

---

# Entwurfsmuster

Die Schinkenwürfel im Spaghetti-Code

# Die sieben W

---

- Was sind Entwurfsmuster?
- Woher kommen (Entwurfs-)Muster?
- Warum kümmern wir uns um sie?
- Welche Entwurfsmuster gibt es?
- Wann sollte man sie einsetzen?
- Wo steht mehr dazu?
- Wie geht's hier weiter?

# Was sind Entwurfsmuster?

- Elemente wiederverwendbarer objekt-orientierter Software

„Jedes Muster beschreibt ein in unserer Umwelt beständig wiederkehrendes Problem und erläutert den Kern der Lösung für dieses Problem, so dass Sie diese Lösung beliebig oft anwenden können, ohne sie jemals ein zweites Mal gleich auszuführen.“

Christopher Alexander, A Pattern Language

- Kein fertiger Code oder festes Design
- Lösungsansatz für typische Probleme in der Softwareentwicklung
- Metapher: IKEA-Regalsystem für OOP

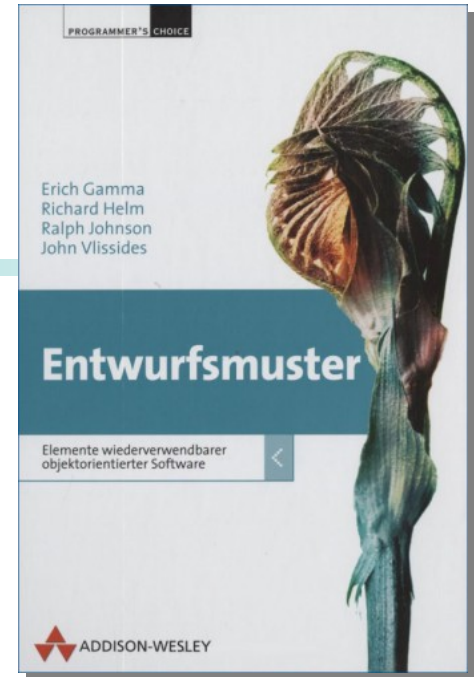
# Was sind Entwurfsmuster?

---



# Woher kommen (Entwurfs-)Muster?

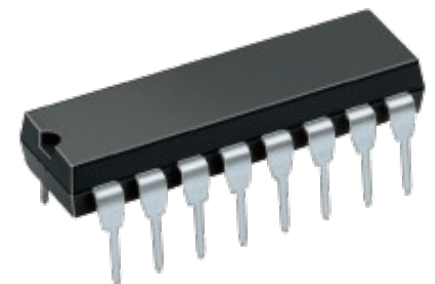
- „Gang of Four“-Buch erschien 1994
  - Das meistreferenzierteste Buch in der Softwaretechnik
  - Alt, aber größtenteils nicht veraltet
- Ursprüngliche Idee stammt von Christopher Alexander, der über (Haus-)Architektur-Muster geschrieben hat
- Ende 1980er: Ausreichend Erfahrung mit OO-Programmierung, um wiederkehrende Lösungen zu erkennen



# Warum kümmern wir uns um sie?

---

- Wissensvermittlung auf abstraktem Niveau
  - Verhindert das Wiedererfinden vieler Räder
- Beginn einer höherwertigen Sprache in der Objektorientierten Programmierung
  - Beschleunigt und stärkt Kommunikation zwischen Entwicklern
- Softwaresysteme werden komplexer
  - Immer größere „Bausteine“ notwendig, um System noch erfassen zu können
  - Analog: „Integrierte Schaltkreise“ in der Elektronik



# Welche Entwurfsmuster gibt es?

---

- Im Originalbuch: 23 beschriebene Muster
- Mittlerweile ([wikipedia.en](https://en.wikipedia.org/wiki/Design_patterns)): ~50 Muster
- Vier Kategorien
  - Erzeugungsmuster
  - Strukturmuster
  - Verhaltensmuster
  - Nebenläufigkeitsmuster (Concurrency pattern)
    - Behandeln typische Lösungen für Multithreading



# Übersicht über Entwurfsmuster

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	Factory Method	Adapter	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Proxy	Chain of Responsibility Command Iterator Mediator Memento Flyweight Observer State Strategy Visitor



# Übersicht über Entwurfsmuster

## The Sacred Elements of the Faith

the holy  
origins

the holy  
structures

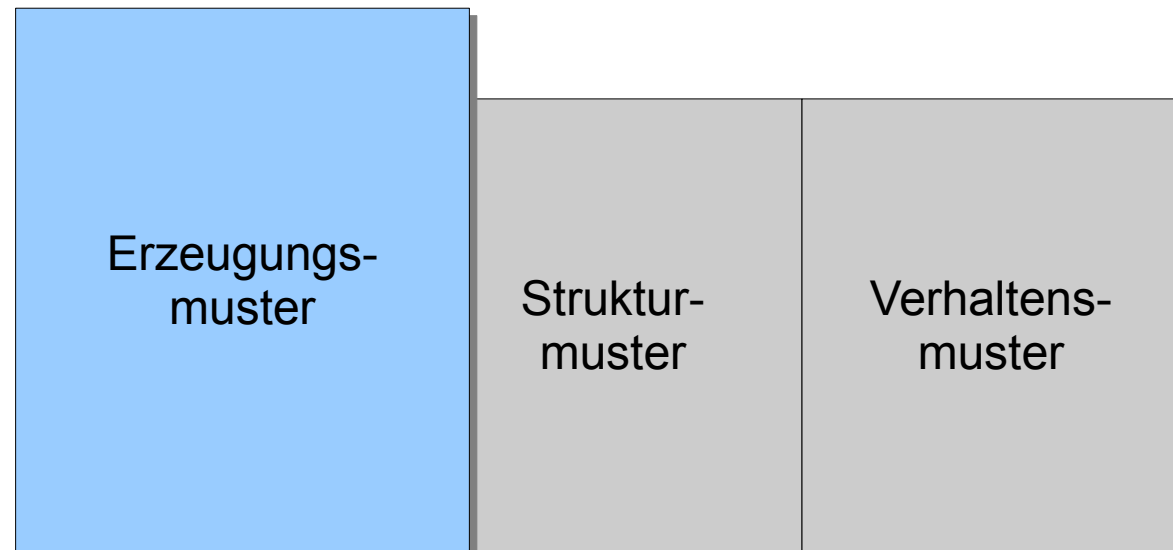
107 FM Factory Method	the holy behaviors					139 A Adapter	
117 PT Prototype	127 S Singleton				223 CR Chain of Responsibility	163 CP Composite	175 D Decorator
87 AF Abstract Factory	325 TM Template Method	233 CD Command	273 MD Mediator	293 O Observer	243 IN Interpreter	207 PX Proxy	185 FA Façade
97 BU Builder	315 SR Strategy	283 MM Memento	305 ST State	257 IT Iterator	331 V Visitor	195 FL Flyweight	151 BR Bridge

Image by Vince Huston: <http://www.vincehuston.org/dp/>

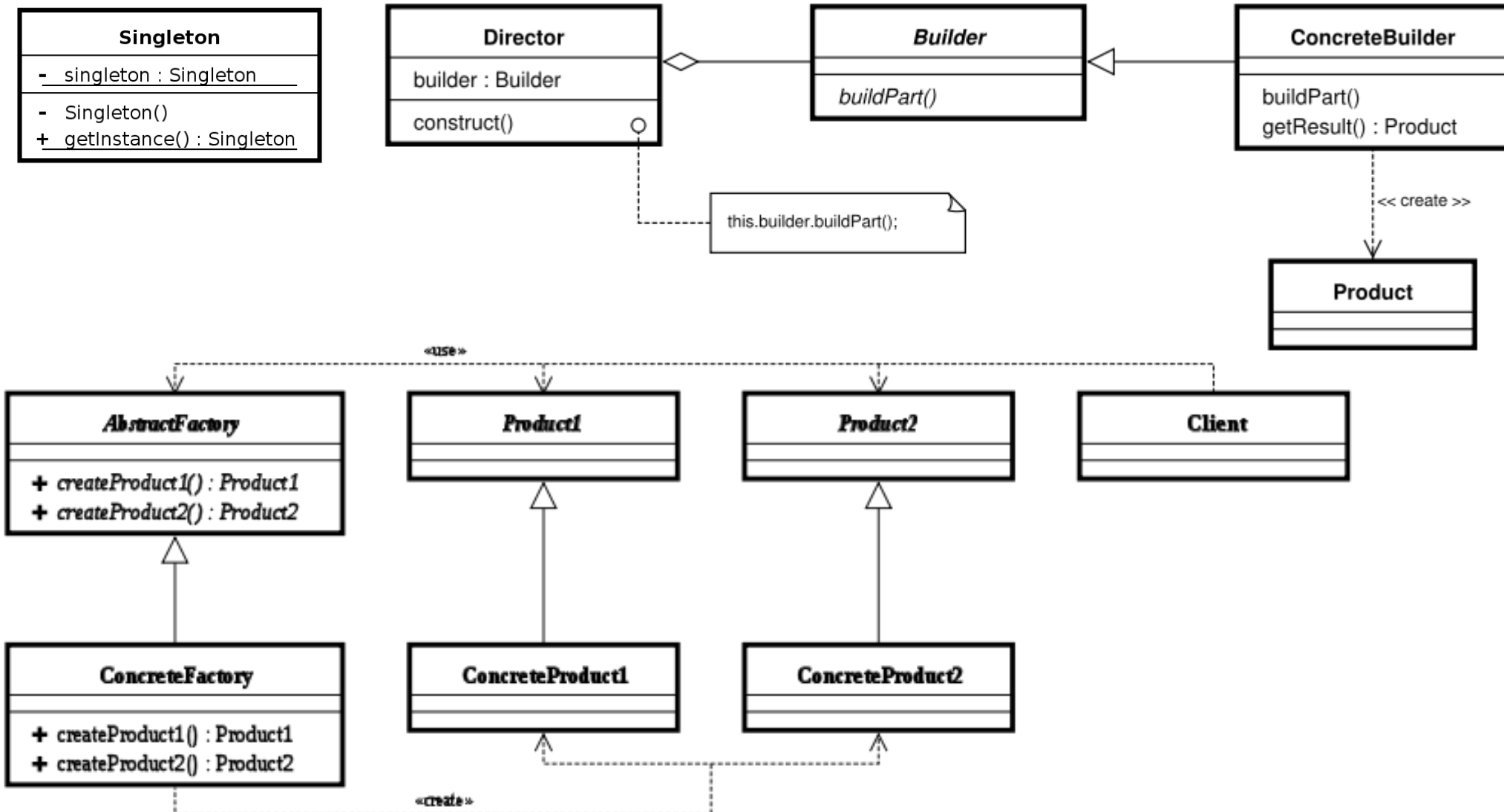
# Erzeugungsmuster

---

- Beziehen sich immer auf die Erstellung von Instanzen
- Bieten Strukturen, wenn die normale, direkte Objekterstellung ungewollte Komplexität mit sich bringen würde
- Grundideen:
  - Verbergen der konkreten Typen
  - Verbergen des „Geburtvorgangs“

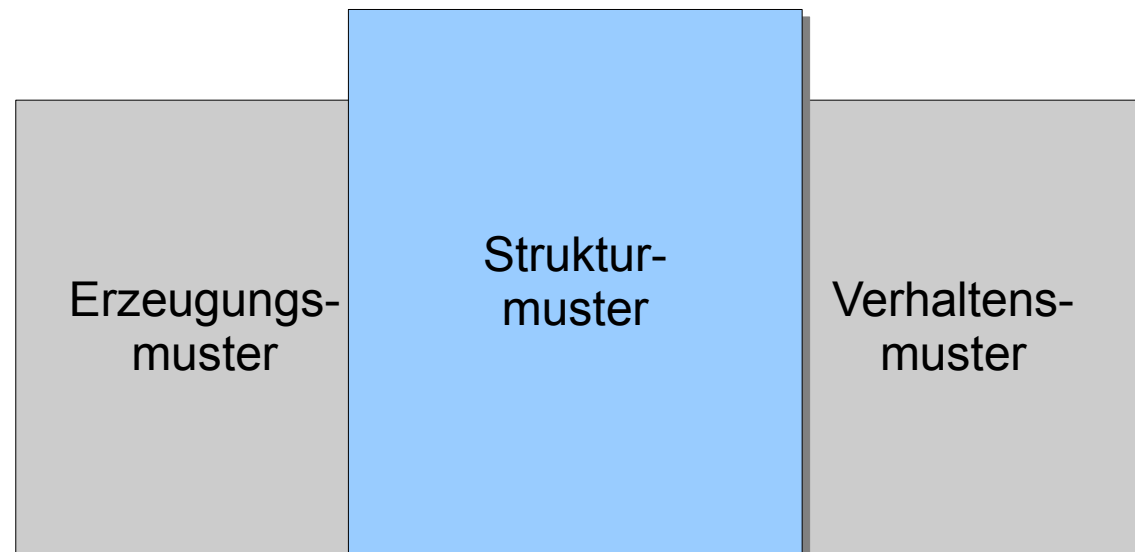


# Beispiele für Erzeugungsmuster

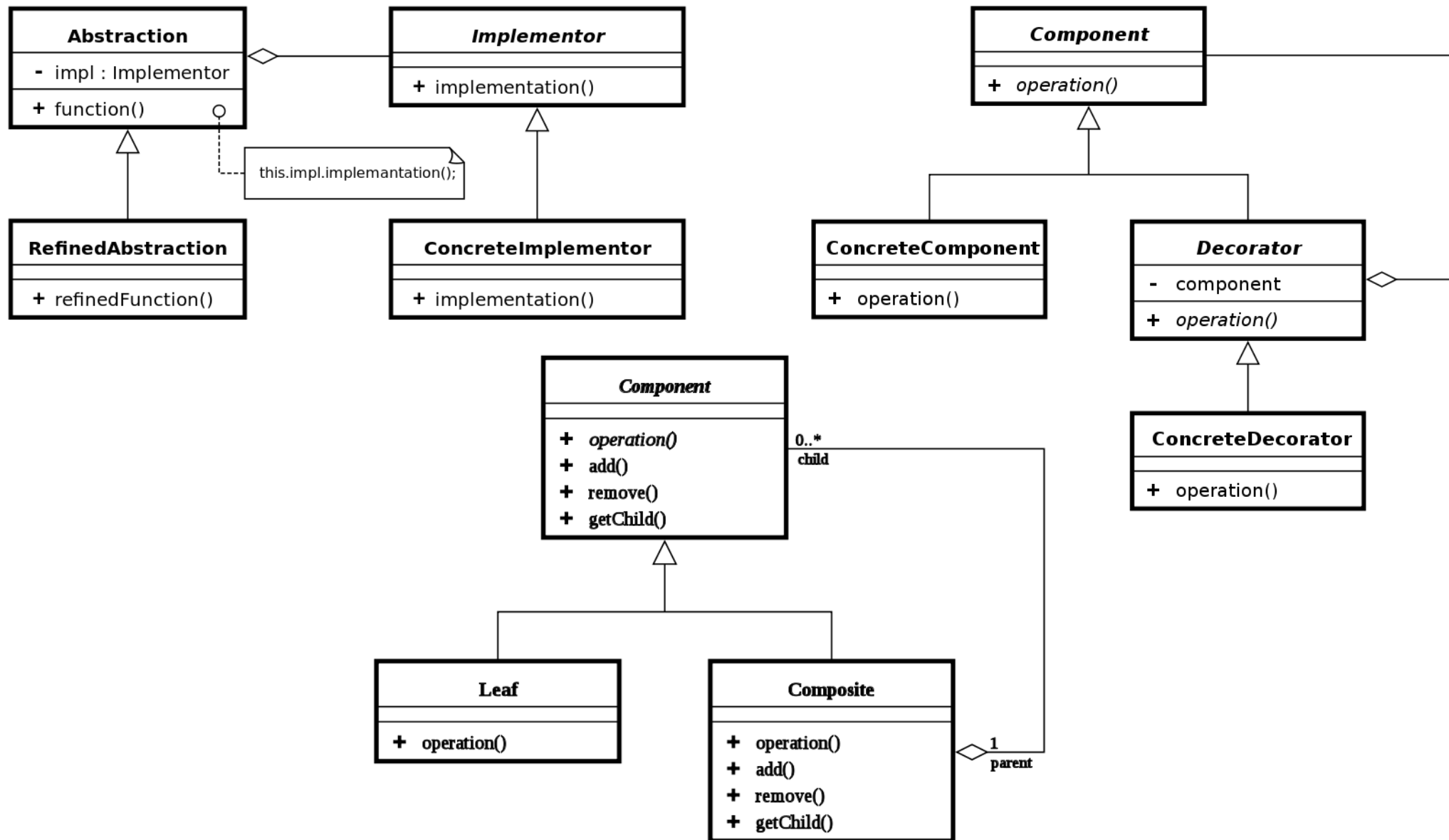


# Strukturmuster

- Vereinfachen das Design der Software durch Beeinflussung der Beziehungen zwischen Objekten und Klassen
- Komposition von Klassen und Objekten
  - Herausbilden übergeordneter Strukturen
- Hauptwerkzeuge:
  - Vererbung zwischen Klassen
  - Assoziationen zu anderen Objekten



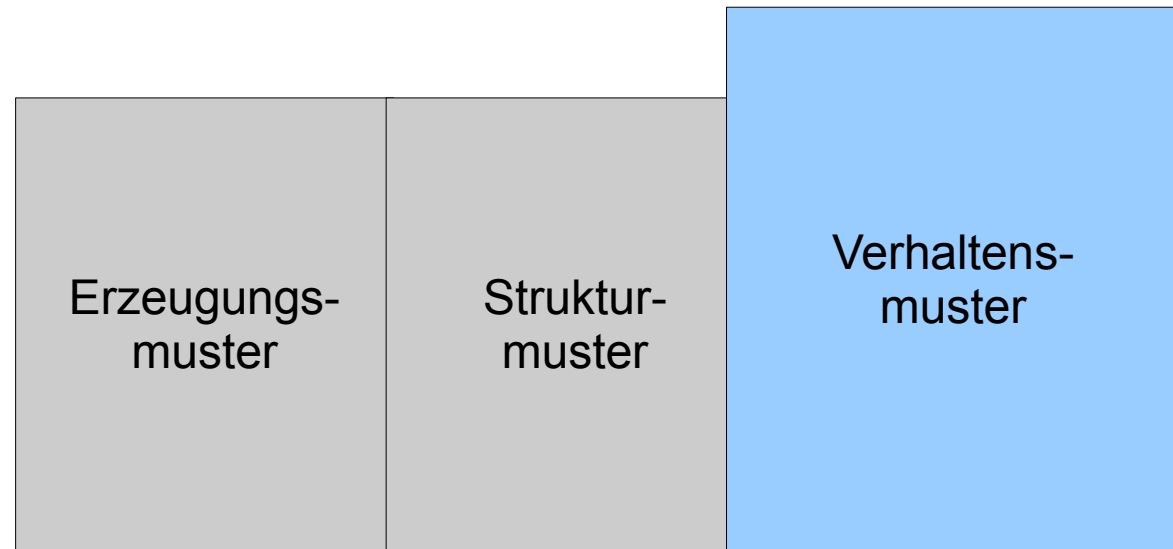
# Beispiele für Strukturmuster



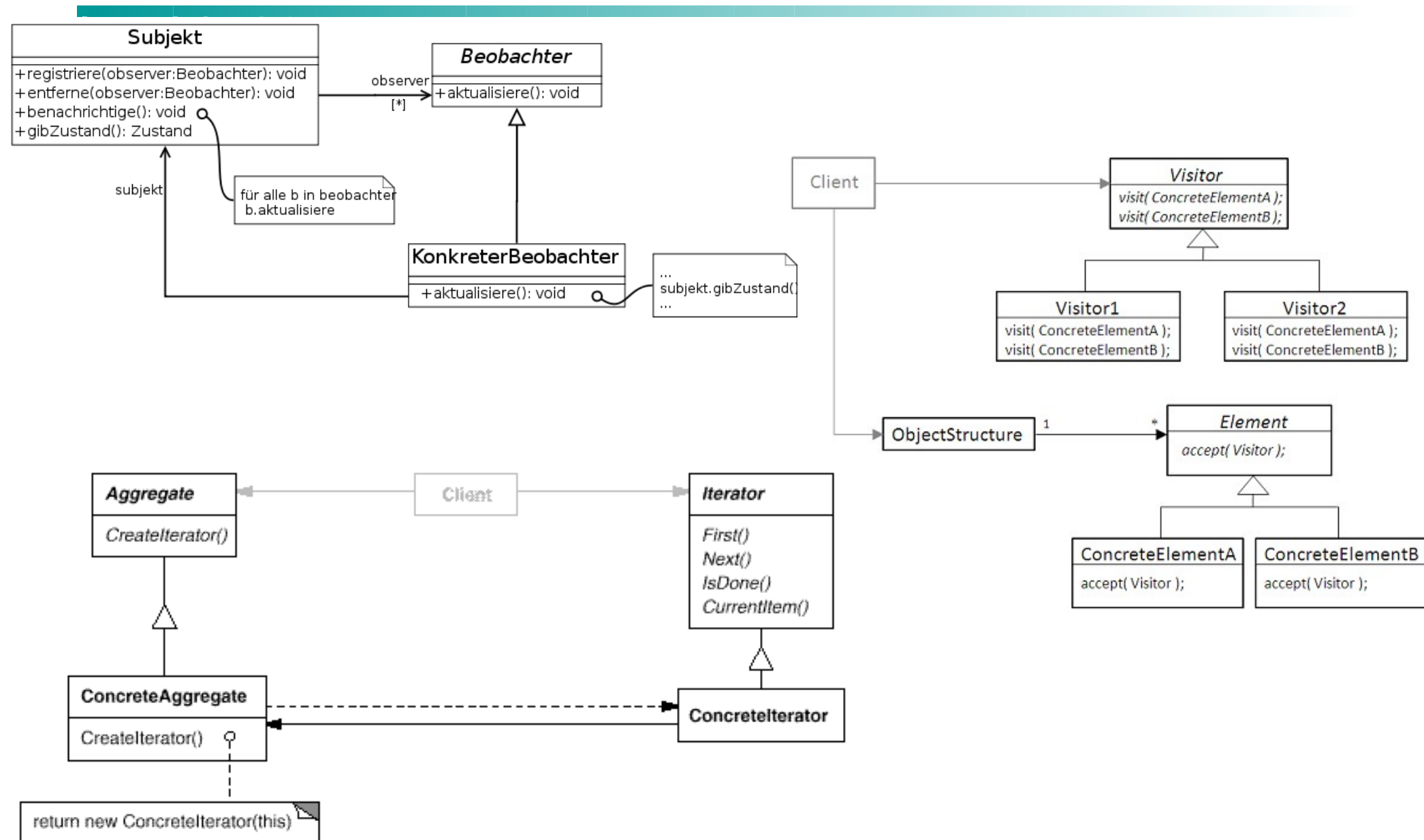
# Verhaltensmuster

---

- Beschreibung häufige Konstellationen der Zusammenarbeit zwischen Objekten
- Flexibilisierung des Verhaltens der Software
  - Oft nicht gut im Klassendiagramm erkennbar
  - Zeigen sich erst zur Laufzeit, hoch dynamisch
- Hauptwerkzeuge
  - (polymorphe) Methodenaufrufe
  - Dynamisch änderbare Assoziationen



# Beispiele für Verhaltensmuster



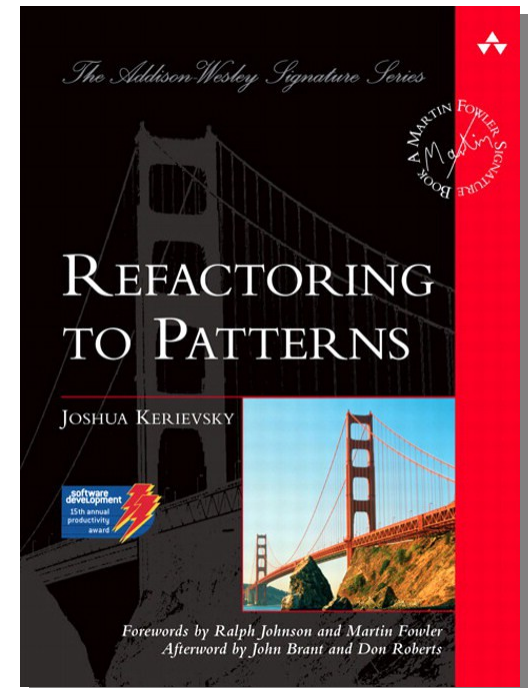
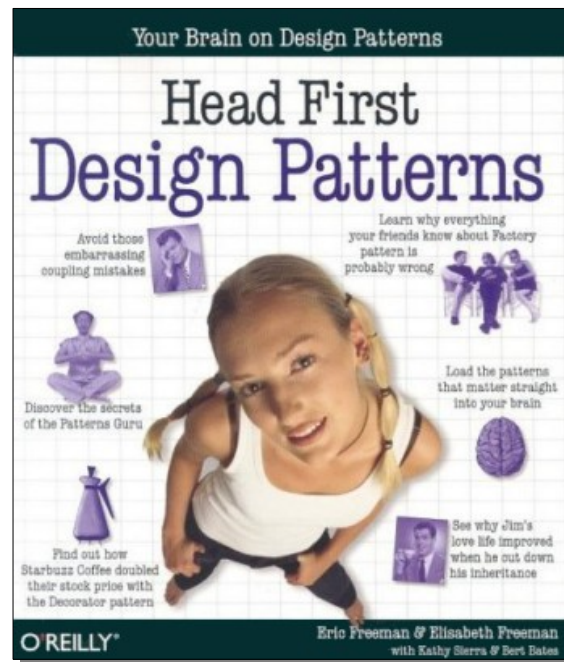
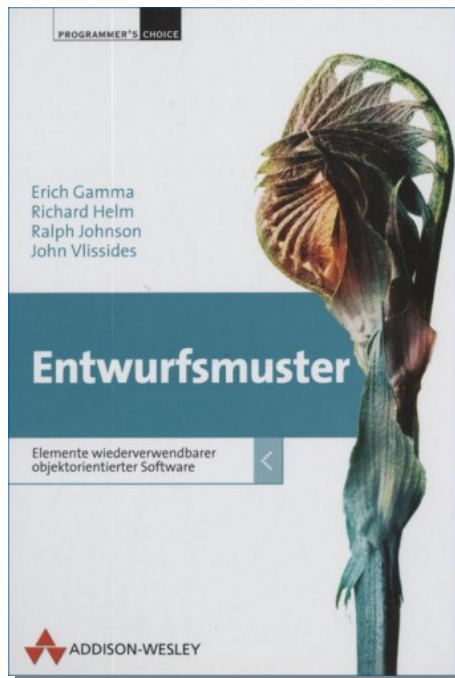


# Wann sollte man sie einsetzen?

---

- Im Entwurf, während der Implementierung, bei der Wartung (Refactoring)
- Lieber sparsam einsetzen
  - „Beginning developers never met a pattern [...] they didn't like.“ - [Jeff Atwood](#)
    - „Patterns, like all forms of complexity, should be avoided until they are absolutely necessary.“
- Vor allem als konzeptionelle Bausteine in der Kommunikation einsetzen
  - Implementierung kann stark variieren

# Wo steht mehr dazu?



# Gibt es auch Schulfernsehen?

- Youtube ist das neue Fernsehen
- PatternCraft von John Lindquist
  - Entwurfsmuster erklärt mit StarCraft II
  - <https://www.youtube.com/playlist?list=PL8B19C3040F6381A2>



# Wie geht's hier weiter?

---

- Singleton – Erzeugungsmuster
- Beobachter – Verhaltensmuster
- Dekorierer – Strukturmuster
- Erbauer – Erzeugungsmuster
- Iterator – Verhaltensmuster
- Kompositum – Strukturmuster
- Event Bus – Verhaltensmuster
- Ableiten von Entwurfsmustern