

Hinweis: Diese Lernnotiz enthält einen sehr sinnvollen Vorschlag um den Lehrstoff der 8. Woche der Veranstaltung (am 6.5) 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

- Verstehen warum Pattern zentrale Grundlage des ingenieurmäßigen Software-Engineerings zur Entwicklung wartbarer komplexer Software-Systeme sind.
- Die elementare Bedeutung von Interfaces und dynamischer Polymorphie für die Objektorientierung verstehen und erklären können.
- Die Grundidee von Model-View-Controller, dessen Varianten und Umsetzungen kennen und selbst einsetzen können.
- Das Observer-Observable-Pattern, dessen Varianten und Umsetzungen im Detail kennen und selbst einsetzen können.

### 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 248 – 257: Einführung in die Design-Pattern-Thematik

[http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADPattern1\\_1.mp4](http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADPattern1_1.mp4) (25:56),  
auch <https://youtu.be/5qKkidKjDw>

Folien 258 – 275: Model View Controller

[http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADPattern1\\_2.mp4](http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADPattern1_2.mp4) (37:09),  
auch <https://youtu.be/-MRm3WtiXAI>

Folien 276 – 283: Observer Observable

[http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADPattern1\\_3.mp4](http://kleuker.iui.hs-osnabrueck.de/Videos/OOAD/OOADPattern1_3.mp4) (20:56),  
auch <https://youtu.be/dm4doc4e57A>

- Lesen Sie die Seiten 179-196 (Kap. 7 ist noch nicht ganz so wichtig, wird am Ende teilweise angeschaut) im Buch. Dokumentieren Sie offene Fragen und schicken Sie sie an den Dozenten.
- Design Pattern sind ein Grund, warum Informatik auch als Ingenieurwissenschaft angesehen werden kann. Es gibt etablierte, garantiert funktionierende Vorgehensweisen, die zentrale wiederkehrende Aufgabenstellungen lösen. Es sind Variationen erlaubt, die weiterhin die Funktionsfähigkeit garantieren. Solche Regeln finden sich in anderen Ingenieurwissenschaften wieder, wird z. B. ein Material in einer bestimmten Art verarbeitet, ist die Stabilität bis zu einem bekannten Grenzwert gesichert und muss nicht jedes Mal neu geprüft werden.  
Im Alltag der Entwicklung kennt jede software-entwickelnde Person (nicht hackende oder autodidaktisch erlernte programmierende Person) wesentliche Design Pattern, so dass sie die Diskussion des richtigen Klassendesigns wesentlich verkürzen.  
Folie 251 macht deutlich, dass software-entwickelnde Personen erst „gehen“ lernen müssen, bevor sie mit Design Pattern das „rennen“ lernen. Konkret bedeutet dies, dass elementare Rahmenbedingungen eingehalten werden. Dies beginnt mit den Coding

Guidelines (für unsere Veranstaltung, s. <http://kleuker.iui.hs-osnabrueck.de/querschnittlich/CodingGuidelinesUndGlossar.pdf>), die zur Programmiersprache gehören, geht über das Vermeiden von Goldrandlösungen, also unnötigen Verallgemeinerungen, da die zu entwickelnde Lösung eventuell später mal vielleicht auch in einer anderen Version benötigt wird, bis zur Nichtanwendung von Objektorientierung in objektorientierten Projekten.

Wer sich das Swing-Beispiel genauer angeschaut hat, erkennt, dass hier ein zu MVC verwandter Ansatz genutzt wird. Der Controller meldet sich bei seinen beiden Knöpfen indirekt an, dazu übergibt er ein Objekt, das das Interface ActionListener realisiert und so in die Liste der Listener des Knopfes eingetragen wird. Nach dem Klicken des Knopfes wird bei jedem angemeldeten Objekt die Methode actionPerformed() aufgerufen, in der dann eine Methode des Controllers aufgerufen wird. In einer Variante hätte der Controller auch selbst das Interface realisieren und so direkt informiert werden können.

MVC-Varianten gibt es in vielen Oberflächentechnologien, von denen Sie einen Teil gerne genauer in der Hausarbeit analysieren dürfen. Dies kann Swing sein, relativ alt, aber bis 2027 mindestens weiterhin Teil des JDK, das kann JavaFX/OpenJFX sein oder ... .

- 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/oad08\\_89265.html](http://kleuker.iui.hs-osnabrueck.de/quiz/oad08_89265.html) und merken Sie sich die oben angegebenen Lösungsbuchstaben.
- Bearbeiten Sie Aufgabenblatt 8. Denken Sie daran, dass ich für Fragen meist kurzfristig erreichbar bin. Laden Sie sich zur Aufgabe das Video <http://kleuker.iui.hs-osnabrueck.de/Videos/AufgabeObsobs.mp4> herunter und schauen Sie es an.
- Prüfen Sie, ob Sie die angegebenen Lernziele erreicht haben.