

OOSE13

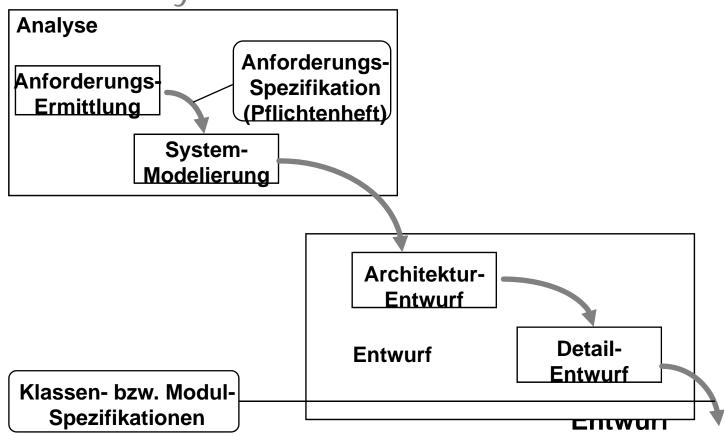
Von der Analyse (OOA) zum Entwurf (OOD) (Wiederholung)

Lehrstuhl Softwaretechnologie, Dr. Birgit Demuth Sommersemester 2016





Von der Analyse zum Entwurf







Einige Kriterien für guten Entwurf

- Korrektheit
 - Erfüllung der Anforderungen
 - Wiedergabe aller Funktionen des Systemmodells
 - Sicherstellung der nichtfunktionalen Anforderungen
- Verständlichkeit
 - Gute Dokumentation
- Anpassbarkeit
- Hohe Kohäsion innerhalb der Komponenten
- Schwache Kopplung der Komponenten
- Wiederverwendung
- Stabilität und Zuverlässigkeit
- Angemessene Ressourcenverwendung





Hohe Kohäsion + Schwache Kopplung

Subsystem A

(z.B. Benutzungsoberfläche)

Subsystem B

(z.B. fachlicher Kern)

- Subsystem B darf keine Information und Funktionalität enthalten, die zum Zuständigkeitsbereich von A gehört und umgekehrt.
- Es muss möglich sein, Subsystem A weitgehend auszutauschen oder zu verändern, ohne Subsystem B zu verändern.
- Die meisten Änderungen von Subsystem B sollten nur relativ einfache Änderungen in Subsystem A nach sich ziehen.
- Beispiele zur konkreten technischen Realisierung siehe MVC-Architektur und Entwurfsmuster

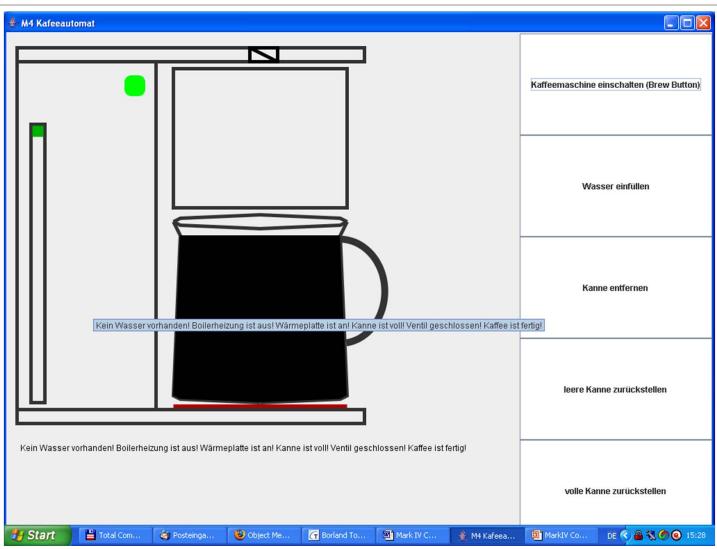




FALLSTUDIE MARK IV COFFEEMAKER /1/,/2/







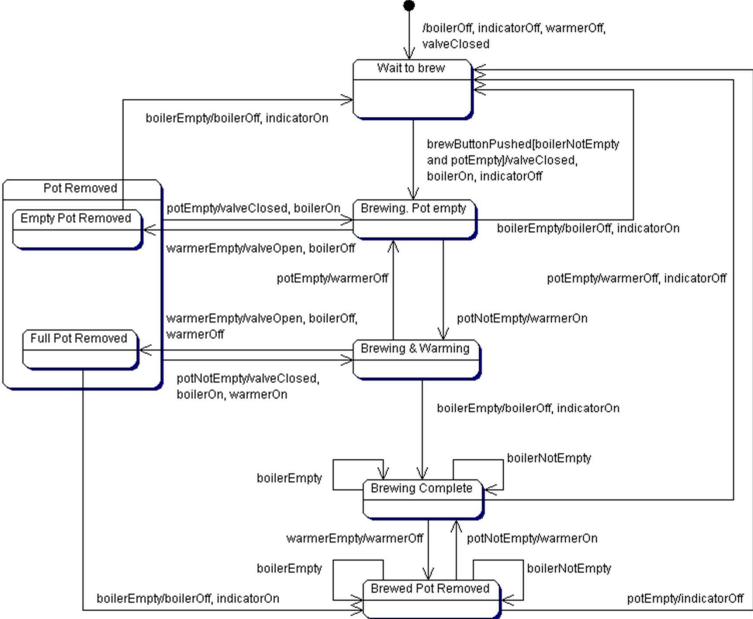




Teile und Funktionen von MarkIV Coffeemaker

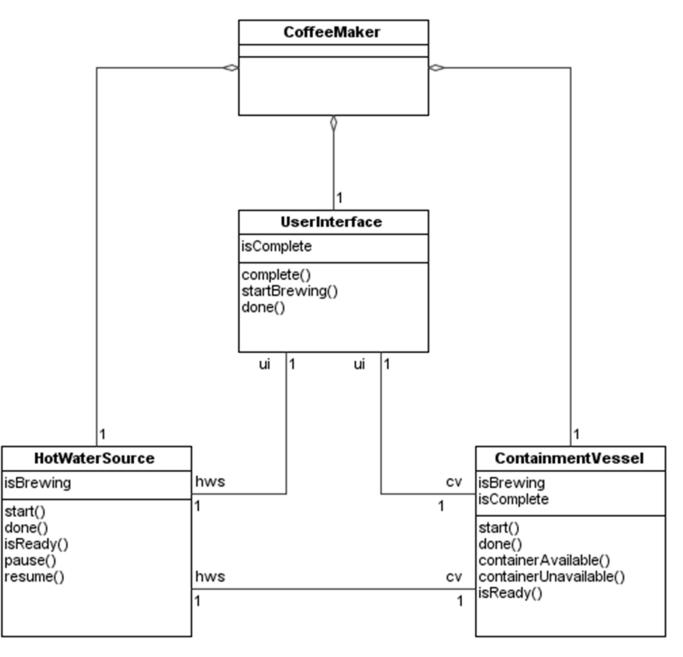
- Heizung für den Boiler (boilerOn/boilerOff)
- Heizung für die Wärmeplatte (warmerOn/warmerOff)
- Sensor f
 ür die W
 ärmeplatte (warmerEmpty, potEmpty, potNotEmpty)
- Sensor f
 ür den Boiler (boilerEmpty, boilerNotEmpty)
- Zubereitungsknopf (brew)
- Anzeigelampe (indicatorOn/indicatorOff) leuchtet auf, wenn der Zubereitungsvorgang beendet und der Kaffee fertig ist.
- Druckventil (valveOpen/valveClosed), das geöffnet wird, um den Druck im Boiler zu reduzieren; der Druckabfall unterbricht den Wasserzufluss zum Filter; das Ventil kann geöffnet oder geschlossen sein.



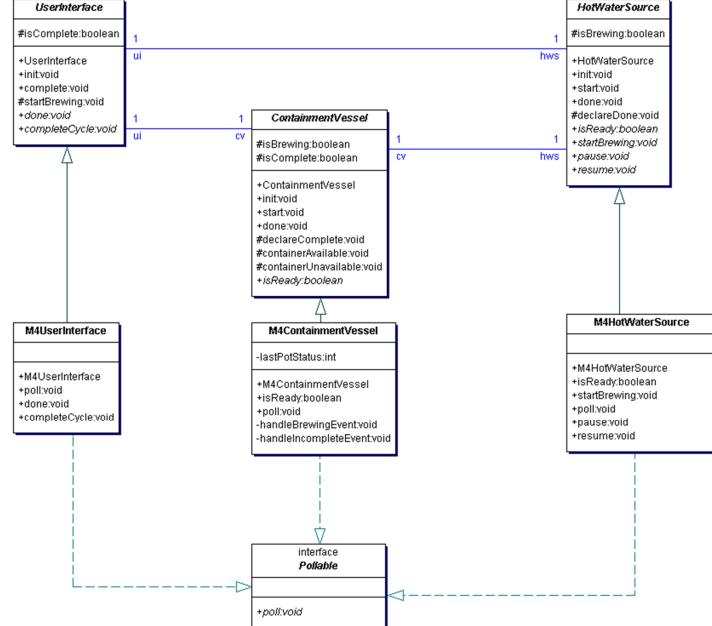




OOA: Analyseklassendiagramm











CoffeeMaker API Spezifikation der Hardwarefunktionen

interface CoffeeMakerAPI +api:CoffeeMakerAPI=null +WARMER_EMPTY:int=0 +POT_EMPTY:int=1 +POT_NOT_EMPTY:int=2 +BOILER_EMPTY:int=0 +BOILER_NOT_EMPTY:int=1 +BREW_BUTTON_PUSHED:int=0 +BREW BUTTON NOT PUSHED:int=1 +BOILER_ON:int=0 +BOILER_OFF:int=1 +WARMER_ON:int=0 +WARMER_OFF:int=1 +INDICATOR ON:int=0 +INDICATOR OFF:int=1 +VALVE_OPEN:int=0

- +getWarmerPlateStatus():int
- +getBoilerStatus():int
- +getBrewButtonStatus():int

+VALVE_CLOSED:int=1

- +setBoilerState(boilerStatus:int):void
- +setWarmerState(warmerState:int):void
- +setIndicatorState(indicatorState:int):void
- +setReliefValveState(reliefValveState:int):void
- +getPlateOn():boolean
- +getBoilerOn():boolean
- +getIndicatorState():boolean
- +getReliefValveState():boolean
- +setBoilerFillState(boilerFillStatus:int):void
- + setBrewButtonState(brewButtonStatus:int):void
- +setPotState(potStatus:int):void

M4CoffeeMakerAPIImplementation

- +buttonPressed:boolean
- +lightOn:boolean
- +boilerOn:boolean
- +valveClosed:boolean
- +plateOn:boolean
- +boilerEmpty:boolean
- +potPresent:boolean
- +potNotEmpty:boolean
- +M4CoffeeMakerAPIImplementation()





Schnittstellen und abstrakte Klassen in Java

Abstrakte Klasse	Schnittstelle (Interface)
Attribute, Konstanten, Operationen	Operationen und ggfs. Konstanten
Kann Default-Verhalten festlegen	Kann KEIN Default-verhalten festlegen
Default-Verhalten kann in Unterklassen überschrieben werden	Überschreiben von Methoden ist nicht möglich
Unterklasse kann nur von einer Klasse erben	Eine Klasse kann mehrere Schnittstellen implementieren
Verwendung für Implementierungsvererbung	Verwendung für Spezifikationsvererbung





MarkIV CoffeeMaker Entwurfsüberlegungen

```
public class CoffeeMaker {
   public static void main(String[] args) {
      CoffeeMakerAPI api = new M4CoffeeMakerAPIImplementation();
     M4UserInterface ui = new M4UserInterface(api);
     M4HotWaterSource hws = new M4HotWaterSource(api);
     M4ContainmentVessel cv = new M4ContainmentVessel(api);
     ui.init(hws,cv);
     hws.init(ui,cv);
     cv.init(ui,hws);
     while(true) {
        ui.poll();
        hws.poll();
        cv.poll();
SS 2016
                            Softwaretechnologie / OOSE 13
```





Implementierung

- Implementierung der Interfaces/Klassen des OOD-Klassendiagramms
- Wie wird nun die Kaffeemaschine MarkIV zum ablauffähigen Programm?
 - Klasse CoffeeMaker also Kommandozeilenprogramm (aber da sieht man nichts :- (Endloszyklus)
 - GUI zur Simulation der Kaffeemaschine, d.h. etwas zum Spielen ©
 - Erweiterung unserer Klasse CoffeeMaker um eine GUI
 - Implementierung mit dem Observer Pattern
 - MarkIVFinal.jar
 - Nächster Schritt: Mark IV mit Threads für die einzelnen Sensoren.





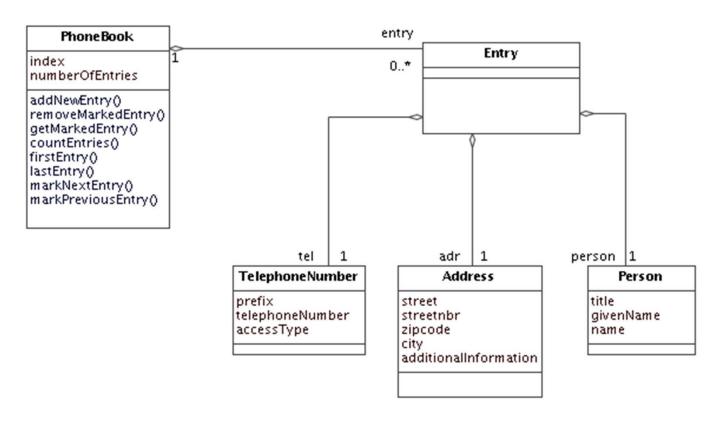
Vom Analysemodell zum Entwurfsmodell **TELEFONBUCH**



SS 2016



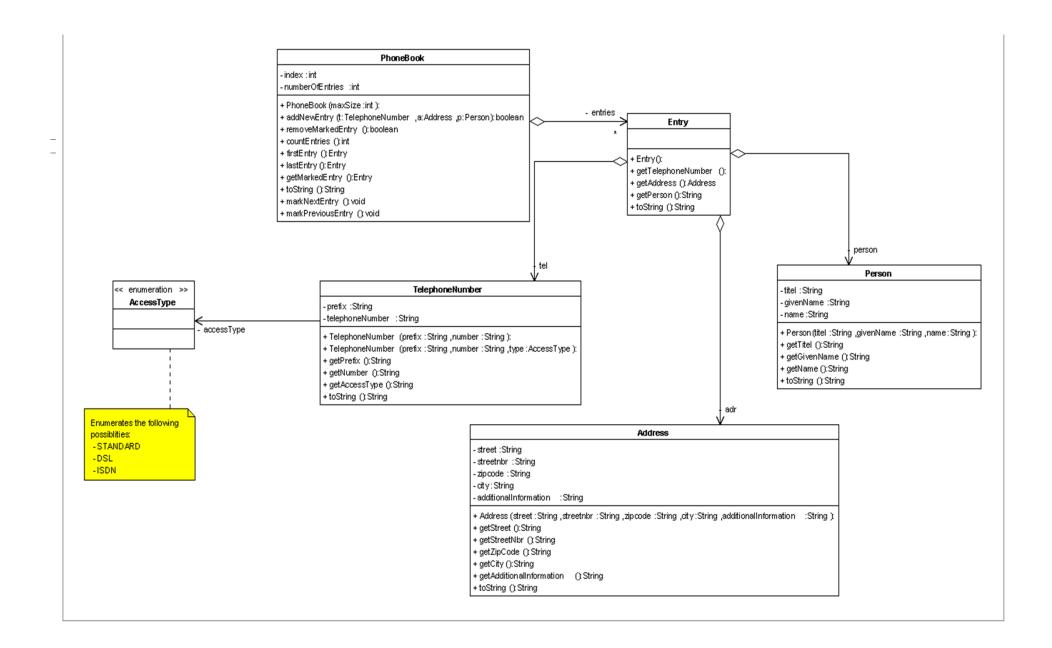
Analysemodell des Telefonbuches















Hinweise zur Vorbereitung auf die Klausur

ZUSAMMENFASSUNG





Referenzen

- (1) Robert Martin: Designing Object-Oriented C++ Applications Using the Booch Method. Prentice Hall, 1995
- (2) Robert Martin: UML for Java Programmers. Prentice Hall, 2003
- (3) Birgit Demuth (Hrsg.): Softwaretechnologie für Einsteiger. Pearson Studium,2. geänderte Auflage, 2014

Weitere Quellen zur Vorbereitung auf die Klausur

- a) Vorlesungsfolien und Übungsmaterial
- b) Learning Outcomes
- c) INLOOP-Aufgaben
- d) Auditorium
- e) Lernraum heute 6. DS
- f) Klausuren auf IFSR-Server (ftp://ifsr.de/klausuren/SWT/)
- g) Wissensfragen im Selbsttest (http://bit.do/OOSE_Test)