

The background features a complex network of thin grey lines and dots, forming various triangular shapes. Some triangles are solid grey, while others are outlined. The overall aesthetic is minimalist and geometric.

PATTERN OF THE WEEK

Week 10: Observer Pattern

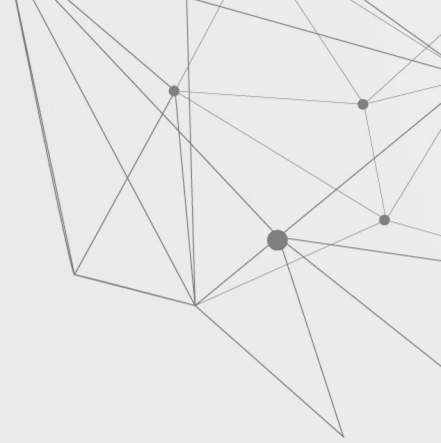
OBSERVER PATTERN
Einfach

01

OBSERVER PATTERN
Flexibel

02

TABLE OF CONTENTS



01

OBSERVER PATTERN

By the book



"THE" OBSERVER PATTERN

- Beobachterentwurfsmuster
- Senden und abonieren von Benachrichtigungen
- IObserver → Update(state) zum Empfangen
- ObserverX implementiert das Interface
- Subject → Notify(state) zum Benachrichtigen aller registrierten Observer
- Subject → Attach(), Detach() zum Hinzufügen/Entfernen von Observern
- Frage: Basisklasse?

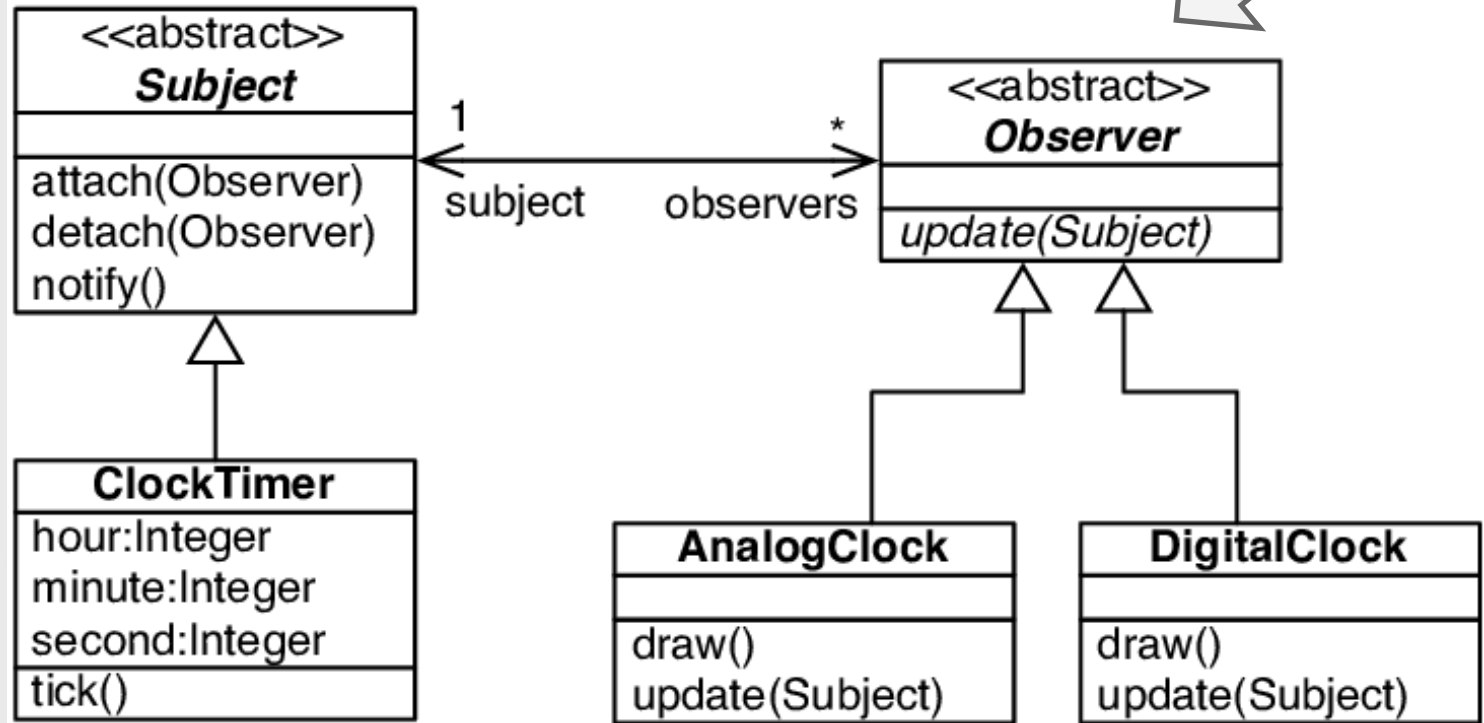


UML DIAGRAM

Attach/Detach/
Notify Logik

Interface für
den Observer

State
Logik



AUFGABE

PROJEKT PDFTOOLS: "FILE CREATED OBSERVER"

Subject mit Attach(), Detach(), Notify()
FileCreatedSubject: SetFileCreated/FileCreated{get;}

Observer mit Liste der Dateien
Observer mit Speichern der Gesamtgröße der Dateien
Observer.ctor(FileCreatedSubject)
IObserver.Update(subject) [Typecheck]

Erweitern des "ArchiveCommand"





02

OBSERVER PATTERN .NET

"THE" OBSERVER PATTERN - POTENTIAL

- Problem: Singleton + Multithreaded
- Erweiterung mit neuen Subjects "sperrig" (Konstruktur)
- Der "State" wird eine unabhängige Klasse
- Generisches Subject (Subject<T>) where T: FileCreatedMessage
- Generischer Observer (IObserver<T>) where T: FileCreatedMessage



"THE" OBSERVER PATTERN - VERBESSERUNGEN

- Observer implementiert `IObserver<Foo>` und `IObserver<Bar>`
- `IObserver<T>.Update(T message)`
- Subject → Attach/Detach auf Observer mit `List<Object>`
- `Subject.Notify(T message)` auf `List<Object>.Of<IObserver<T>>`
- State-Subject wird zur State-Klasse (DTO, POCO)



OBSERVER PATTERN .NET

IMPLEMENTIEREN DES GENERISCHEN IOBSERVER

`IObserver<T>`
`IObserver.Handle(T msg)`
State-Klasse: `FileCreated`

`Subject.Publish(T msg)`
Dependency Injection `ISubject`

`Observer WebRequestHistory: Handle<RequestExecuted>`
`Logger : IObserver<FC>, IObserver<RE>`

