

# ENTWURFSMUSTER

---

Lars Briem

([briem.lars@googlemail.com](mailto:briem.lars@googlemail.com))

Duale Hochschule Baden Württemberg - Standort Karlsruhe

- ▶ Warum Muster
    - ▶ Muster beschreiben wiederkehrende Probleme in unserer Umwelt
      - ▶ Inklusive des Kerns einer Lösung dafür
      - ▶ Lösungen können beliebig oft auf unterschiedliche Art ausgeführt werden
    - ▶ Vergleichbare Muster existieren in der Architektur und Natur
- ⇒ Muster helfen Probleme effizient zu lösen

# Entwurfsmuster

- ▶ Elemente wiederverwendbarer objekt-orientierter Software
  - ▶ Helfen Probleme zu lösen
  - ▶ Liefern ein erprobtes Konzept
  - ▶ Basieren auf realen Entwicklungen
  - ▶ Offenbaren Beziehungen tiefergehender Strukturen und Mechanismen

⇒ IKEA-Baukastensystem für OOP

# Nutzen von Entwurfsmustern

- ▶ Vermittlung von Wissen auf abstraktem Niveau
  - ▶ Räder werden nicht immer wieder neu erfunden
- ▶ Ausprägung einer höherwertigen Sprache in OOP
  - ▶ Vereinfachen und beschleunigen die Kommunikation zwischen Entwicklern

# Nutzen von Entwurfsmustern

- ▶ Helfen komplexer werdende Softwaresysteme zu beherrschen
  - ▶ Größere Bausteine helfen, den Überblick zu behalten
  - ▶ Siehe auch integrierte Schaltkreise in der Elektronik

# Gliederung von Entwurfsmustern

- ▶ Zweck bzw. Verwendung
  - ▶ Erzeugungsmuster
  - ▶ Strukturmuster
  - ▶ Verhaltensmuster
- ▶ Geltungsbereich
  - ▶ Auf Klassenebene
    - ▶ Statisch
    - ▶ Wird beim Kompilieren festgelegt
  - ▶ Auf Objektebene
    - ▶ Dynamisch
    - ▶ Wird zur Laufzeit festgelegt

# Erzeugungsmuster

- ▶ Trennen die Erstellung der Objekte von deren Verwendung
- ▶ System wird unabhängig von der Zusammensetzung bzw. Implementierung der Objekte

# Erzeugungsmuster

- ▶ Konkrete Instanzen werden einfacher ersetzbar für anderes Verhalten
- ▶ Kapseln Wissen über
  - ▶ Konkret verwendete Klasse bzw. Implementierung
  - ▶ Erstellung und Kombination der Objekte



# Strukturmuster

- ▶ Kombinieren Klassen und Objekte, um größere Strukturen zu schaffen
- ▶ Kombination von mehreren Interfaces
- ▶ Übersetzung von einem zum anderen Interface
- ▶ Kombination von Funktionalität zur Laufzeit
- ▶ Sparen von Ressourcen bzw. Laufzeit

# Verhaltensmuster

- ▶ Zuweisung von Verantwortlichkeiten an Objekte
- ▶ Austausch von Algorithmen bzw. Verhalten
- ▶ Kommunikation zwischen Objekten
- ▶ Steuerung des Kontrollflusses einer Anwendung zur Laufzeit (auch komplexer)
  - ▶ Verbindung zwischen Elementen der Software

# Übersicht der Entwurfsmuster

## Ein Auszug der "Gang of Four" Muster

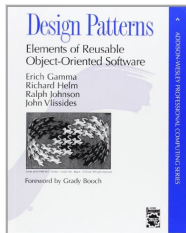
	Erzeugungsmuster	Strukturmuster	Verhaltensmuster
Klasse	Fabrikmethode	Adapter(Klasse)	Interpreter Schablonenmethode
Objekt	Abstrakte Fabrik Einzelstück Erbauer Prototyp	Adapter(Objekt) Brücke Dekorierer Fassade Fliegengewicht Kompositum Stellvertreter	Beobachter Besucher Iterator Kommando Memento Strategie Vermittler Zustand Zuständigkeitskette

# Integration in die Entwicklung

- ▶ Während dem kompletten Entwicklungsprozess einsetzbar
  - ▶ Entwurf, Implementierung, Refactoring
- ▶ Hauptsächlich als Kommunikationsmittel einsetzen
  - ▶ Implementierungsdetails können stark variieren
- ▶ Vorsicht für "zu viel" Entwurfsmustern
  - ▶ Bei unnötiger Verwendung von Entwurfsmustern wird der Code unnötig komplex
  - ▶ Nur einsetzen, wenn notwendig

# Zusammenfassung

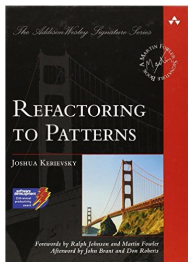
- ▶ Liefern Lösungen für wiederkehrende Probleme
- ▶ Sind in mehrere Kategorien eingeteilt
- ▶ Bauen auf allgemeinen Programmier Prinzipien auf
- ▶ Es existieren auch Anti-Entwurfsmuster
  - ▶ Big Ball of Mud
  - ▶ Spaghetti Code
  - ▶ ...



- ▶ 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-Wesley
  - ▶ ISBN: 978-0321213358