

Entwurfsmuster

- **Generalisierte Lösungsideen**
 - Beschreiben den Lösungsweg
- **Klassen**
 - **Architekturmuster:** Lösungen beim Grobentwurf (z.B. Aufbau einer Mehrschichtenarchitektur)
 - Entwurfsmuster beim Feinentwurf:
unabhängig von Programmiersprache und –umgebung
 - **Idiome:** programmiersprachliche Lösungsbeschreibungen (mit der Sprache Smalltalk eingeführt)

Überblick über Entwurfsmuster

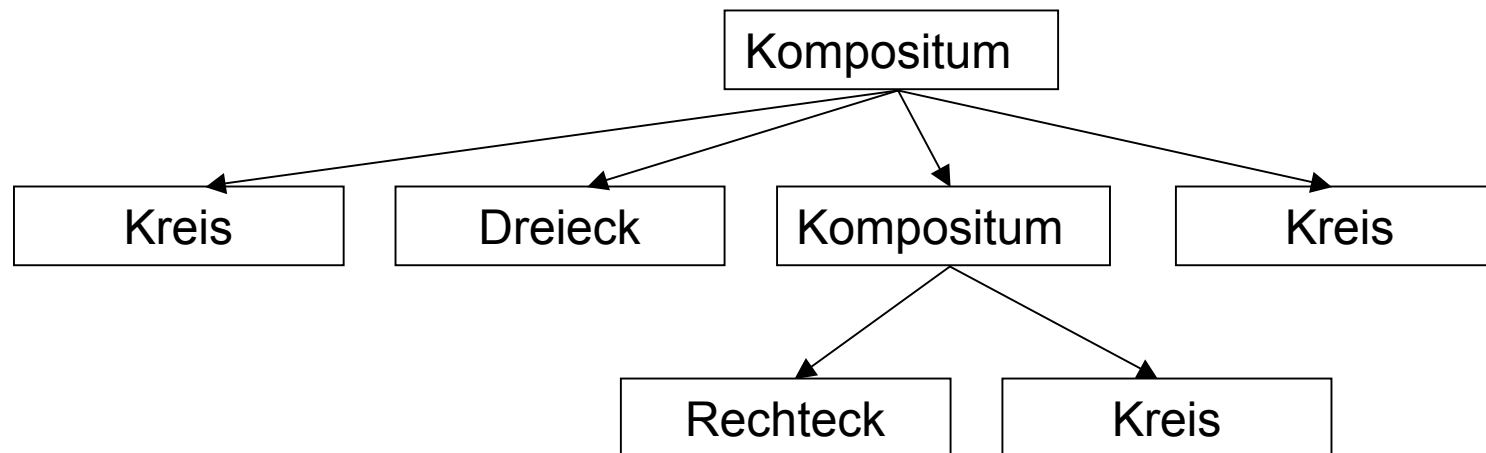
- Adapter:
 - Versieht bestehende Klasse mit neuer Schnittstelle
- Brücke (bridge):
 - Trennt Schnittstelle einer Klasse von ihrer Implementierung
 - Plattformunabhängigkeit
- Dekorierer (Decorator):
 - Ermöglicht es Eigenschaften eines konkreten Objekts unabhängig von seiner Klasse dynamisch zu verändern und zu erweitern
- Fassade (Facade):
 - Einfache Schnittstelle für Menge von Klassen/Subsystem
 - Verbirgt Komplexität

Weitere Entwurfsmuster

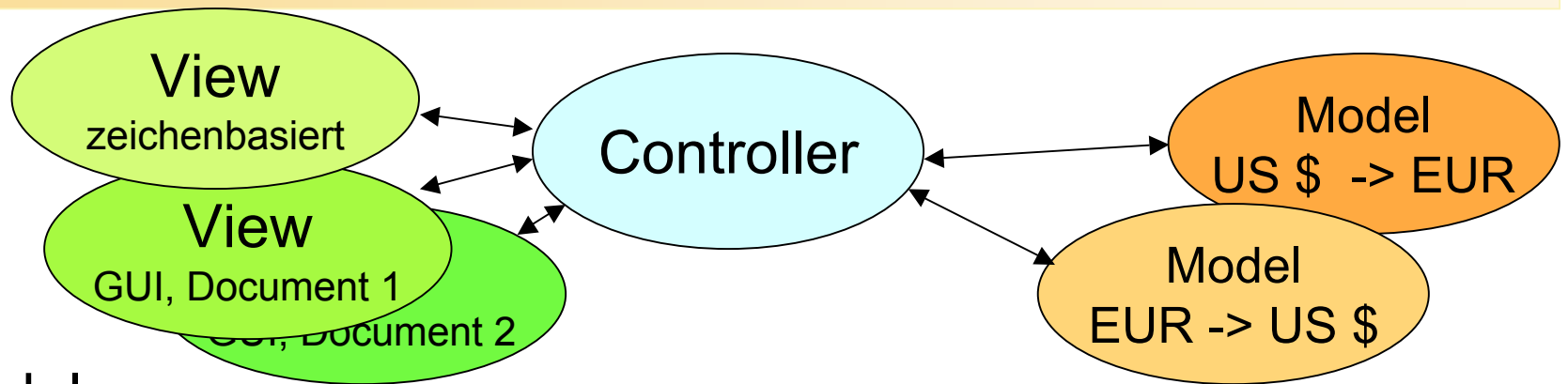
- **Memento:**
 - Möglichkeit, ehemalige Objektzustände zu restaurieren
 - Realisierung von undo-Funktionen
- **Beobachter (Observer):**
 - Muster beschreibt Mechanismus, der alle von einem Objekt abhängigen Objekt von einer Zustandsänderung (attribut) informiert
- **Besucher (Visitor):**
 - Beschreibt, wie eine Operation iterativ über eine Menge von Objekten ausgeführt werden kann
 - Beteiligte Objekte müssen keine Kenntnis vom Besucher (Identität, Operationen) haben

Kompositum-Muster

- Beschreibt baumartige Aggregationen
 - Können als einzelnes Objekt oder als Zusammensetzung von Objekten benutzt werden
 - Geometriebeispiel:



Model/View/Controller



- **Model:**
 - Implementiert einen bestimmten Algorithmus
 - Unabhängig von Umgebung
- **View:**
 - Kommuniziert mit der Umgebung
 - Implementiert Ein-/Ausgabeschnittstelle für Model
- **Controller:**
 - Steuert Datenaustausch zwischen Model und View