Verhaltensmuster Observer

Software Architektur

Wintersemester 2019

WWI 17 SE B, Gruppe 3

Andreas Gülükoglu,
Felix Waage,
Fabio Westphal,
Yvonne Werner,
Milena Zahn



Agenda

- Definition und Erklärung
- Vor- und Nachteile
- UML-Diagramm
- Beispiel
- Quellenangaben



Definition Observer

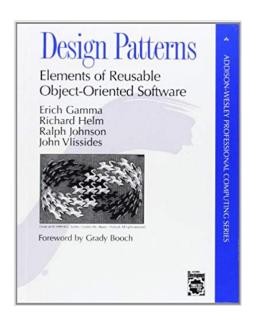


Bild quelle: https://www.amazon.de/Patterns-Elements-Reusable-Object-Oriented-Software/dp/0201633612

"Definiere eine 1-zu-n-Abhängigkeit zwischen Objekten so, dass die Änderung des Zustands eines Objekts dazu führt, das alle abhängigen Objekte benachrichtigt und automatisch aktualisiert werden."



Erklärung

Observer = Verhaltensmuster zur Weitergabe von Änderungen an einem Objekt an von diesem Objekt abhängige Strukturen

Akteure

- Subjekt (Observables): enthalten Liste von Beobachtern, Schnittstelle zum An- und Abmelden von Beobachtern und Schnittstelle zur Benachrichtigung von Beobachtern über Änderungen
- Beobachter (Observer): definieren Aktualisierungsschnittstelle

Anwendung wenn:

- · Änderung eines Objekts Änderungen an anderen Objekten nach sich zieht
- ein Objekt andere Objekte benachrichtigen soll, ohne diese im Detail zu kennen



Vor- und Nachteile

Vorteile durch Kapselung der Beobachter hinter Schnittstelle

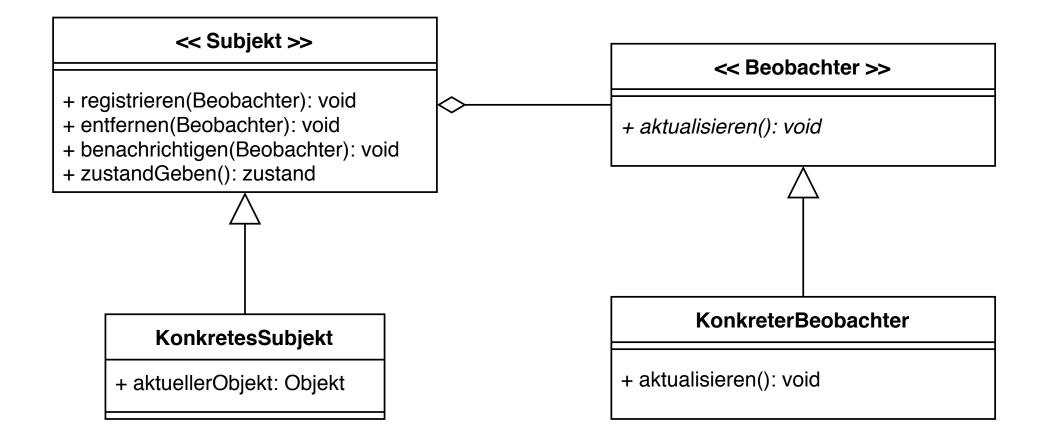
- · Hohe Wiederverwendbarkeit und Erweiterbarkeit
- Dynamik
- · Unabhängiges Wiederverwenden von Beobachteten und Beobachtern

Nachteile

- Beobachter werden immer informiert, auch wenn sie die Informationen nicht benötigen
- Bei vielen Beobachtern führen Änderungen an Beobachteten zu hohen Änderungskosten



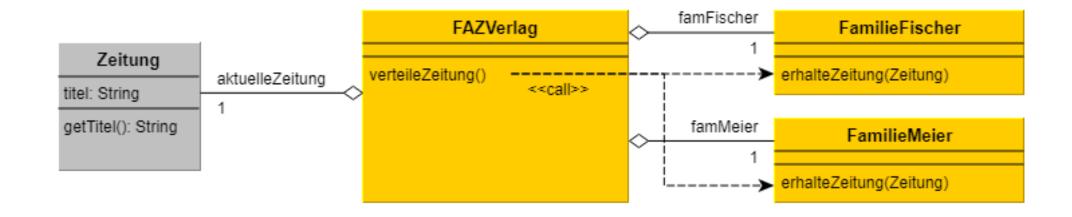
UML-Diagramm





Beispiel

Ausgangsszenario



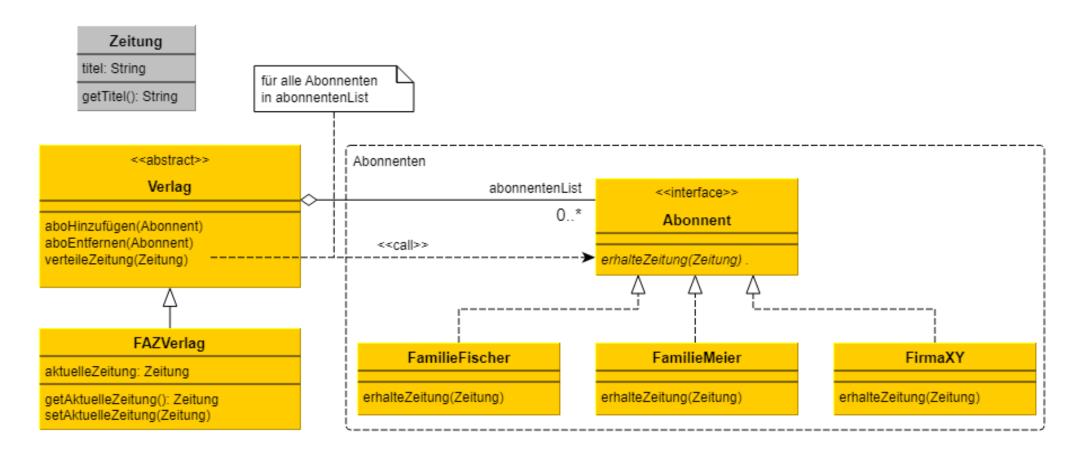


Enge Kopplung zwischen FAZVerlag und seinen Abonnenten

→ Erweiterbarkeit



Beispiel





Quellen

- Entwurfsmuster Observer https://de.wikipedia.org/wiki/Beobachter_(Entwurfsmuster)
- https://www.philipphauer.de/study/se/design-pattern/observer.php
- https://en.wikipedia.org/wiki/Observer_pattern
- https://deacademic.com/dic.nsf/dewiki/156492



Vielen Dank für Eure Aufmerksamkeit!