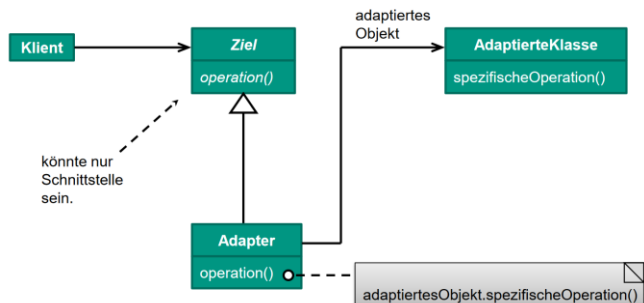


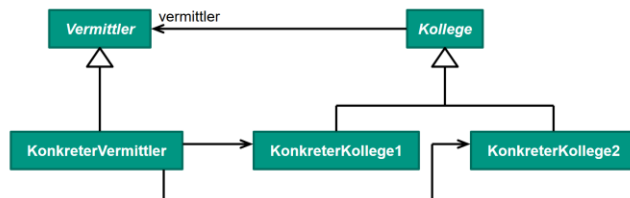
Entkopplungsmuster

Teilen ein System in mehrere Einheiten, so dass einzelne Einheiten unabhängig voneinander erstellt, verändert, ausgetauscht und wiederverwendet werden können. Der Vorteil von Entkopplung ist, dass ein System durch lokale Änderungen verbessert, angepasst und erweitert werden kann, ohne das ganze System zu modifizieren.

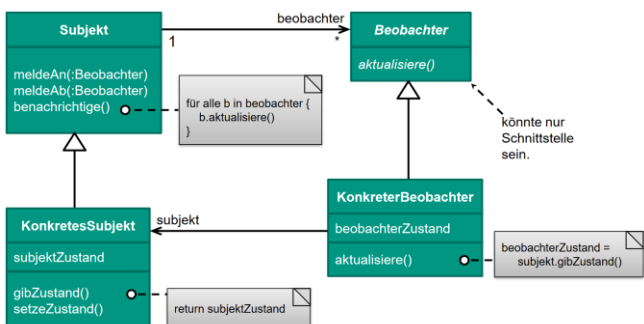
ADAPTER Passt die **Schnittstelle** einer Klasse an eine andere von ihren Klienten erwartete Schnittstelle an.



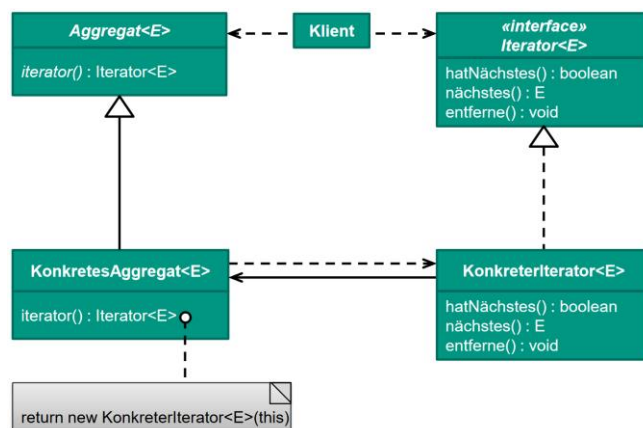
VERMITTLER Definiere ein Objekt, welches das **Zusammenspiel** einer Menge von Objekten in sich kapselt.



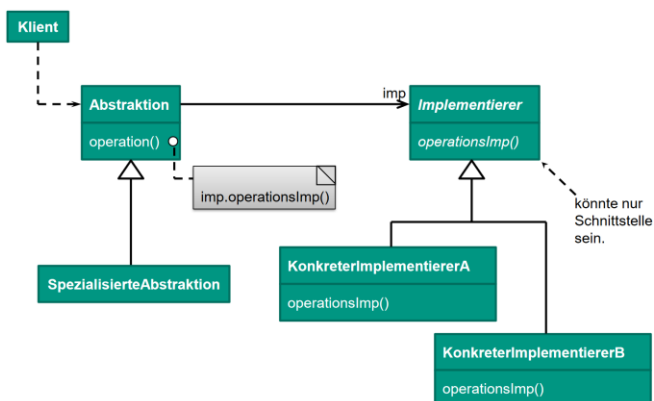
BEOBSACHTER Definiert eine **1-zu-n** Abhängigkeit zwischen Objekten, so dass die Änderung eines Zustandes eines Objektes dazu führt, dass alle abhängigen Objekte **benachrichtigt** und **automatisch aktualisiert** werden.



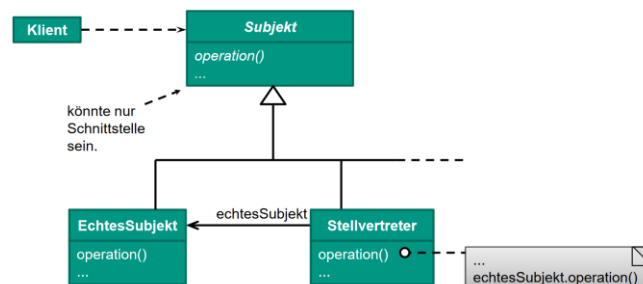
ITERATOR Ermöglicht den **sequentiellen Zugriff** auf die Elemente eines zusammengesetzten Objekts, ohne seine zugrundeliegende Repräsentation offenzulegen.



BRÜCKE Entkoppelt eine **Abstraktion** von ihrer **Implementierung**, so dass beide unabhängig voneinander variiert werden können.



STELLVERTRETER Kontrolliert den Zugriff auf ein Objekt mit Hilfe eines vorgelagerten **Stellvertreterobjekts**.

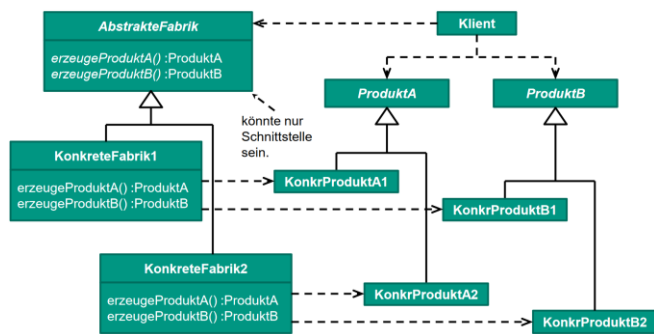


Varianten-Muster

Ziehen Gemeinsamkeiten aus einer Menge von Elementen heraus, um diese einheitlich verwendbar zu machen und Redundanz zu vermeiden.

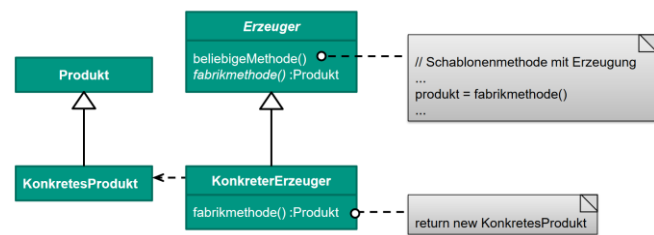
ABSTRAKTE FABRIK

Bietet eine Schnittstelle zum Erzeugen von Familien verwandter oder voneinander abhängiger Objekte, ohne ihre konkreten Klassen zu benennen.



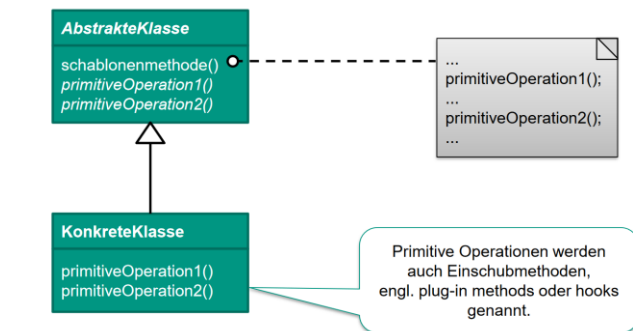
FABRIKMETHODE

Definiert eine Klassenschnittstelle mit Operationen zum Erzeugen eines Objekts, aber lasse Unterklassen entscheiden, von welcher Klasse das zu erzeugende Objekt ist. Ermöglicht, das Erzeugen von Objekten an Unterklasse zu delegieren.



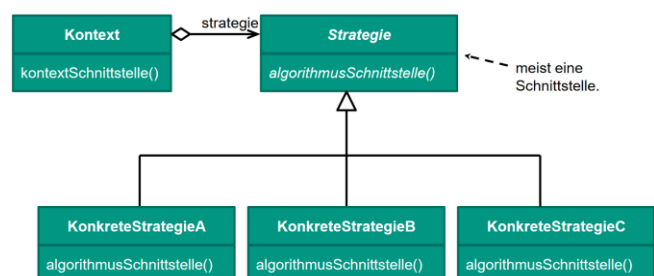
SCHABLONENMETHODE

Definiert das Skelett eines Algorithmus in einer Operation und delegiere einzelne Schritte an Unterklassen.



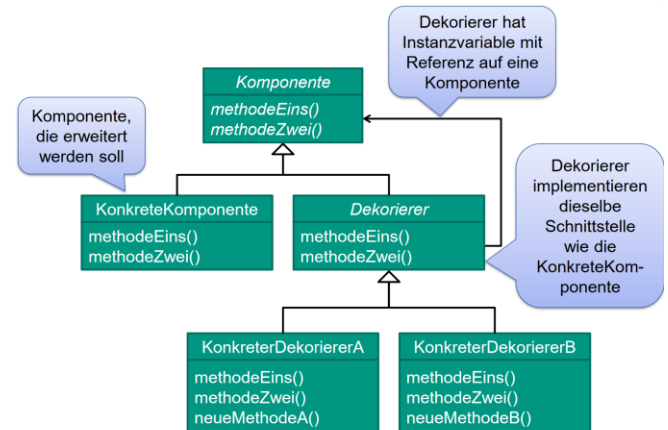
STRATEGIE

Definiert eine Familie von Algorithmen, kapselt sie und macht sie austauschbar. Das Strategiemuster ermöglicht es, den Algorithmus unabhängig von nutzenden Klienten zu variieren.



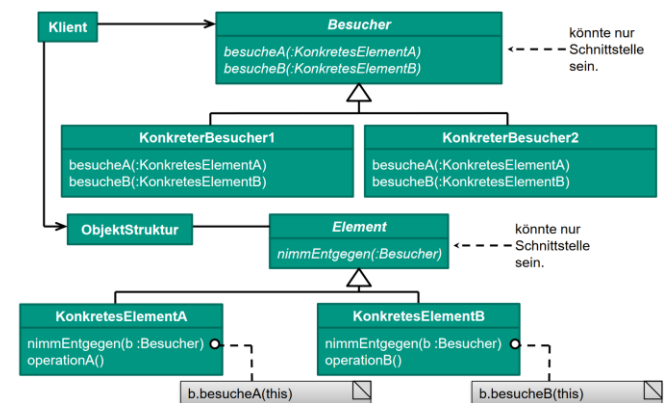
DEKORIERER

Fügt dynamisch neue Funktionalität zu einem Objekt hinzu.



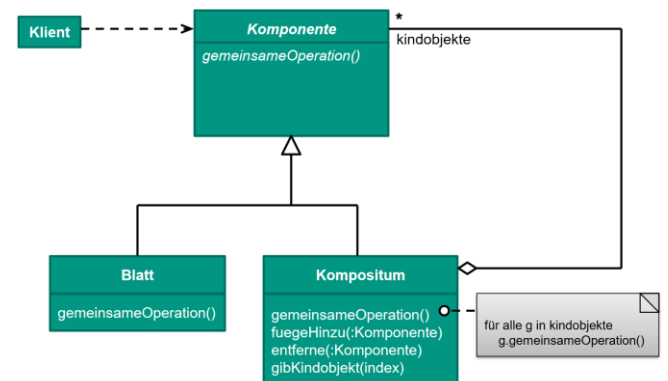
BESUCHER

Kapselt eine auf den Elementen einer Objektstruktur auszuführende Operation als ein Objekt.



KOMPOSITUM

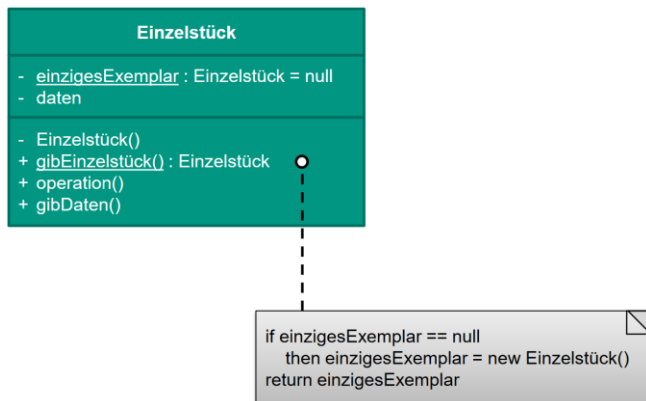
Fügt Objekte zu Baumstruktur zusammen, um Teil-Ganzes-Hierarchie zu repräsentieren und Kompositionen von Objekten einheitlich zu behandeln.



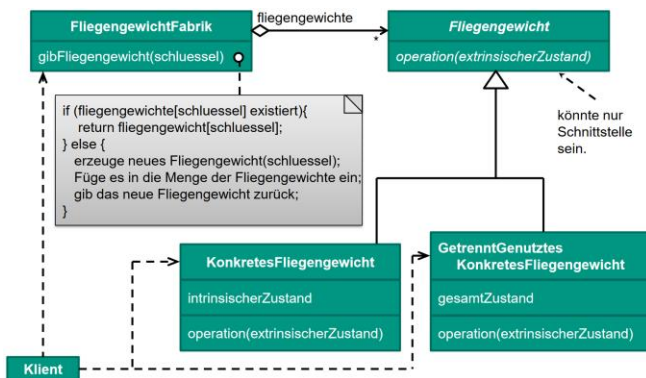
Zustandshandhabungs-Muster

Bearbeiten den Zustand von Objekten, unabhängig von deren Zweck.

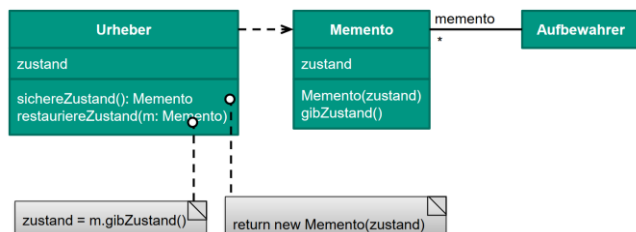
EINZELSTÜCK Sichert zu, dass eine Klasse **genau ein Exemplar** besitzt, und stelle einen globalen Zugriffspunkt darauf bereit.



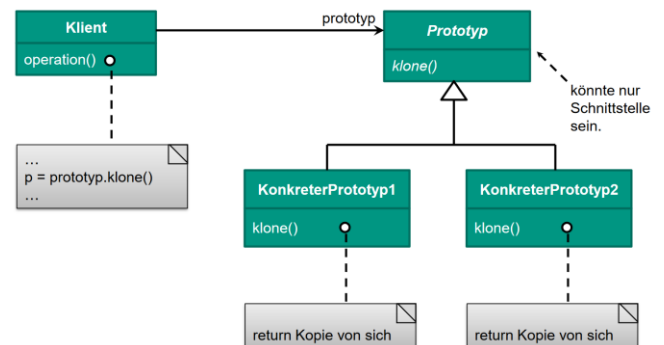
FLIEGENGEWICHT Nutzt Objekte kleinster Granularität **gemeinsam**, um große Mengen von ihnen **effizient speichern** zu können.



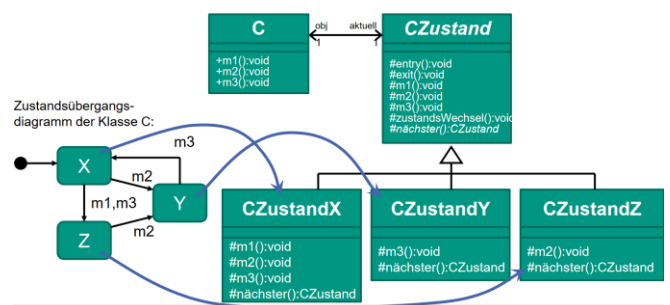
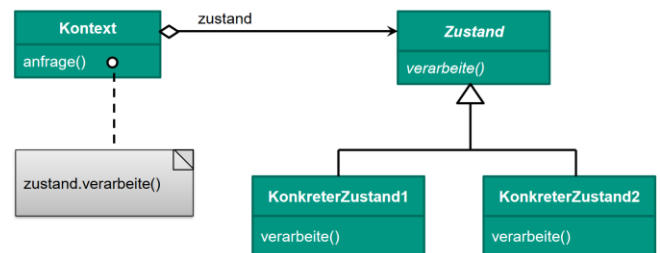
MEMENTO Erfasst und **externalisiert** den **internen Zustand** eines Objekts, ohne seine Kapselung zu verletzen, so dass das Objekt später in diesen Zustand **zurückversetzt** werden kann.



PROTOTYP Bestimmt die **Arten zu erzeugender Objekte** durch die Verwendung eines typischen Exemplars und erzeuge neue Objekte durch Kopieren dieses Prototyps.



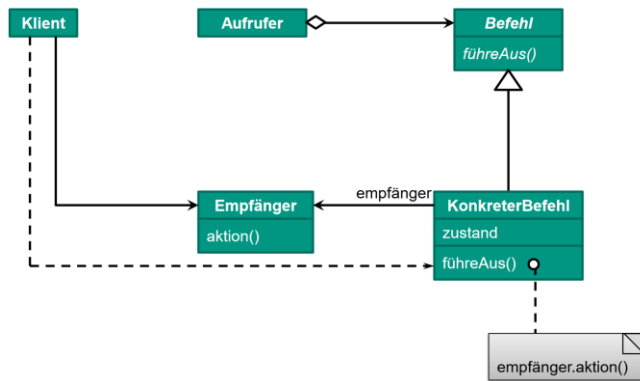
ZUSTAND Ändere das **Verhalten** des Objektes, wenn sich dessen **interner Zustand** ändert.



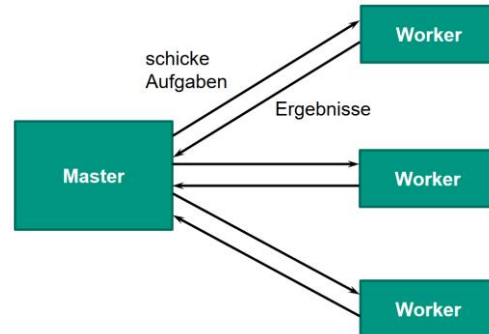
Steuerungs-Muster

Steuern den Kontrollfluss und bewirken, dass zur richtigen Zeit, die richtigen Methoden aufgerufen werden.

BEFEHL Kapselt einen **Befehl als ein Objekt**. Dies ermöglicht es, Klienten mit verschiedenen Anfragen zu parametrisieren, Operationen in eine Warteschlange zu stellen, ein Logbuch zu führen und Operationen rückgängig zu machen.



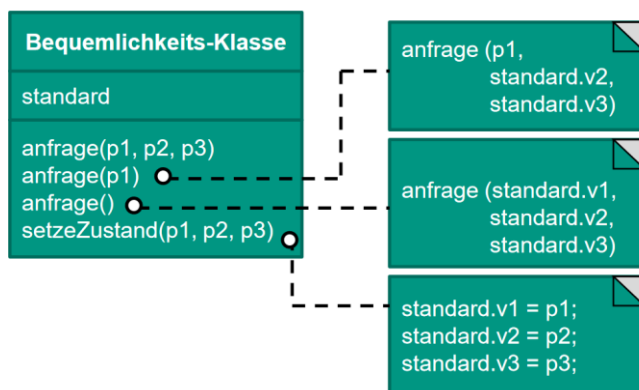
AUFTRAGGEBER/-NEHMER Auftraggeber/-nehmer bietet **fehlertolerante** und **parallele Berechnung**. Ein Auftraggeber verteilt die Arbeit an identische Arbeiter (Auftragnehmer) und berechnet das Endergebnis aus den Teilergebnissen, welche die Arbeiter zurückliefern.



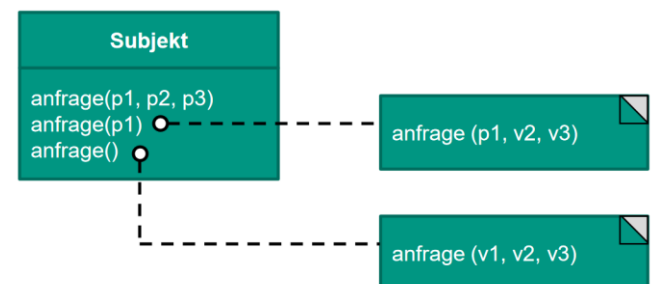
Bequemlichkeitsmuster

Sparen etwas Schreib- und Denkarbeit.

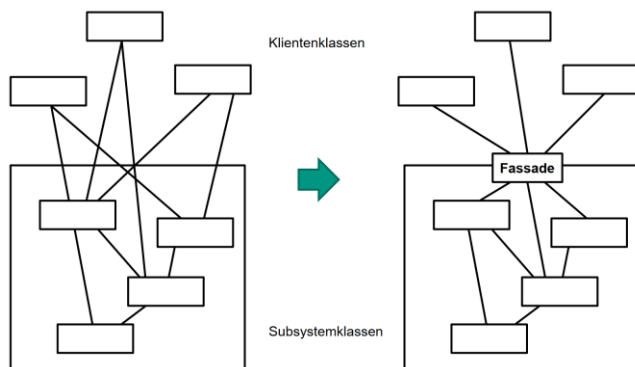
BEQUEMLICHKEITS-KLASSE Vereinfacht den **Methodenaufwurf** durch Bereithaltung der Parameter in einer **speziellen Klasse**.



BEQUEMLICHKEITS-METHODE Vereinfacht **Methodenaufwürfe** durch die Bereitstellung häufig genutzter Parameterkombinationen in **zusätzlichen** überladenen **Methoden**.



FASSADE Bietet eine **einheitliche Schnittstelle** zu einer Menge von Schnittstellen eines Subsystems.



NULL-OBJEKT Stelle einen **Stellvertreter** zur Verfügung, der die gleiche Schnittstelle bietet, aber nichts tut und verhindert so null-Abfragen im Code.

