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

Objektorientierte Analyse und Design

Sommersemester 2022
5. Lernnotiz

Hinweis: Diese Lernnotiz enthält einen sehr sinnvollen Vorschlag um den Lehrstoff der 5. Woche der Veranstaltung 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 (Mi 13:45) 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.

Anmerkung: Die hier besprochenen Folien sind nicht in dieser Form im Buch enthalten, da das Buch eher für in der Software-Entwicklung unerfahrenere Personen ist und Sie bereits drei Semester intensive objektorientierte Programmierausbildung hinter sich haben. Im Buch werden deshalb getrennt zunächst die Klassenmodellierung, dann Sequenzdiagramme und dann der zugehörige Code besprochen. Bei erfahreneren Personen bietet es sich aber an, zumindest kurz beim bekannten Programmcode zu starten und dann (zurück) zur Klassenmodellierung überzugehen. Weiterhin haben sie die Abläufe bei der Nutzung einer Sammlung (Collection, z. B. List) im Hinterkopf, so dass es sinnvoll ist, Klassendiagramme unmittelbar zusammen mit Sequenzdiagrammen zu betrachten. Die weiter unten angegebenen textuellen Zusammenfassungen sollen nur etwaige Unklarheiten klären, falls die Formulierungen in den Videos nicht eindeutig sind.

Ziele

- Fähigkeit zum Lesen und zur Erstellung eigener Klassendiagramme mit der UML-Standardnotation.
- Fähigkeit zum Lesen und zur Erstellung eigener Sequenzdiagramme zur Validierung von Klassendiagrammen mit der UML-Standardnotation.
- Fähigkeit zur Anwendung erster typischer Design-Schritte bei klassischen OO-Modellierungsherausforderungen (Controller, Entity).

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 130 – 141: Einfache Klassen und Sequenzdiagramme in der UML
  http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADGrobdesign1.mp4 (38:54),
  auch https://youtu.be/w2k1ROtIBUM

  Folien 142 – 165: Weitere Klassenmodellierung in der UML
  http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADGrobdesign2.mp4 (44:17),
  auch https://youtu.be/lnPfIqENmVs

- Lesen Sie die Seiten 93 - 109 im Buch.
  Dokumentieren Sie offene Fragen und schicken Sie sie an den Dozenten.

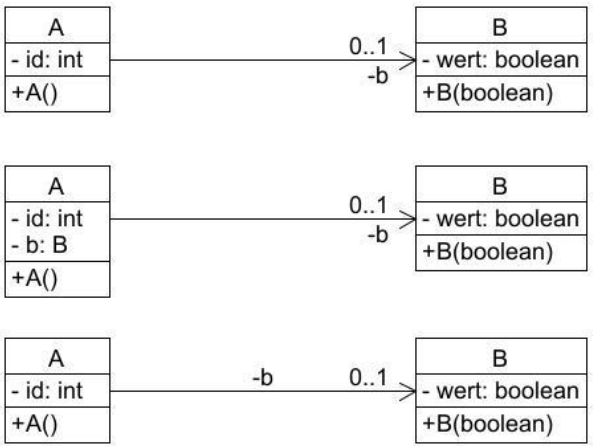
- Zum Einstieg in die Klassenmodellierung sollten Sie von den Folien 135-137 zunächst nur den Code auf der linken Seite lesen, um zu erkennen, dass es eine sehr gewöhnliche Java-Klasse ist. Generell wird immer versucht POJOs (Plain Old Java Objects) zu nutzen, die einen parameterlosen Konstruktor sowie get- und set-Methoden für alle Objektvariablen haben. Nun sollte die Klasse in der UML-Form auf den gleichen Folien angesehen werden. Generell sind Stereotypen eine einfache und sehr mächtige Form die UML für spezielle Bereiche zu erweitern. Es gibt einige Stereotypen, die vorgegeben sind, wie es von den Use Cases bereits bekannt ist. In den genutzten Klassendiagrammen sind zunächst nur <<interface>> für Interfaces und

<<abstract>> für irgendwas abstrakt markiertes fest vergeben. Stereotypen können dabei bei allen Elementen, also auch Variablen- und Methodennamen, Parametern oder Typen stehen. Da die UML OO-sprachenunabhängig ist, gibt es leicht andere Notationen, z. B., dass Variablen immer in der Form <name>:<Typ> dargestellt werden. Einfache Objektvariablen, auch unsauber Instanzvariablen oder Attribute genannt, können optionale Startwerte bekommen, Klassenvariablen, unsauber static-Variablen genannt, werden unterstrichen.

- Folie 138 macht ein wichtiges Konzept der UML deutlich, dass sie bewusst zur Modellierung eingesetzt wird und viele Dinge deshalb optional sind. Dadurch ist der Fokus nicht sofort auf die Implementierung, sondern auf die Findung eines ersten Modells ausgerichtet. Das erste Ziel ist es immer die benötigten Klassen mit Objektvariablen zu identifizieren, ähnlich wie es von der Erstellung eines Datenmodells für Datenbanken der Fall ist.

- Folie 152 ist ein Einschub zum Selbststudium, woran man erkennen kann, dass Leute sich die Programmierung selbst beigebracht haben oder nicht richtig zugehört haben, was bei der Entwicklung in größeren Teams immer zu Problemen führt.

- Die drei Klassendiagramme auf der rechten Seite zeigen eine korrekte Modellierung und zwei Fehler. Zu modellieren ist, dass eine Klasse A eine Objektvariable b vom Typ B hat, die nicht unbedingt einen Wert enthalten muss. Der Fehler in der Mitte ist, dass die Variable b doppelt auftaucht. Generell gilt, dass Abhängigkeiten zu anderen Klassen graphisch zu modellieren sind und niemals doppelt auftreten. Im unteren Diagramm steht der Name der Objektvariablen in der Mitte. Formal kann er so weder der linken noch der rechten Seite zugeordnet werden. In der Mitte kann ein informeller Name der Relation stehen, der das Verständnis erleichtern kann, später aber nicht in den Programmcode einfließt. Da Sie als Personen mit einiger Erfahrung in der Objektorientierung angesehen werden, fallen in der Veranstaltung solche Konzepte weg und wir nähern uns schnell einem Implementierungsmodell.



- 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 weiter Aufgabenblatt 4. Denken Sie daran, dass ich für Fragen meist kurzfristig erreichbar bin.

- Prüfen Sie, ob Sie die angegebenen Lernziele erreicht haben.