# Abhängigkeiten zwischen Bausteinen

Abhängigkeit von Bausteinen zueinander können (u. a.) folgender Art sein:

* Verwendungsabhängigkeit: Ein Baustein benötigt zur Ausführung seiner Aufgaben Dienste eines anderen Bausteins. (siehe Delegation)
* Enthaltensabhängigkeit (siehe Komposition, siehe Aggregation): Ein Baustein oder eine Instanz eines Bausteins enthält einen anderen Baustein oder Instanzen eines anderen Bausteins.
* Enthaltensabhängigkeit kann sich auf direktes Enthaltensein beziehen oder auf das Enthaltensein von Referenzen.
* Vererbungsabhängigkeit: Ein Baustein erbt im Sinne der Objektorientierung von einem anderen.
* Zeitliche Abhängigkeit: Eine Aktion eines Bausteins ist nur in zeitlicher Abhängigkeit anderer Aktionen (durch ihn selbst oder andere Bausteine) möglich.
* Örtliche Abhängigkeit: Ein Baustein ist von der Existenz eines anderen Bausteins am gleichen Ort (etwa im gleichen Adressraum, auf dem gleichen Server, unter der gleichen IP-Adresse) abhängig.

Quelle Basiswissen für Softwarearchitekten, 4. Auflage Glossar

# Begriffe verschiedener Abhängigkeiten

## Assoziation

Modelliert Verbindungen zwischen Objekten (ganz allgemein: zwischen Bausteinen).

Jede Assoziation kann durch Kardinalitäten und (Rollen-)Namen detaillierter beschrieben werden.

## Aggregation

Sonderfall der Beziehung oder Assoziation. Zwischen den beteiligten Elementen liegt eine „Ist- Teil-von“- oder eine „Besteht-aus“-Beziehung vor.

🡺Komposition, 🡺Assoziation

## Delegation

Ein Baustein leitet eine Aufgabe an einen anderen Baustein weiter, statt die Aufgabe selbst zu erledigen.

## Komposition

Besondere Form der 🡺 Aggregation. Beim Löschen des Ganzen müssen auch alle Teile gelöscht werden. Jedes Teil kann – zu einem Zeitpunkt – nur zu einem Ganzen gehören. Es kann jedoch einem anderen Ganzen zugeordnet werden.

## Vererbung

Attribute, Operationen und Beziehungen einer Oberklasse werden an die zugehörigen Unterklassen

vererbt.

* Einfachvererbung - Es gibt höchstens eine direkte Oberklasse.
* Mehrfachvererbung Es kann mehr als eine direkte Oberklasse geben.

# Entkopplung von Abhängigkeiten

Die konsequente Anwendung von Entwurfsmuster helfen Module bzw. Bausteine so zu entwerfen, dass diese weitgehend lose gekoppelt sind.

Auf den Internetseiten der Universität in Paderborn habe ein PDF-Dokument gefunden, das erläutert welches Entwurfsmuster mit welcher Art von Entkopplung wirkt

Quelle: <https://ag-kastens.cs.uni-paderborn.de/lehre/material/oop/folien/comment301-308.2.pdf>

|  |  |
| --- | --- |
| Ziel | Entwurfsmuster |
| Objekte einer fest benannten Klasse erzeugen | * Abstract Factory
* Factory Method
 |
| Abhängigkeit von spezieller Software Plattform | * Abstract Factory
* Bridge
 |
| Abhängigkeit von speziellen Implementierungen | * Abstract Factory
* Bridge
 |
| Abhängigkeit von speziellen Algorithmen | * Strategy
 |
| Zu enge Kopplung | * Abstract Factory
* Bridge
* Observer
 |
| Funktionalität erweitern durch Vererbung | * Bridge
* Observer
* Strategy
 |

Das Problem, das Ich sehe, dass fast all diese Entwurfsmuster, nicht in den Lernzielen explizit erwähnt werden (im Gegensatz zu den anderen Entwurfsmuster)