**
 - Interface with an `Update` method.
 - Receives updates from the subject.

Types of Observer Patterns

- **Pull Model**

- The observer requests updates from the subject.

- **Push Model**

- The subject sends updates to the observer.
-

Pros and Cons:

- **Pros:**

- Decouples subjects and observers.
- Allows dynamic relationship between objects.

- **Cons:**

- Potential memory leaks if observers are not properly detached.
 - Complexity increases with a large number of observers.
-

SOLID Principles:

- **Single Responsibility Principle (SRP):**

- Each class has a single responsibility.

- **Open/Closed Principle (OCP):**

- New observers can be added without modifying the subject.

- **Liskov Substitution Principle (LSP):**

- Observers can be substituted without affecting the subject.

- **Interface Segregation Principle (ISP):**

- Observers implement only the necessary update method.

- **Dependency Inversion Principle (DIP):**

- Subjects and observers depend on abstractions.

Comparison:

Strategy Pattern:

- Encapsulates algorithms, enabling the client to choose which one to use at runtime.
- Observers react to changes in the subject, while strategies define different ways to execute an algorithm.

Diagrams



