DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

BEOBACHTER

Lars Briem

(briem.lars@googlemail.com)

Duale Hochschule Baden Württemberg - Standort Karlsruhe

Beobachter

- Definiere 1-zu-viele Beziehung zwischen Objekten
- Benachrichtige und Aktualisiere alle abhängigen Objekte automatisch, wenn 1 Objekt den Zustand ändert

Beobachter – Einordnung

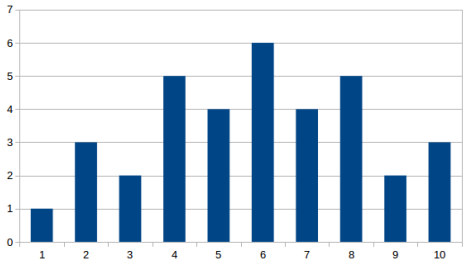
- Objektbasiertes Verhaltensmuster
- Langlebig
- Auch bekannt als
  - Dependents
  - Observer
  - Publish-Subscribe
  - Signal-Slot

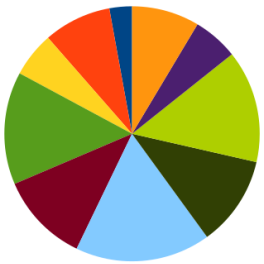
Beobachter – Motivation

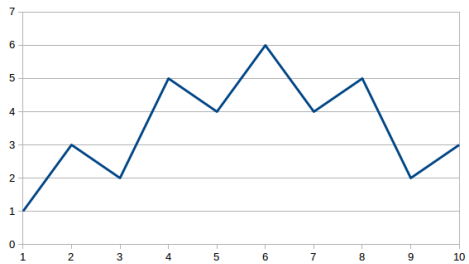
- Sicherstellung bzw. Erhaltung der Konsistenz in modularen Systemen
- Lose Kopplung der Komponenten bei Erhaltung der Konsistenz
- Bei Änderung der Daten sollen alle möglichst schnell über Änderungen informiert werden

Beobachter – Beispiel

- Verschiedene UI für gleiche Daten
  - Liniendiagramm
  - Balkendiagramm
  - Punktwolke
  - ...
- Bei Änderung der Daten sollen sich die Diagramme automatisch aktualisieren



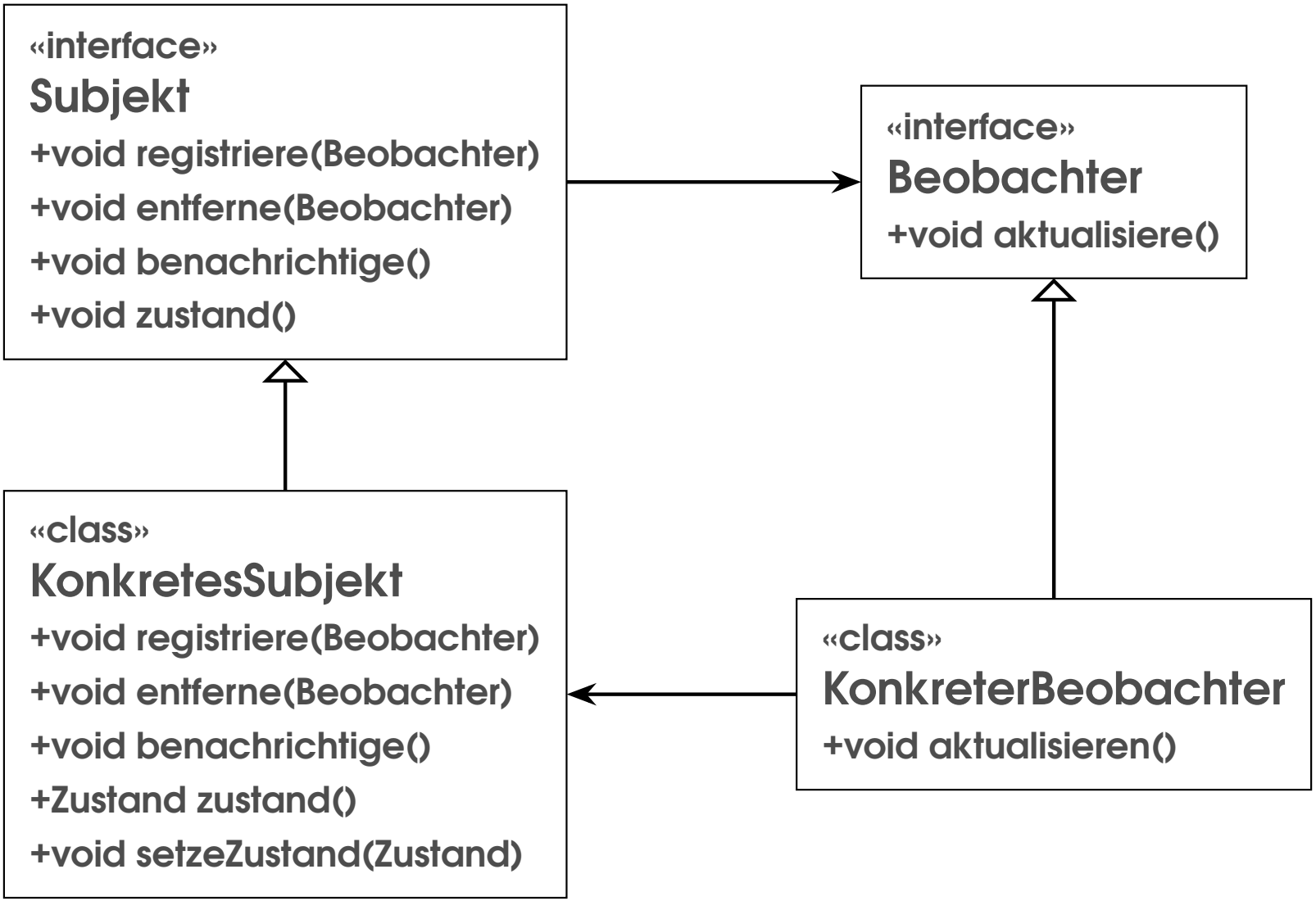




Beobachter – Anwendung

- Wenn die Änderung eines Objekts die Änderung eines anderen Objekts nach sich zieht und nicht bekannt ist, wie viele Objekte sich ändern
- Wenn ein Objekt andere benachrichtigen soll, ohne den konkreten Typ der Objekte zu kennen
  - Führt zu loser Kopplung
- Wenn eine Abstraktion mehrere Aspekte hat, die von einem anderen Aspekt derselben Abstraktion abhängen

Beobachter – Struktur



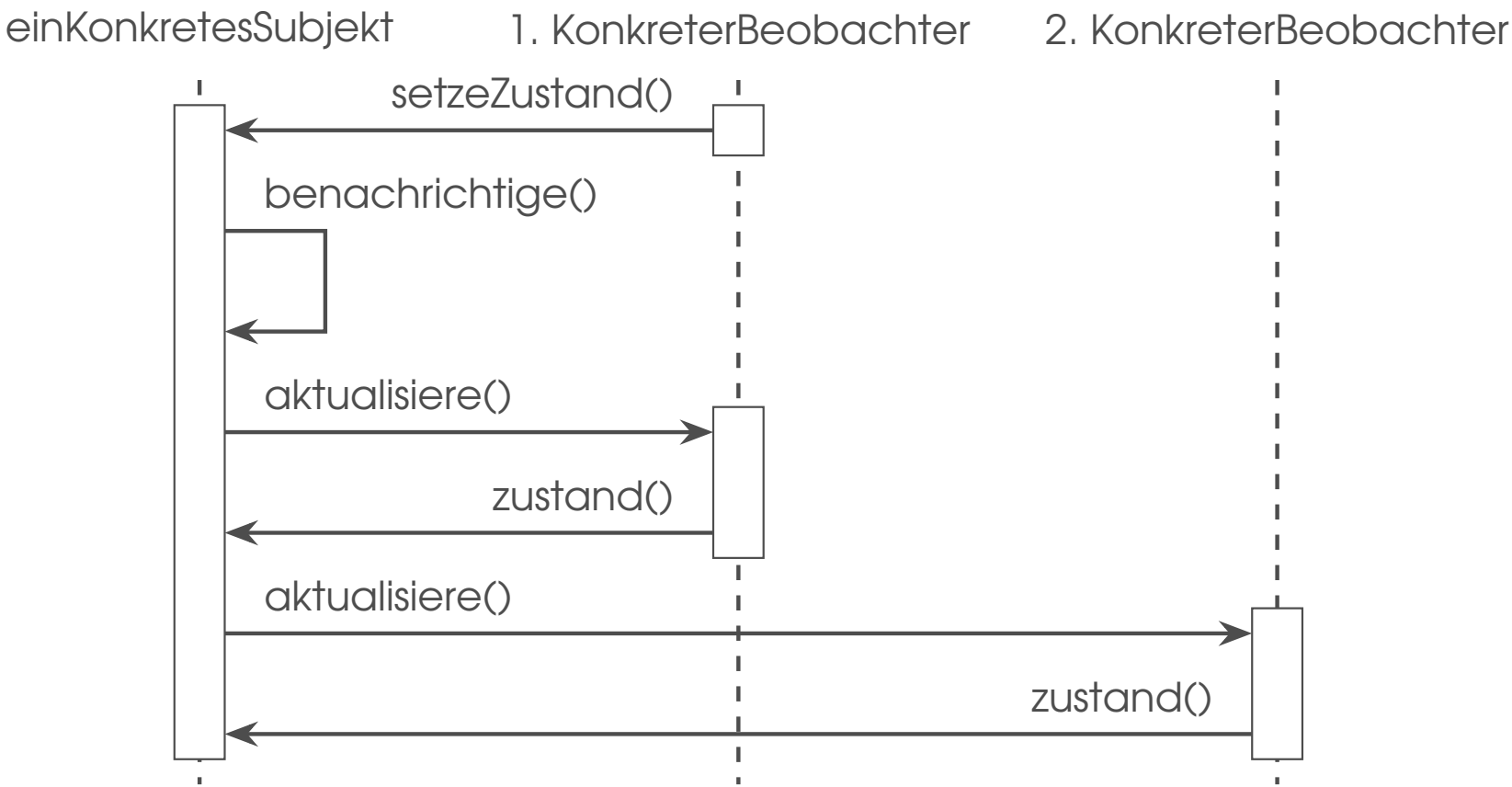
Beobachter – Akteure

- Subjekt
  - Kennt beliebig viele Beobachter
  - Stellt Interface zur Registrierung und Abmeldung von Beobachtern bereit
  - Stellt Interface zum Abrufen des aktuellen Zustands bereit
- Konkretes Subjekt
  - Speichert für Beobachter interessanten Zustand
  - Benachrichtigt Beobachter über Zustandsänderung

Beobachter – Akteure

- Beobachter
  - Definiert Schnittstelle zur Benachrichtigung bzw. Aktualisierung der Objekte
- Konkreter Beobachter
  - Hält Referenz auf konkretes Subjekt
  - Speichert Zustand, der konsistent mit Subjekt sein soll
  - Implementiert Beobachter Interface zur Aktualisierung des Zustands

Beobachter – Interaktion der Akteure



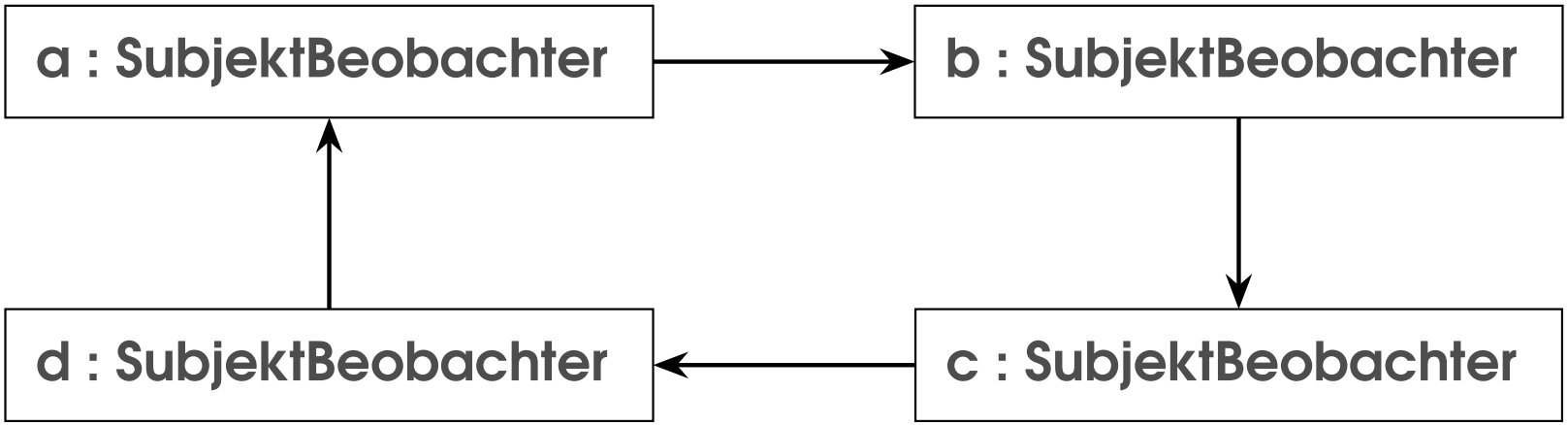
Beobachter – Auswirkungen

- Lediglich eine abstrakte Kopplung zwischen Subjekt und Beobachter über einfaches Interface
  - Subjekt und Beobachter können in unterschiedlichen Schichten liegen
  - Beide getrennt wiederverwendbar
- Automatische Broad-/Multicast Kommunikation an interessierte Objekte
  - Dynamische Menge von Beobachtern
  - Jederzeit änderbar

Beobachter – Auswirkungen

- Unerwartete Aktualisierung
  - Beobachter kennen sich nicht und wissen nicht, was eine Veränderung des Zustands bewirkt
  - Beobachtungszyklen können entstehen

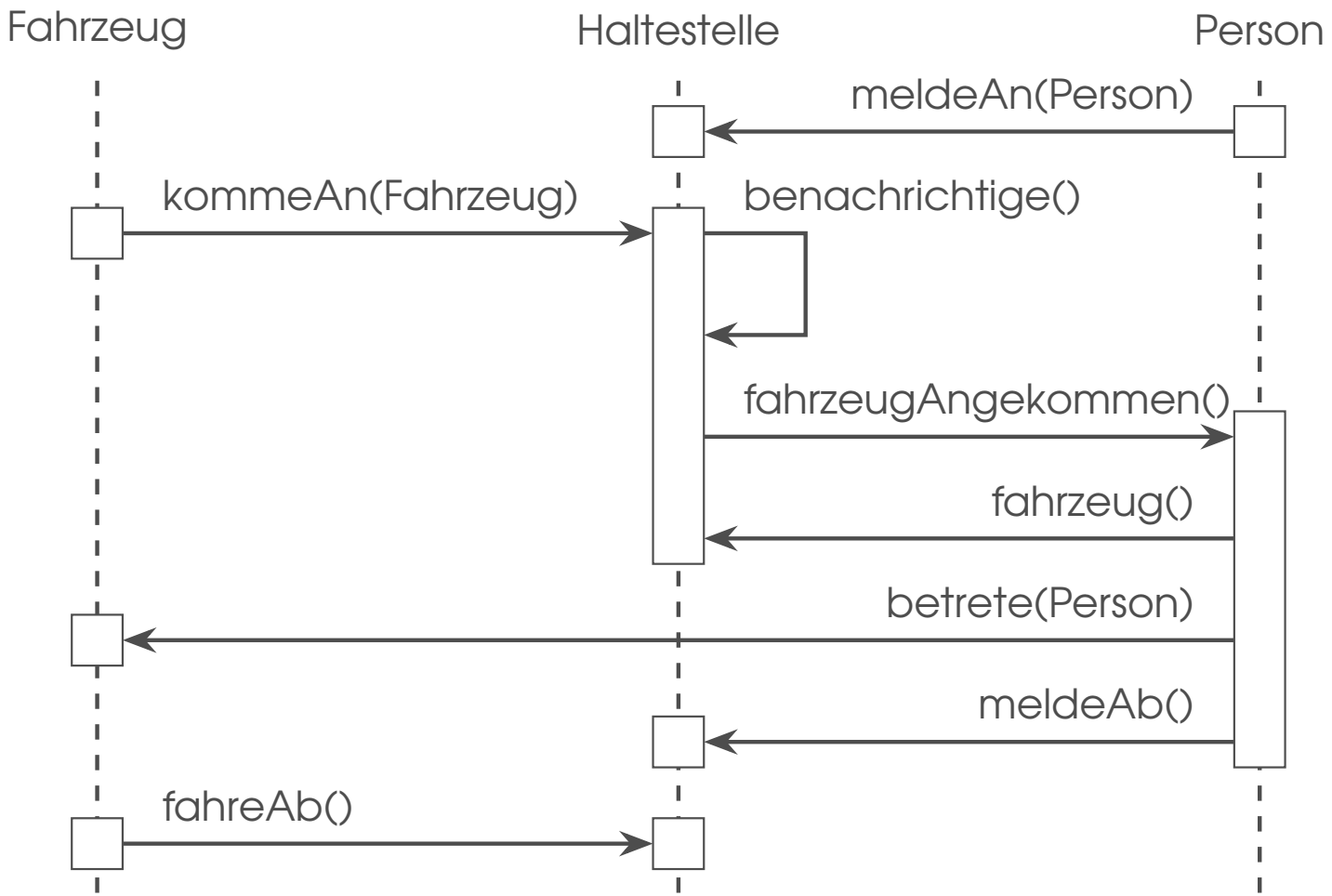
  $\Rightarrow$ Nur echte Aktualisierung weitergeben



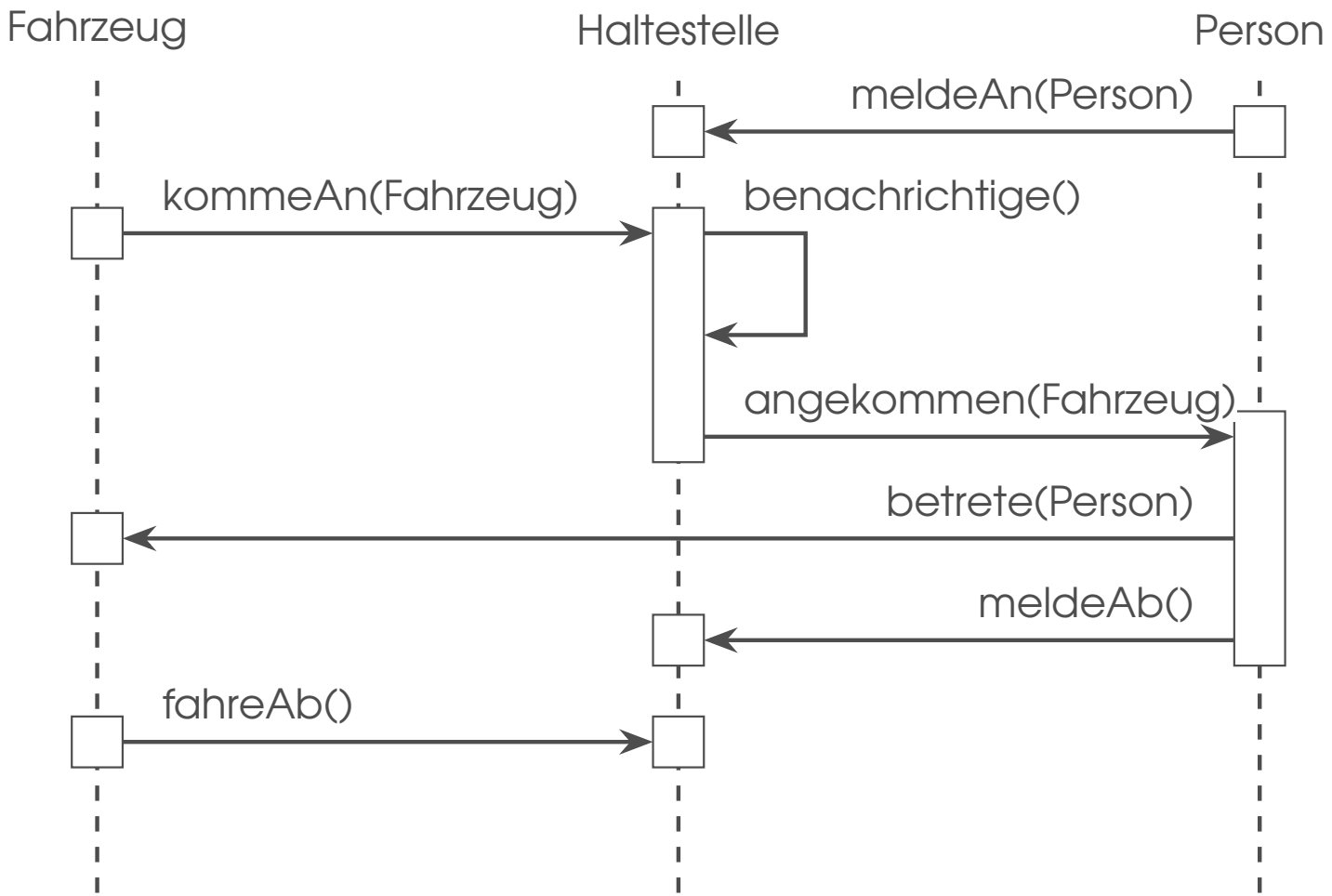
Beobachter – Arten

- Push-Modell
  - Subjekt benachrichtigt Beobachter
  - Inklusive Informationen über Änderung
  - Stärkere Kopplung des Subjekts an Beobachter, da es Annahmen trifft, was Beobachter interessiert
- Pull-Modell
  - Subjekt benachrichtigt Beobachter über Änderung (minimale Nachricht)
  - Beobachter muss sich Informationen selbst holen
  - Beobachter müssen Änderungen selbst herausfinden

Beobachter – Beispiel - Pull



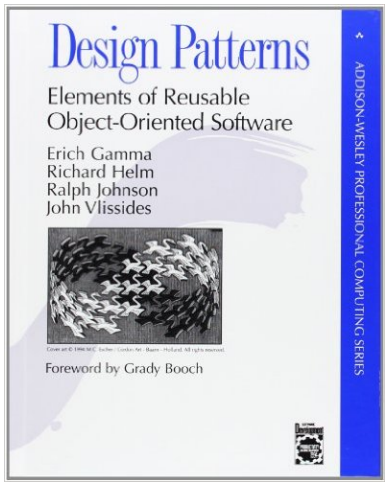
Beobachter – Beispiel - Push



Beobachter – Zusammenfassung

- Automatische Konsistenz von abhängigen Zuständen
- Koppelt die Elemente nur lose aneinander
- Sofortige Benachrichtigung bei Änderung des Zustands
- 2 Arten von Benachrichtigungen

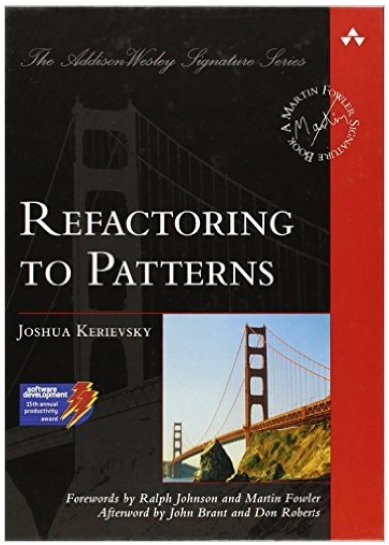
Literatur



- Design Patterns
  - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
  - Addison-Wesley
  - ISBN: 978-0201633610

Weitere Infos

- Entwurfsmuster auf YouTube
  - John Lindquist erklärt Entwurfsmuster mit StarCraft II
  - `https://www.youtube.com/playlist?list=PL8B19C3040F6381A2`



- Refactoring to Patterns
  - Joshua Kerievsky
  - Addison-Weslay
  - ISBN: 978-0321213358