



Hinweis: Diese Lernnotiz enthält einen sehr sinnvollen Vorschlag um den Lehrstoff der 6. Woche der Veranstaltung (am 7.4) zu erlernen. Er ist gegliedert in die generellen Ziele und die Arbeitsschritte. Es ist notwendig, dass Sie die in dieser Lernnotiz genannten Videos bis zum Ende der offiziellen Vorlesungszeit (Mo 11:30) durchgearbeitet haben. Zur Vorlesungszeit besteht die Möglichkeit in Zoom Fragen zu stellen und weitergehende Themen zu diskutieren.

<https://hs-osnabrueck.zoom.us/my/kleuker>

Einzelne Termine können kurzfristig per E-Mail vereinbart werden.

Ziele

- Vertiefte Fähigkeit zum Lesen und zur Erstellung eigener Klassendiagramme mit der UML-Standardnotation.
- Vertiefte Fähigkeit zum Lesen und zur Erstellung eigener Sequenzdiagramme zur Validierung von Klassendiagrammen mit der UML-Standardnotation.
- Generierungsmöglichkeiten von UML-Werkzeugen kennen.

Arbeitsschritte

- *Laden Sie sich die folgenden Videos zuerst herunter, wenn Sie die HS-Plattform nutzen und schauen Sie sich diese an. Es ist sinnvoll die Folien danach nochmals durchzugehen.*

Folien 166 – 188: Ableitung von Klassendiagrammen aus Anforderungen
<http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADGrobdesign3.mp4> (39:40),
auch <https://youtu.be/c2t1ljV1t4>

Folien 189 – 198: Genauere Betrachtung von Sequenzdiagrammen
<http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADGrobdesign4.mp4> (18:28),
auch <https://youtu.be/ZyjGtpJsdV4>

Folien 199 – 203: GUI-Anbindung und Anforderungstracing
<http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADGrobdesign5.mp4> (12:06),
auch <https://youtu.be/FQizDJqX69A>

Folien 204 – 211: Einstieg in die Code-Generierung
<http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADUMLzuCode1.mp4> (15:54),
auch <https://youtu.be/MAKu2cshkoo>

- Lesen Sie die Seiten 109 - 133 (also auch den Anfang von Kap. 6) im Buch, einige Teile sollten Ihnen aus der letzten Veranstaltung bekannt vorkommen. Dokumentieren Sie offene Fragen und schicken Sie sie an den Dozenten.
- Die Folien 166-173 bauen ein weiteres Beispiel mit Klassen auf, bei der viele Objekte der einen Klasse durch ein Objekt einer übergeordneten Klasse verwaltet werden. Wieder werden Methodenaufrufe delegiert und Teile der Parameter zur Identifikation der darunter liegenden Objekte genutzt. Diese Verknüpfung wird in den Klassendiagrammen mit den gelben Kästen angedeutet. Von unten nach oben werden Prüfungsergebnisse Klausuren, diese dann Klausurlisten eines Semesters eines Studiengangs und diese dann einer Klausurlistenverwaltung zugeordnet. In einer kleinen Variante wäre es sinnvoll, wenn die Klausurlistenverwaltung auf die Studierendenverwaltung zugreifen könnte, damit in den letzten beiden Methoden kein Studierend-Objekt sondern eine Matrikelnummer übergeben werden könnte.



Die Folie 174 erinnert an die Fallstudie, die bisher in der Vorlesung und im Buch genauer betrachtet wird. Es geht um eine Projektplanung, in die der Projektplan, bearbeitende Person und der erreichte Arbeitsstand eingetragen werden. Projekte werden gerne hierarchisch mit einem Projektstrukturplan geplant. Dabei werden Hauptaufgaben in Teilaufgaben soweit heruntergebrochen, dass ihr Aufwand kalkulierbar wird und so einzelnen mitarbeitenden Personen zugeordnet werden kann. Es entsteht so eine klassische Baumstruktur, bei der die Blätter, Knoten die keine weitere Verzweigung haben, Aufgaben oder Arbeitspakete und die Zwischenknoten Projekte oder genauer Teilprojekte genannt werden. Bei größeren inkrementellen Projekten ist es dabei nicht unüblich nur aktuell anstehende Zweige bis ins Detail zu planen. Wenn der Aufwand für Aufgaben kalkuliert ist, kann der Aufwand für Teilprojekte einfach als Summenbildung kalkuliert werden. Die genauen Anforderungen für die Software stehen im Buch.

- Der ganz naive Weg zur Klassenfindung, der hier für Unerfahrene zur Sicherheit vorgestellt wird, ist iterativ in zumindest zwei Schritten. Zunächst werden die Klassen mit ihren Objekt- und Klassenvariablen gesucht, danach die Methoden. Bei Anforderungen nach der Rupp-Schablone kann diese Suche sehr systematisch erfolgen, da hier die Satzstruktur ausgenutzt werden kann. Nomen stehen oft für Klassen, Adjektive für Eigenschaften und damit Objektvariablen und Verben oft für Tätigkeiten eines Objekts, also dessen Methoden. Die Folien 175-185 fassen einen solchen Weg zusammen. Zunächst werden die Anforderungen zusammen mit dem Glossar angesehen und versucht eine Klassenstruktur abzuleiten. Dies ist ein hochdynamischer Prozess, der in Folien nur schwer abgebildet werden kann, da hier der Fokus auf den Ergebnissen liegt. Folie 177 zeigt ein erstes Diagramm nur mit Objektvariablen. Assoziationen sind noch nicht gerichtet, haben aber ihre Multiplizitäten. Typisch für eine Baumstruktur ist der rekursive oder reflexive Aufbau, bei dem ein Projekt durch beliebige viele andere (Teil-Projekte) verfeinert wird. Die mit einem Schrägstrich markierten Objektvariablen sind bei einer zweiten Analyse als solche erkannt worden, die aus anderen Modellanteilen vollständig berechnet werden können. Diese können in einem Analysemodell stehen, da sie aber nicht implementiert werden sollen, wird meist auf sie verzichtet.

Folie 179 skizziert einen zentralen Ansatz, der systematischer in Folie 203 wieder aufgegriffen wird. Für jedes neue Modellelement (Artefakt), also Klasse, Variable und Methode, wird festgehalten warum es existiert, also aus welcher Anforderung es abgeleitet wurde. Nur so ist es später prüfbar, ob alle Anforderungen umgesetzt wurden und ob Modellelemente aus „dem Bauchgefühl“ ohne konkrete Anforderungen irrtümlich entstanden sind. Gerade in langfristig durchgeführten Projekten, z. B. zusammen mit einer Hardware-Entwicklung wie es z. B. im Landmaschinenbau der Fall ist, werden diese Informationen sehr sorgfältig mit Werkzeugen verwaltet.

Die Folien 202-203 geben einen detaillierten Überblick, was für Tracing-Informationen, also Spurenverfolgung, es sinnvollerweise in der Anforderungsverfolgung geben soll. Diese Art der Zuordnung ist bei jedem Schritt sinnvoll, bei dem Informationen aus einer Modellart aus einer anderen Modellart konstruiert werden. Da hierbei ein Modellelement mit mehreren anderen verknüpft werden kann, sind weitere Verfeinerungen der Kanteninformationen, wie z. B. zu wieviel Prozent eine Umsetzung passiert, sinnvoll. In Folie 203 ist erkennbar, dass Anf 4 nicht umgesetzt und dass Anf 5 anscheinend frei erfunden wurde. Die Nachverfolgung von Anforderungen ist später ein wichtiger Teil der Validierung ob und wo Anforderungen umgesetzt wurden.



Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing-Wiss. und Informatik
- Software-Entwicklung -

Objektorientierte Analyse und Design

Sommersemester 2025

6. Lernnotiz

- Lesen Sie das zur Vorlesung gehörende Fragen-Und-Antworten-Dokument, das meist kurz nach der Vorlesung auf der Veranstaltungsseite in der Nähe dieser Lernnotiz steht.
- Bearbeiten Sie das Quiz unter http://kleuker.iui.hs-osnabrueck.de/quiz/ood06_89245.html und merken Sie sich die oben angegebenen Lösungsbuchstaben.
- Bearbeiten Sie Aufgabenblatt 6. Denken Sie daran, dass ich für Fragen meist kurzfristig erreichbar bin.
- Da die nächste Vorlesung leider nicht am 28.4 stattfindet, bezieht sich das folgende Aufgabenblatt, genauer Blatt 6b, generell auf OOAD. Denken Sie daran, dass ich für Fragen meist kurzfristig erreichbar bin.
- Prüfen Sie, ob Sie die angegebenen Lernziele erreicht haben.