Objektorientierte Programmierung

10 - Analysemuster

Alexander Stuckenholz

Version 2022-05-04

Inhalt

HOCHSCHULE HAMM-LIPPSTADT

Muster

Hat man einige Softwaresysteme umgesetzt, stellt man fest, dass sich gewisse Strukturen aus Klassen, Schnittstellen und Beziehungen häufig wiederholen.

- Solche Strukturen, die geeignet sind, gewisse Probleme zu lösen, nennt man **Muster**.

Muster sind uncodierte Abstraktionen, man kann sie nicht direkt wie eine Bibliothek nutzen.

- Vielmehr dienen Muster als Vorlage für die Konstruktion einer bestimmten Problemlösung.
- Sie helfen dabei, Entwürfe zu strukturieren und zu kommunizieren und geben Orientierung in komplexen Systemen.
- Muster sind kondensierte Erfahrung, der man sich bedienen kann (aber nicht muss).

Muster spielen in allen Phasen und für alle Aspekte der Softwareentwicklung eine Rolle.

- In der Analyse, im Entwurf, für kleine Details oder große Architekturfragen.
- Entsprechend unterschiedlich ist der Abstraktionsgrad von Mustern.

Analysemuster

HOCHSCHULE HAMM-LIPPSTADT

Muster, die in der Analysephase eingesetzt werden, werden als **Analysemuster** bezeichnet.

- Sie dienen dazu, fachliche Sachverhalte der geschäftlichen Praxis beispielhaft auf objektorientierte Strukturen abzubilden, siehe [1].
- Man spricht auch von den *konzeptuellen Facetten* des Systems bzw. dem *Domänenmodell*.
- Technischen Details spielen in der Analyse bekanntlich zunächst eine untergeordnete Rolle.

Analysemuster befassen sich oft mit Konzepten, die in Unternehmen anzutreffen sind, z.B.:

- Organisationsstrukturen, Mitarbeiter, Kunden, Konten, Produkte, Bestellungen, Mengen- und Zeitangaben, Verträge, usw.
- Es ist alles andere als trivial, all diese Dinge sinnvoll und präzise auf Klassen und Beziehungen abzubilden.

Im Folgenden werden einige Szenarien und dafür hilfreiche Analysemuster vorgestellt.

Gehalt

HOCHSCHULE HAMM-LIPPSTADT

Die Mitarbeiter eines Unternehmens werden in einem System durch die gleichnamige Klasse abgebildet.

- Die Klasse besitzt die Eigenschaft Gehalt.
- Diese Eigenschaft kann zu jedem Zeitpunkt nur genau einen Wert aufnehmen.

Wird das Gehalt durch einen neuen Wert überschrieben, geht der alte Wert verloren.

- Es ist dann nicht nachzuvollziehen, welche unterschiedlichen Gehälter ein Mitarbeiter über die Zeit hinweg erhalten hat.

Nun soll aber der zeitliche Verlauf des Gehalts abgebildet werden können.

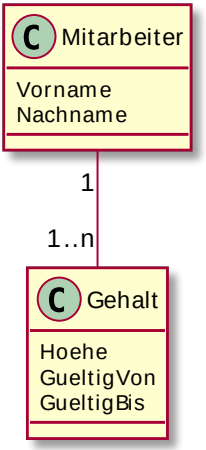
Historie

Dieses Problem wird durch das Analysemuster **Historie** adressiert.

- Die zu historisierende Eigenschaft wird als eigene Klasse extrahiert.
- In unserem Beispiel ist dies die Klasse *Gehalt*.
- Sie erhält die Höhe des Gehalts und den Gültigkeitszeitraum als Eigenschaft.

Über eine Assoziation können dem Mitarbeiter dann mehrere Objekte der Klasse *Gehalt* zugeordnet werden.

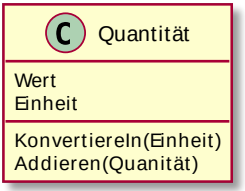
- Die Veränderung der Höhe des Gehalts kann dann nachvollzogen werden.
- Die Gültigkeitszeiträume der Gehaltsobjekte dürfen sich nicht überschneiden.



Quantität

Attribute, wie die Höhe des Gehalts, werden häufig über einen elementaren Datentyp (z.B. `double`) abgebildet.

- Eine solche Abbildung ist aber sehr unpräzise.
- Möglicherweise hat das Unternehmen Standorte in unterschiedlichen Ländern mit unterschiedlichen Währungen.



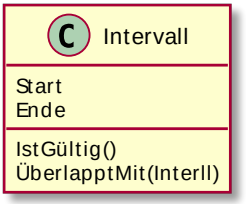
Bei solchen Angaben sollte daher auf die Verwendung eines einfachen Datentyps verzichtet werden, siehe [1].

- Stattdessen sollte ein neuer Typ **Quantität** eingeführt werden
- Ein Objekt dieser Klasse kann dann die Einheit des Werts (einfach oder zusammengesetzt) definieren.
- Auch können Transformationsmethoden und Rechenoperationen korrekt abgebildet werden.

Intervall

Um im Analysemuster Historie die Eigenschaft einer Klasse zu historisieren, wird eine neue Klasse über eine Assoziation verbunden.

- Diese neue Klasse beinhaltet einen Gültigkeitszeitraum, der angibt, in welchem Intervall das Attribut den entsprechenden Wert annimmt.
- Dieser Gültigkeitszeitraum sollte bestenfalls ebenfalls als eigene Klasse abgebildet werden.
- So können entsprechende Prüfmethoden implementiert werden.



In unserem Beispiel handelt es sich bei der Unter- und Obergrenze um Datumswerte.

- Aber auch andere, numerische Werte sind möglich und denkbar.

Bestellung

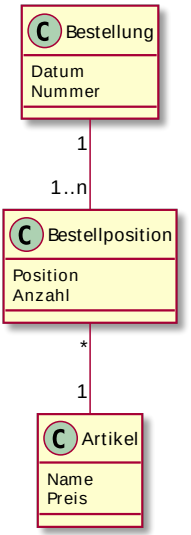
HOCHSCHULE HAMM-LIPPSTADT

In einem Shop-System müssen Bestellungen abgebildet werden.

- Einer Bestellung sind viele Artikel zugeordnet.
- Bestellung und Artikel besitzen eigene Attribute.
- Aber auch die Zuordnung der Artikel zu der Bestellung besitzt Eigenschaften, z.B. die Bestellmenge oder die Position des Artikels.

Dieses Szenario wird durch das Analysemuster **Liste** bzw. **Koordinator** adressiert.

- Neben der Klasse Bestellung und Artikel wird eine eigene **Assoziationsklasse** *Bestellposition* eingeführt.
- Diese verbindet einen Artikel mit der Bestellung.
- Zudem kann in der Bestellposition die Menge usw. abgebildet werden.



Produktkategorie

HOCHSCHULE HAMM-LIPPSTADT

In einem Online-Bestellsystem sollen Produkte in Kategorien organisiert werden können.

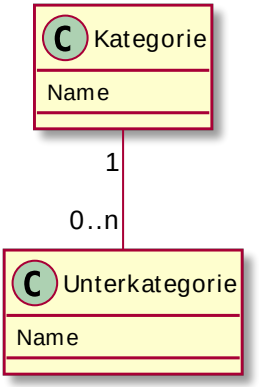
- Eine Produktkategorie besitzt dabei einen Namen.
- Eine Produktkategorie kann Unterkategorien besitzen.

Eine naive Modellierung bildet eine Kategorie und eine Unterkategorie jeweils als eigene Klasse ab.

- Beide Klassen werden über eine Assoziation miteinander verbunden.

Dies ist aber keine gute Modellierung.

- In der Analysephase bilden Klassen fachliche Konzepte ab.
- Es existiert aber kein konzeptueller Unterschied zwischen einer Kategorie und einer Unterkategorie.



Reflexive Assoziation

HOCHSCHULE HAMM-LIPPSTADT

Eine Assoziation darf auch eine Klasse mit sich selbst verbinden.

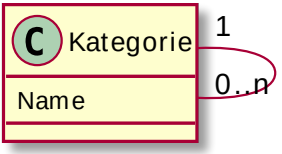
- Eine derartige Assoziation wird **reflexive Assoziation** genannt.

So kann das vorherige Szenario viel geschickter abgebildet werden.

- Einem Objekt der Klasse Kategorie können beliebig viele andere Kategorie-Objekte als Unterkategorien zugeordnet werden.

Eine solche Modellierung kann alle möglichen Bäume bzw. Graphen abbilden.

- z.B. Vater/Mutter/Sohn, Verzeichnisstrukturen, usw.



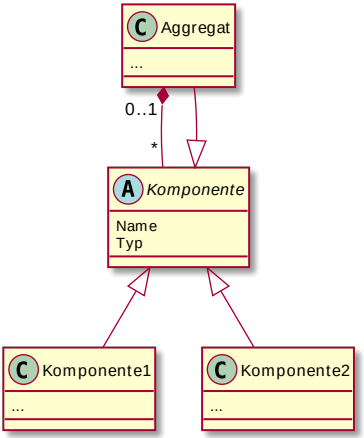
Kompositum

Das Analysemuster **Kompositum** nutzt ebenfalls die reflexive Assoziation als Stilmittel.

- Es wird immer dann eingesetzt, wenn ein größeres Ganzes, z.B. eine Maschine oder System, aus Einzelteilen besteht und als Modell abgebildet werden muss.

Die abstrakte Basisklasse abstrahiert dabei von konkreten Komponenten.

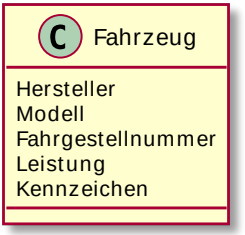
- Eine Aggregat besteht aus einer beliebigen Menge von Komponenten.
- Da das Aggregat aber selber eine Komponente ist, kann es auch aus weiteren Aggregaten bestehen.



Autovermietung

Eine Autovermietung hat sehr viele Fahrzeuge im Bestand.

- Die Klasse Fahrzeug besitzt Eigenschaften, wie den Herstellernamen, das Modell, die Fahrgestellnummer und das Kennzeichen.



Diese Modellierung zeigt aber einige Nachteile:

- Die Werte mancher Eigenschaften haben eine hohe Redundanz, z.B. wiederholt sich der Hersteller sehr oft.
- Es kann kein Fahrzeugmodell eingeführt werden, wenn nicht auch schon ein konkretes Fahrzeug dazu im Bestand ist.

Offenbar sollte die Klasse in mehrere Konzepte aufgeteilt werden.

Exemplartyp

Das Analysemuster **Exemplartyp** adressiert diese Situation.

- Die Eigenschaften werden dabei auf zwei Klassen verteilt.

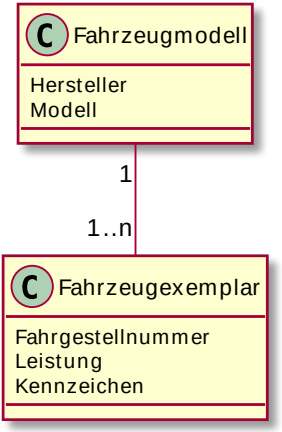
Die Klasse *Fahrzeugmodell* sammelt alle modellspezifischen Eigenschaften.

- Dies ist z.B. der Hersteller und das Modell.

Eine weitere Klasse *Fahrzeugexemplar* repräsentiert die konkreten Fahrzeuge im Bestand.

- Diese trägt Eigenschaften, wie die Fahrgestellnummer und die Leistung.

Dem Fahrzeugmodell können dann viele konkrete Fahrzeugexemplare zugeordnet werden.



Vorrat

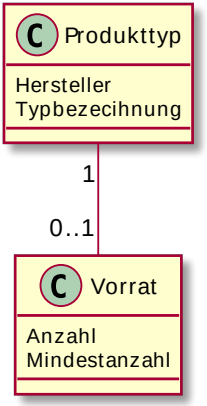
Das Analysemuster Exemplartyp sollte immer dann angewandt werden, wenn die einzelnen Exemplare voneinander unterscheidbar sind.

- z.B. Fahrzeuge, Bücher, ...

Bei einem anonymen **Vorrat** können die Exemplare aber nicht voneinander unterschieden werden.

- z.B. Schrauben, Betriebsstoffe, ...

Der Vorrat wird dann als ein einzelnes Objekt abgebildet.



Rollen

HOCHSCHULE HAMM-LIPPSTADT

In einer Beziehung können Objekte gegenüber anderen Objekten **Rollen** einnehmen.

- Eine Person ist gegenüber einem Ort der Bewohner.
- Der Ort ist gegenüber der Person der Wohnort.

Eine Assoziation zwischen Klassen bildet dann die Bedeutung einer Rolle ab.

- Der Rollenname kann zum besseren Verständnis auch an die Enden der Assoziation geschrieben werden.

Objekte einer Klasse können auch mehrere, unterschiedliche Rollen gegenüber einem anderen Objekt einnehmen.

- Eine Person kann in einer Fußballmannschaft der Trainer oder der Torwart sein.

Das Konzept der Rollen ist ebenfalls ein bekanntes Analysemuster.

Beispiel

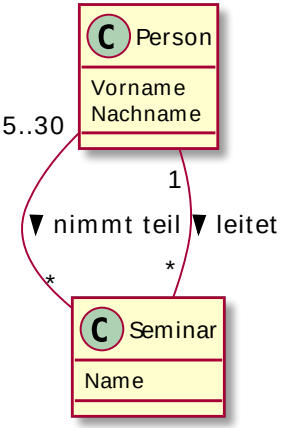
Eine Person kann für ein Seminar der Teilnehmer oder der Dozent sein.

- Beide Rollen werden als jeweils eigene Assoziation zwischen den Klassen Person und Seminar abgebildet.

Der Dozent sollte dabei nicht gleichzeitig auch Teilnehmer sein.

- Diese Bedingung lässt sich mit dem Klassendiagramm allein nicht darstellen.
- Eine solche Bedingung kann aber mithilfe der Object Constraint Language (OCL) ausgedrückt werden, siehe [2].

Die Art und Anzahl der Rollen muss bei dieser Modellierung über die Zeit stabil bleiben.



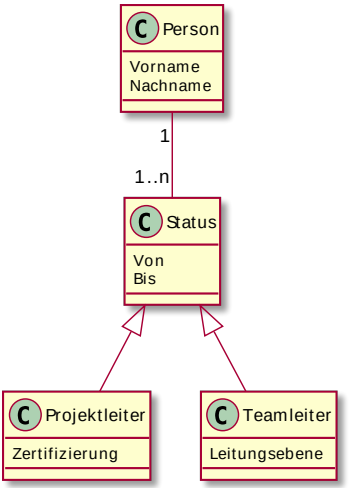
Wechselnde Rollen

Sind die Rollen nicht starr und können sich potenziell ändern, wird das Analysemuster der **wechselnden Rollen** angewandt.

- Im Beispiel kann der Person, ähnlich wie in der Historie, eine zeitlich begrenzte Rolle zugewiesen werden.

Die konkreten Rollen, Team- oder Projektleiter, werden dann von einer gemeinsamen (abstrakten) Basisklasse abgeleitet.

- Auch hier sollten sich die Gültigkeitszeiträume nicht überschneiden.



Kombination von Mustern

HOCHSCHULE HAMM-LIPPSTADT

Muster sind keine starren Konstrukte, sondern Hilfsmittel bei der Umsetzung von Softwaresystemen.

- Muster helfen dabei, Probleme zu erkennen und diese mit bewährten Strategien zu lösen.
- Alle Muster, egal ob Analyse- oder Entwurfsmuster, können beliebig miteinander kombiniert und an die eigenen Bedürfnisse angepasst werden.

Zum Beispiel kann mit der Historie jede Eigenschaft einer Klasse historisiert werden.

- Dabei kann es sich z.B. auch um die Rolle eines Mitarbeiters (Rollen) oder die Zuordnung einer Komponente eines Aggregats handeln.
- Die Quantität oder das Intervall kann auch leicht mit einer Koordinator-Klasse verbunden werden.
- Im Prinzip gibt es keine Grenzen bei der Kombination der Muster.

Zusammenfassung

Wir haben heute gelernt, ...

- was Muster im Allgemeinen sind und welchen Verwendungszweck sie haben.
- dass Analysemuster dabei helfen, die Fachlichkeit eines Systems in einem Modell abzubilden.
- welche Probleme die Analysemuster Historie, Quantität, Intervall und Koordinator lösen.
- wozu man reflexive Assoziationen nutzen kann und wie dieses Strukturmittel im Muster Kompositum angewandt wird.
- wie die Analysemuster Exemplartyp und Vorrat angewandt werden.
- wie man die Analysemuster Rollen und wechselnde Rollen verwendet.

Aufgabe 1

In den Aufgaben des vorherigen Abschnitts sollten einige Informationsmodelle in Form von UML-Klassendiagrammen erstellt werden.

- Untersuchen Sie die Modelle auf die Anwendbarkeit von Analysemustern.
- Wo ist es sinnvoll, Analysemuster anzuwenden?

Aufgabe 2

Überlegen Sie, wie das Informationsmodell eines Streaminganbieters wie Spotify oder Apple Music aussehen könnte.

- Welche Muster finden hier Verwendung?

Quellen I

[1] Martin Fowler. *Analysis Patterns, Reusable Object Models*. 1. Aufl. Menlo Park, Calif: Addison-Wesley Longman, Amsterdam, 1996. 357 S. ISBN: 978-0-201-89542-1.

[2] Jos Warmer und Anneke Kleppe. *The Object Constraint Language: Getting Your Models Ready for MDA*. 2. Aufl. Boston, Mass. Munich: Addison-Wesley Professional, 27. Aug. 2003. 236 S. ISBN: 978-0-321-17936-4.