

## Curriculum

# Modul Design Patterns

Modulkürzel:	depa
Modulname:	Design Patterns
Zugehörige Modulbeschreibung:	<a href="https://www.fhnw.ch/de/studium/module/9114931">https://www.fhnw.ch/de/studium/module/9114931</a>
Verfasser:	Dominik Gruntz Wolfgang Weck
Datum:	17. 09. 2018

# 1. Funktion im Rahmen der Gesamtausbildung

## 1.1. Leitidee

In diesem Modul wird aufgezeigt, wie grundsätzliche Design-Probleme gelöst werden. Ein Schwergewicht bilden dabei Entwurfsmuster für den System-Design. Die Studierenden sollen lernen, die wichtigsten Entwurfsmuster anzuwenden und die Struktur eines Systems mit Entwurfsmustern zu beschreiben. Behandelt wird eine Auswahl typischer Entwurfsmuster, deren Einsatz in heutigen Software-Projekten eine Selbstverständlichkeit geworden ist.

Zur Beschreibung der Muster wird UML eingesetzt, aber UML wird nur soweit thematisiert als es für das Verstehen der Entwurfsmuster nötig ist (Klassen-, Objekt- und Sequenzdiagramme).

## 1.2. Voraussetzungen / Vorkenntnisse

Dieser Kurs kann auch von Studierenden belegt werden, die nicht im SG Informatik eingeschrieben sind. Voraussetzung ist jedoch, dass grundlegende Kenntnisse in der objektorientierten Programmierung mit Java vorhanden sind, insbesondere Grundkenntnisse der Java-Bibliotheken für Collections und GUIs.

# 2. Unterrichtsorganisation

## 2.1 Lern- und Arbeitsformen

Die Studierenden wenden ihr Wissen in einer konkreten Fallstudie an. Ausgehend von einem gegebenen Framework wird eine Anwendung realisiert und mit zusätzlichen Funktionen erweitert.

Als Beispiel wird in der Fallstudie ein Grafikeditor entwickelt bzw. erweitert, aber es könnte auch ein anderes Rahmenwerk verwendet werden, z.B. ein Framework für verteilte Spiele. Auch an anderen Hochschulen werden solche Ansätze durchgeführt wie z.B. das Framework Salespoint der TU Dresden [S. Zschaler, B. Demuth, L. Schmitz: *SalesPoint: A Java framework for teaching object-oriented software development*, in: Science of Computer Programming 79 (2014) 189-203].

In jeder Woche wird ein neues Problem aus dem Grafikeditor aufgegriffen, Lösungen diskutiert, ein Entwurfsmuster eingeführt und danach mit Beispielen aus den Java-Bibliotheken illustriert. Die Studierenden müssen dann ihren Grafikeditor um die besprochene Funktion erweitern.

Im Stundenplan sind drei Wochenlektionen für diesen Kurs reserviert. Im Schnitt werden zwei davon in Form von Kontaktlektionen verwendet und eine, in welcher die Studierenden selbstständig an Lernaufgaben oder am Grafikeditor arbeiten. Diese beiden Unterrichtsformen wechseln sich ab, d.h. wir erwarten die Präsenz der Studierenden während drei Lektionen. Der Dozent ist bei Problemen ansprechbar.

Die Konzeption dieses Moduls und seine Positionierung im Rahmen der Ausbildung sind in folgendem Paper beschrieben:

*Design Patterns: Between Programming and Software Design*,  
Christoph Denzler and Dominik Gruntz, International Conference on Software Engineering  
in: Proceedings of ICSE'08, Educational Track, p. 801-804, 2008.  
<http://doi.acm.org/10.1145/1368088.1368202>

## 2.2. Zeitaufwand für Studierende (summarisch)

Kontaktunterricht:	25h
Begleitetes Selbststudium:	10h
Unbegleitetes Selbststudium während Unterrichtszeit:	35h
Unbegleitetes Selbststudium während unterrichtsfreier Zeit:	20h
Gesamtaufwand:	90h

## 2.3. Leistungsbeurteilung

### 2.3.1 Leistungsbeurteilung im Semester

Die Bewertung der Studierenden erfolgt durch eine schriftliche Prüfung (angekündigt) von 90Min. In dieser Prüfungen werden sowohl Verständnisfragen als auch Programmierfragen gestellt, die durchaus

Bezug auf die Übungen nehmen können.

Teil 1: 30 Min, ohne Hilfsmittel

Teil 2: 60 Min, zwei Seiten A4 Notizen / Zusammenfassung

Abwesenheiten von den Prüfungen werden nur aus folgenden Gründen akzeptiert: Krankheit (Arztzeugnis als Beleg vorweisen, eMail an Dozierende vor oder am Tag der Prüfung), Militärdienst oder Beurlaubung durch die Studiengangleitung. Eine aus diesen Gründen versäumte Prüfung wird mit einer Nachprüfung nachgeholt.

Wer aus anderen Gründen eine Prüfung verpasst der erhält für die Prüfung die Note 1.0 und das Modul gilt als nicht bestanden. Ein nicht bestandenes Modul kann einmal wiederholt werden.

#### **2.3.2 Modulschlussprüfung**

Schriftlich, 90 Min.

Teil 1: 30 Min, ohne Hilfsmittel

Teil 2: 60 Min, vier Seiten A4 Notizen / Zusammenfassung

### **2.4. Arbeitsmittel**

Folgende Unterlagen werden zur Verfügung gestellt:

- Folienkopien aus der Vorlesung
- Framework für Übungen

Als Literatur wird eines der folgenden Bücher empfohlen:

- Head First Design Patterns, O'Reilly
- Erich Gamma et al, Design Patterns, Addison-Wesley
- Mark Grand, Patterns in Java, John Wiley
- Cay Horstmann, Object-Oriented Design & Patterns, John Wiley

Einige Bücher zum Thema Design Patterns sind von der FHNW-Bibliothek als E-Books lizenziert und damit innerhalb des FHNW-Netzwerkes (bzw. via VPN) online zugreifbar. Eine entsprechende Literaturliste finden Sie auf dem AD.

## **3. Themen der Unterrichtsblöcke**

### **3.1. Objektorientierte Programmierung "revisited" (1 Woche)**

In einem ersten (kurzen) Block werden die wichtigsten Grundbausteine aus der objektorientierten Programmierung wiederholt: Klassen und Interfaces, abstrakte Basisklassen, Polymorphismus und Vererbung. Das Modul Design Patterns baut auf diesen Grundlagen auf. Anhand eines Beispiels wird aufgezeigt, warum die Verwendung von Entwurfsmustern sinnvoll ist und welche Qualitätsziele damit erreicht werden können: Erweiterbarkeit / Open-Closed Prinzip.

### **3.2. Entwurfsmuster (13 Wochen)**

Im Rahmen der Fallstudie „Grafikeditor“ werden verschiedene Entwurfsmuster eingeführt und von den Studierenden umgesetzt. Konkret handelt es sich um die Muster

- Observer
  - Strategy / State
  - Composite
  - Prototype
  - Factory
  - Singleton
  - Decorator
  - Proxy
  - Command
  - Dependency Injection
- und allenfalls weitere Muster

### **3.3. Design Prinzipien (1 Woche)**

In einem dritten Block werden Gemeinsamkeiten aus den behandelten Entwurfsmustern diskutiert. Zudem wird die Anwendung der Muster kritisch hinterfragt: Wann soll ein erweiterbares Design gewählt werden und wann nicht?

## 4. Drehbuch

W	Inhalt	KS	BSS	USS	Tot
1	Einführung Design & Design Patterns (Erweiterbarkeit und Evolvierbarkeit), Rekapitulation OOP (Bausteine für Design: Abstrakte Klassen, Interfaces, Polymorphismus) Übung 1: Color Picker	1.5	0.75	3	5.25
2	Observer Pattern: Besprechung des Color-Pickers und Einführung Observer Pattern inkl. Beispiele Übung 2: Modell für Grafikeditor (Observer Pattern)	1.5	0.75	3	5.25
3	Observer Pattern: Spezielle Aspekte des Observer Patterns und Hilfe bei der Arbeit am Grafikeditor. Übung 3: Figurenpalette	1.5	0.75	3	5.25
4	State Pattern: Motivation und Einführung. Übung 4: Figuren Handles (State Pattern)	1.5	0.75	3	5.25
5	Strategy Pattern: Spezielle Themen Übung 5: Grid (Strategy Pattern)	1.5	0.75	1.75	4
6	Composite Pattern: Motivation und Einführung und Behandlung spezieller Themen Übung 6: Gruppenfiguren (Composite Pattern)	1.5	0.75	1.75	4
7	Prototype Pattern: clone in Java Anwendung (Immutability), allenfalls spezielle Probleme in Java Übung 7: Copy & Paste (Prototype Pattern)	2	0.25	2	4.25
8	Immutability Übung 8: Save & Load	1.5	0.75	1.5	3.75
9	Prüfung (2 Lekt) (USS = Vorbereitung auf Prüfung)	1.5	0	6	7.5
10	Decorator: Motivation und Einführung Übung 9: Take & Wrap (Decorator Pattern)	1.5	0.75	1.75	4
11	Decorator + Proxy Pattern: Diskussion spezieller Aspekte Nachbesprechung der Prüfung	1.5	0.75	1.75	4
12	Command Pattern Übung 10: Undo/Redo (Command Pattern)	1.5	0.75	2	4.25
13	Abstract Factory: Motivation und Beispiele, Idee der Komponentensoftware. Übung 11: Konfiguration (Dependency Injection Pattern)	1.5	0.75	3	5.25
14	Singleton Pattern	1.5	0.75	2	4.25
15	Zusammenfassung	1.5	0.75	0	2.25
UFZ	Vorbereitung Modulschlussprüfung + MSP	2	0	19.5	21.5
	<b>Total</b>	<b>25</b>	<b>10</b>	<b>55</b>	<b>90</b>

## 5. Termine

Die schriftliche Prüfung findet am Dienstag, 13.11.2018 statt, von 10:00 – 11:30 bzw. von 11:30 – 13:00.