

# [MO] 2009 SS

Sonntag, 17. April 2016

18:24

## 1. Erläutern der Begriffe

### a. Use Case

Ein Use-Case beschreibt die Interaktion zwischen einem Akteur und einem System. Dieses System kann ein IT-System oder auch ein Geschäftssystem sein. Der Use-Case besteht aus einer Menge von Aktionen. Er bringt dem Akteur in irgendeiner Form einen Nutzen.

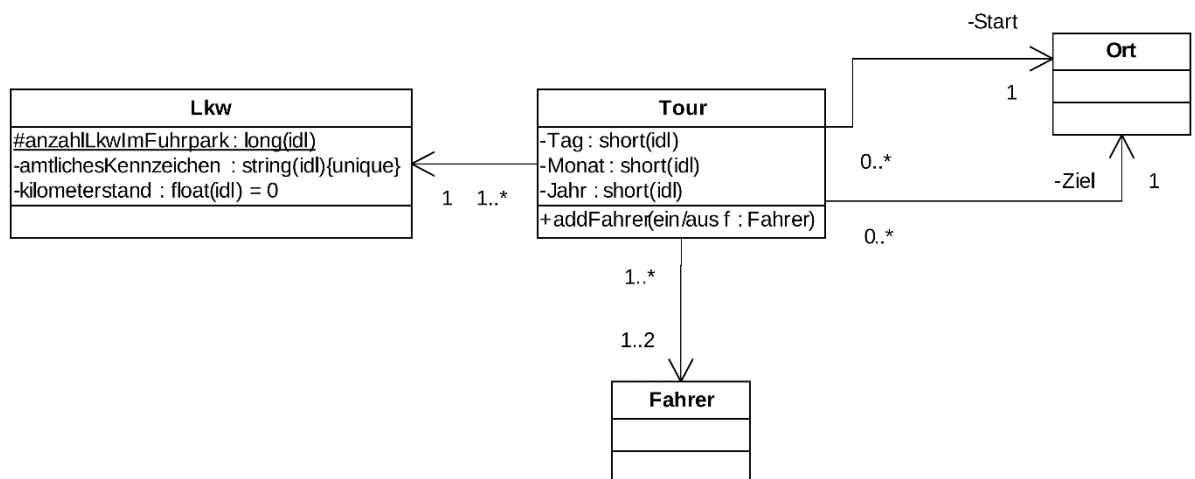
### b. Model driven architecture

Begriff der Object Management Group. Man versucht, den Aufbau eines Softwaresystems zunächst unabhängig von einer Plattform oder Programmiersprache in Modellform zu entwickeln. Anschließend wird der Quellcode teilweise automatisiert erzeugt.

### c. Kapselung

OO-Prinzip, nach dem Informationen vor direktem externen Zugriff abgeschottet werden. Der Zugriff darf - sofern erlaubt- nur über explizit definierte Schnittstellen erfolgen.

## 2. Arbeiten Sie Mit dem Klassendiagramm



### a. Programmieren Sie die angegebenen Attribute der Klasse LKW in Java

```
class Lkw {
    protected static long anzahlLkwImFuhrpark;
    private String amtlichesKennzeichen;
    private float kilometerstand = 0;
}
```

### b. Programmieren Sie die angegebenen Assoziationen in Java. Berücksichtigen Sie dabei die Navigierbarkeit nur, insofern die UML sie zwingend vorschreibt

```
class Tour {
    private Lkw lkw;
    private Fahrer fahre;
    private Ort start;
    private Ort ziel;
}
```

- c. Welche Notationselemente des vorliegenden Klassendiagramms lassen sich nicht unmittelbar in Attribute einer Java-Klasse übertragen? Beschreiben sie verbal, an welcher Stelle im Programm Sie die entsprechenden Anforderungen als Java-Anweisungen implementieren würden. Die Implementierung ist zunächst nicht erforderlich [3 Min.]

Die Attribut-Eigenschaft {unique} besitzt keinen eigenen modifier in Java, es muss also in Kontruktor und Setter-Methoden eine entsprechende Prüfroutine eingebaut werden.

Die Übergabe eines Parameters als inout muss in Java indirekt implementiert werden, da Java ausschließlich Pass-by-Value kennt, was aber aufgrund der JVM-Implementierung Referenz-ähnliche Eigenschaften aufweist. Um nun dafür zu sorgen, dass das Ergebnis des Parameterrufs auch wieder in der ursprünglichen Variable landet, ist eine Implementierung ähnlich dieser Form notwendig:

```
variable = methode (variable);
```

und im Methodenrumpf:

```
return variable;
```

3. Zeichnen Sie ein UML-Klassendiagramm für ein System zur Verwaltung eines Fußballturniers im Modus "Jeder gegen Jeden". Berücksichtigen sie dabei folgende Anforderungen:

- Jede Mannschaft wird durch einen eindeutigen Namen beschrieben
- Jede Mannschaft kann bis zu 20 Spieler als spielberechtigt nominieren. Sie muss mindestens 11 Spieler nominieren
- Für jede Mannschaft sollen mindestens folgende Attribute berücksichtigt werden: Punkte, erzielte Tore, Kassierte Tore, Tordifferenz.
- Das System soll das Durchschnittsalter der Spieler einer Mannschaft und das Durchschnittsalter aller Spielberechtigten Spieler des Turniers kennen.
- Das System soll die Anzahl der Spieler einer Mannschaft und die Anzahl aller Spielberechtigten Spieler des Turniers kennen.
- Jeder Spieler ist einer von 4 Kategorien zugeordnet: Torwart, Verteidiger, Mittelfeldspieler, Stürmer
- Eine Spielbegegnung zwischen zwei Mannschaften wird durch Austragungsort und -Zeit beschrieben und endet mit einem Spielergebnis, das aus den Toren der beiden Mannschaften besteht (z.b. 3:1 oder 2:2)
- Für jeden Spieler werden Name und Geburtsdatum sowie eine eindeutige Spielernummer hinterlegt.

**Hinweise:**

- Konstruktoren sowie Getter- und Setter-Methoden brauchen Sie nicht zu modellieren.
- Modellieren Sie alle notwendigen Assoziationen. Die Attribute, mit denen diese Assoziationen implementiert werden, brauchen Sie nicht zu modellieren.
- Entscheiden Sie sich bei allen Assoziationen für eine sinnvolle Modellierung ihrer Navigierbarkeit.

ggf. überarbeiten



# [SE] 2009/2010 WS

Sonntag, 17. April 2016

19:01

[75 Min. / 30 Punkte ==> 2,5p/Min.]

## 1. Erläutern Sie folgende Begriffe:

### a. Echtzeitsystem

System, welches unmittelbar auf Eingaben reagiert. Dies erfolgt üblicherweise mittels nebenläufiger, miteinander kooperierender Prozesse. Es wird unterschieden in "soft real-time", bei welcher die Funktionsweise sich verschlechtert, wenn die Ergebnisse nicht rechtzeitig vorliegen und "hard real-time", bei welchen die die Funktionsweise bei nicht rechtzeigen Ergebnissen gestört ist.

### b. Softwarekomponente

Im komponentenbasierten Software Engineering sind einzelne Komponenten voneinander unabhängig. Die Kommunikation zwischen einzelnen Komponenten wird durch ein Komponentenmodell festgelegt.

### c. Software Reengineering

Zur Verbesserung der Wartbarkeit eines bereits existierenden Systems durch Neuimplementierung. Die kann neue Architektur, Dokumentation und bzw. oder Sprache bedeuten.

Die Funktionalität bleibt gleich. Eine Neuentwicklung würde mehr kosten. Diese wird aber meist in Kauf genommen, um neue Funktionen berücksichtigen zu können.

## 2. Ein Fußballverband möchte seinen Ligaspielbetrieb künftig mit Hilfe eines objektorientiert entwickelten Web-Anwendungssystems verwalten. Die Anforderungen werden grob beschrieben:

- Die Ergebnisse werden von Schiedsrichtern am Abend nach Spielende via Internet eingepflegt.
- Wenn alle Ergebnisse eines Spieltages in einer Liga vorliegen, gibt der Ligaausschussvorsitzende die Ergebnisse frei.
- Das System ermittelt daraus die aktuelle Tabelle, die für jedermann im WWW abrufbar ist.
- Das Sekretariat des Verbandes pflegt die Stammdaten des Systems (Spieler, Mannschaften)
- Jede Mannschaft wird durch einen eindeutigen Namen beschrieben.
- Jede Mannschaft kann bis zu 20 Spieler als spielberechtigt nominieren. Sie muss mindestens 11 Spieler nominieren.
- Für jede Mannschaft sollen mindestens folgende Attribute berücksichtigt werden: Punkte, erzielte Tore, kassierte Tore, Tordifferenz.
- Das System soll das Durchschnittsalter der Spieler einer Mannschaft und das Durchschnittsalter aller spielberechtigten Spieler kennen.
- Das System soll die Anzahl der Spieler einer Mannschaft und die Anzahl aller spielberechtigten Spieler des Turniers kennen.
- Jeder Spieler ist einer von 4 Kategorien zugeordnet: Torwart, Verteidiger, Mittelfeldspieler, Stürmer.
- Eine Spielbegegnung zwischen zwei Mannschaften wird durch den Austragungsort und -zeit beschrieben und endet mit einem Spielergebnis, das aus den Toren der beiden Mannschaften besteht. (z.B. 3:1 oder 2:2)
- Für jeden Spieler werden Name und Geburtsdatum sowie eine eindeutige Spielernummer hinterlegt.

- a. **Entscheiden Sie sich für ein geeignetes Vorgehensmodell für das Softwareprojekt und begründen Sie ausführlich Ihre Entscheidung. [20 Min]**

#

#

- b. **Begründen Sie, warum Ihres Erachtens die Entscheidung für eine Individualsoftware oder für eine Standardsoftware getroffen werden soll, indem Sie Vor- und Nachteile von Individual- und Standardlösung für diesen konkreten Fall gegeneinander abwägen [20 Min]**

#

#

3. **Beantworten Sie bitte die folgenden Frage kurz, aber in ganzen Sätzen.**

**Ein von Ihnen entwickeltes Softwaresystem läuft zur Zufriedenheit der überwiegenden Mehrheit der Anwender. Einige Anwender klagen aber, das System sei höchst instabil und unzuverlässig. Welchen Grund vermuten Sie für diese Diskrepanz und wie reagieren Sie? [7 Min]**

Eine mögliche Ursache ist die Verwendung unterschiedlicher Teilmengen der Funktionalität durch unterschiedliche Nutzergruppen. Es wäre demnach erforderlich herauszufinden, wie die Nutzergruppe, die das System als unzuverlässig erlebt, die Software nutzt, welche Funktionalität mit welchen Daten, in welcher Reihenfolge, etc.

4. **M. Wagner schreibt in c't 13/2008 zum Thema Agile Softwareentwicklung folgendes:**

**„Uneinigkeit herrscht auch in Bezug auf die Ressourceneffekte. Der Vorwurf lautet, dass durch die häufigen Release-Zyklen der Testaufwand im Vergleich zu herkömmlichen Vorgehensweisen deutlich steigt. Für jeden der kurzen Release-Zyklen muss immer wieder der gesamte Funktionsumfang getestet werden, wohingegen man bei traditionellen Verfahren nur einmal zum Abschluss testet.“**

**Begründen Sie Ihre Meinung zu dieser Aussage. (5 Punkte)**

Vorteile Tests: frühes Fehler entdecken, dadurch weniger Fehlentwicklung

Nachteil: erhöhter Aufwand für Tests

# [MO] 2009/2010 WS

Mittwoch, 20. April 2016

14:44

## 1. Erläutern Sie die folgenden Begriffe

- **Akteur (in einem Use Case)**

Ein Akteur steht außerhalb des Systems und kann Use Cases initiieren. Der Akteur kann ein System oder eine Rolle, aber keine Person sein. Akteure können mittels Generalisierung verbunden werden, dies bedeutet in der Praxis aber keine strenge ist-ein-Beziehung.

Akteure werden normalerweise als Strichmännchen dargestellt, können bei externen Systemen aber als Rechteck bzw. Symbol (z.B. Logo) dargestellt werden.

- **Separation of Concerns**

Aufteilung des Softwaresystems in Komponenten anhand seiner Anforderungen

- **Geschäftsprozess**

'zusammenhängenden abgeschlossene Folge von Tätigkeiten, die zur Erfüllung einer betrieblichen Aufgabe notwendig sind'.

Ein Geschäftsprozess zieht sich meist durch mehrere Organisationseinheiten.

## 2. Betrachten Sie die Klassendiagramme A und B sowie Ausschnitte aus den zugehörigen Implementierungen 1 und 2. Welche der Implementierungen gehört zu welchem der Diagramme? Begründen Sie Ihre Antwort

**Klassendiagramm A**

**Klassendiagramm B**

**Implementierung 1:**

```
public class Ganzes {  
    ...  
    public void addTeil(Teil t){...}  
    ...  
}
```

### **Implementierung 2:**

```
public class Ganzes {  
    ...  
    public void addTeil(){Teil t = new Teil(...);...}  
    ...  
}
```

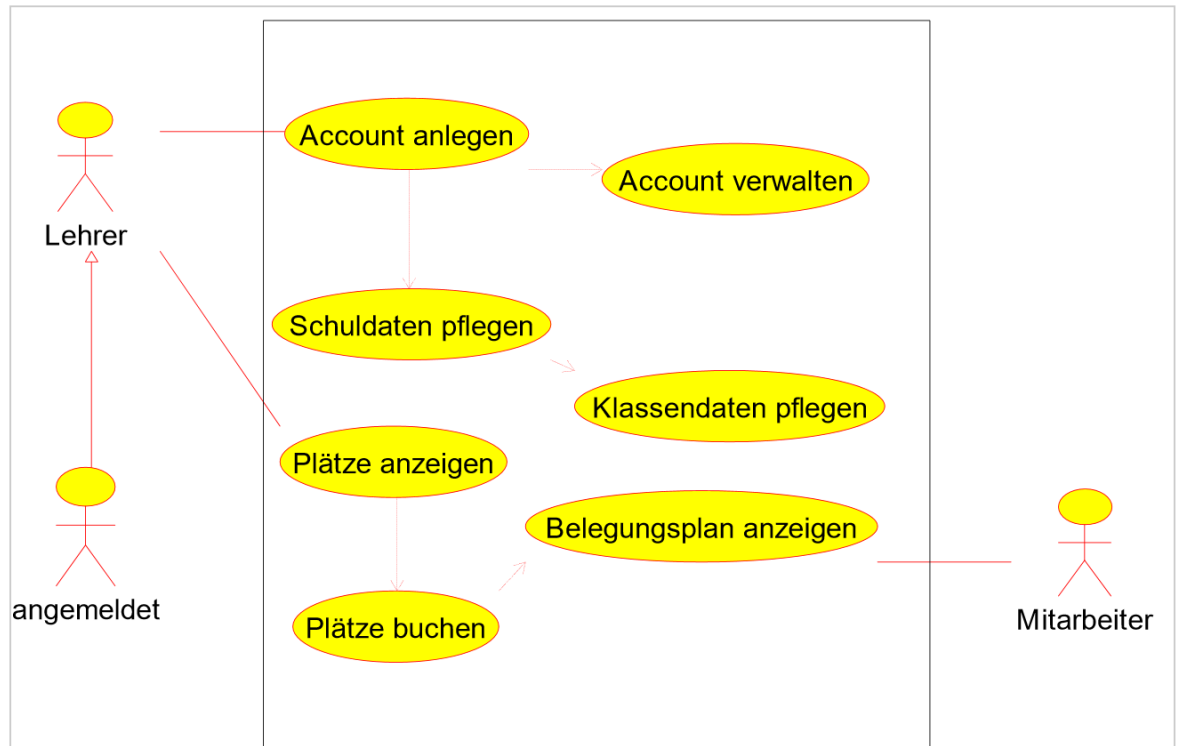
Implementierung 1: Gehört zur Aggregation, da hier ein add im Code steckt, das bedeutet das Ganze ist von seinen Teilen unabhängig. Die Klasse des Ganzen erzeugt ihre Teile nicht selbst.

Implementierung 2: Gehört zur Komposition, da hier ein new im Code steckt, somit ist das Ganze von seinen Teilen abhängig. Die Klasse des Ganzen erzeugt seine Teile selbst.

3. **Ein kommunales Schullandheim hat derzeit nur einen Webauftritt, der aus statischen Webseiten besteht. Auf ihnen werden das Heim vorgestellt, Bilder der Zimmer gezeigt, Preise genannt und Kontaktdaten des Herbergsvaters veröffentlicht. Bislang kennt sich niemand im Haus mit Internettechnologien aus.**

**Ein neuer Zivildienstleistender schlägt nun vor, den Webauftritt zu erweitern, weil die Schullandheime und Herbergen in der weiteren Umgebung schon viel umfangreicher im Web vertreten sind. Er ist bereit, die neuen Seiten mit Hilfe der Programmiersprache PHP zu entwickeln, die er bereits beherrscht. Der Herbergsvater überlegt sich folgende Anforderungen:**

- In der neuen Version sollen die für den Schullandheimaufenthalt verantwortlichen Lehrer, die für ihre Klassen ein Schullandheim suchen, online über die Anzahl der noch verfügbaren Plätze in einer bestimmten Woche informiert werden.
  - Sie können dann online eine verbindliche Buchung vornehmen.
  - Die verantwortlichen Lehrer können einen Benutzeraccount anlegen und ein Passwort pflegen.
  - Die verantwortlichen Lehrer können dann online nach Eingabe des Passwortes Daten einpflegen. Für jede Schule können hinterlegt, eingesehen und geändert werden:
    - Name, Vorname, Privatadresse, private Telefonnummer und Mobiltelefonnummer des verantwortlichen Lehrers
    - Adresse, Telefonnummer und Name der Schule
    - Schulart
  - Für jeden Aufenthalt einer Klasse können hinterlegt, eingesehen und geändert werden:
    - Name, Vorname und Mobiltelefonnummer aller begleitenden Lehrer
    - Anzahl der Mädchen und Jungen, die angemeldet werden.
    - Anzahl der Schüler unter 18 Jahre
    - Anzahl der Schüler über 18 Jahre
    - Die Mitarbeiter des Schullandheimes können die Daten aller Kunden ansehen, und verändern, wenn z.B. telefonisch Änderungen übermittelt werden.
  - Die Mitarbeiter des Schullandheimes können sich einen Belegungsplan anzeigen lassen, der für jede Kalenderwoche Auskunft über die Auslastung des Heimes gibt.
- a. **Zeichnen Sie ein Use-Case-Diagramm für den neuen Webauftritt [8 Min.]**
- b. **Zeichnen Sie ein Klassendiagramm für das System [10 Min.]**
- **Konstruktoren, Getter/Settermethoden müssen nicht modelliert werden**
  - **Assoziation sollen modelliert werden (die Attribute dieser Assoziationen können weggelassen werden)**
  - **Sinnvolle Modellierung von Navigierbarkeit und Multiplizitäten**



## [MO] 2011/2012 WS

Dienstag, 19. April 2016

21:05

### 1. Erläutern Sie folgende Begriffe:

#### a. OOAD

OOAD bedeutet objektorientierte Analyse und Design und bezeichnet die Anforderungsanalyse und Systementwurf nach OO-Prinzipien beim Entwickeln eines Softwaresystems.

Die UML bietet Werkzeuge zum standardisierten Erfassen von Analyse und Design.

#### b. Akteur

Eine generische Einheit im Use-Case-Diagramm, welche eine Rolle (keine konkrete Person) darstellt. Diese Rolle wird von einem externen Benutzer oder System eingenommen und interagiert mit dem System. Der Akteur steht immer außerhalb des Systems.

#### c. Synchrone Nachricht

Der Sender wartet bis der Empfänger die Nachricht abgearbeitet hat. Der Empfänger antwortet auf diese Nachricht.

### 2. Die UML beinhaltet 13 [+Profildigramme] Diagrammarten. Die beiden meistverwendeten sind das Klassendiagramm und das Use-Case-Diagramm. Nennen Sie 6 weitere Diagrammarten der UML.

- Sequenzdiagramm
- Paketdiagramm
- Aktivitätsdiagramm
- Timingdiagramm
- Zustandsdiagramm
- Objektdiagramm



⇒ SPATZ'O

3. Erstellen Sie ein Use-Case-Dokument für den Anwendungsfall "Freundschaftsanfrage bestätigen" bei einem sozialen Netzwerk wie Facebook. Ein Use-Case-Diagramm sollen sie nicht zeichnen.

Use Case: Freundschaftsanfrage bestätigen

Description

Nutzer bestätigt Freundschaftsanfrage eines anderen Nutzers auf Facebook

Actors

Facebook-Nutzer

**Preconditions:**

- Nutzer ist auf dem Facebook-Webportal angemeldet
- ein anderer Nutzer hat diesem eine Freundschaftsanfrage gesendet

**Basic Flow**

1. Nutzer erhält visuelle Benachrichtigung, dass er eine Freundschaftsanfrage erhalten hat.
- 2a. Nutzer klickt/tapt auf Benachrichtigung um Anfrage zu öffnen
- 2b. Nutzer ignoriert Benachrichtigung --> Ende Basic Flow. Ggf. Alternate Flow
3. System öffnet Freundschaftsanfrage. Dem Nutzer wird ein erklärender Text, Profilbild und Name des Anfragenden Nutzers sowie Annahme-, Ablehnung- und Abbruchschaltflächen angezeigt.
- 4a. Nutzer bestätigt Anfrage mit Klick/Tapp auf entsprechende Schaltfläche
- 4b. Nutzer lehnt Annahme mit Klick/Tapp auf Schaltfläche ab
- 4c. Nutzer bricht Vorgang mit Klick/Tapp auf Schaltfläche ab --> Ende UC
5. Hat der Nutzer die Anfrage angenommen bzw. abgelehnt, wird ihm die Aktion mit einer visuellen Meldung bestätigt. Diese Meldung enthält eine Schaltfläche zum Rückgängig machen der Aktion
- 6a. Nutzer zufrieden --> UC beendet
- 6b. User macht Aktion mit Klick auf Schaltfläche rückgängig --> UC beendet. Ggf. Alternate Flow

**Alternate Flow**

1. Nutzer hat Benachrichtigung über Anfrage vorher ignoriert
2. Nutzer öffnet Nachrichtencenter und klickt auf Anfrage --> Basic Flow 3.

**Postconditions**

- Nutzer wurde über Freundschaftsanfrage informiert
- Nutzer hat sich entschieden, in einer der drei Optionen zu reagieren: Annahmen, Ablehnung, Auf Später verschieben

4. Eine Gärtnerei will einen Katalog mit ihrem vollständigen Sortiment online verfügbar machen, um Kunden zum Besuch im Ladengeschäft zu animieren. Eine Bestellfunktion ist zunächst nicht vorgesehen. Die Nutzer melden sich nicht an, sondern bleiben Anonym.

Entwerfen Sie ein Klassendiagramm, wenn der Katalog folgende Pflanzen enthalten soll:

1. Laubgehölze werden beschrieben durch:
  - Kategorie (laubabwerfend oder immergrün)
  - Lieferform (Containerware oder Wurzelware)
  - Standort (sonne, Halbschatten oder Schatten)
  - Blütenfarbe (weiß, rot, gelb, blau, orange, rosa, violett)

Blütemonat  
Wuchsbreite in cm  
Wuchshöhe in cm  
Wuchsgeschwindigkeit in cm/Jahr  
Preis

2. Nadelgehölze werden beschrieben durch:  
Lieferform (Containerware oder Wurzelware)  
Standort (Sonne, Halbschatten oder Schatten)  
Wuchsbreite in cm  
Wuchshöhe in cm  
Wuchsgeschwindigkeit in cm/Jahr  
Preis

3. Kletterpflanzen werden beschrieben durch:  
Lieferform (Containerware oder Wurzelware)  
Kategorie (laubabwerfend oder Wurzelware)  
Standort (Sonne, Halbschatten oder Schatten)  
Blütenfarbe( weiß, rot, gelb, blau, orange, rosa, violett)  
Blütemonat  
Herbstfärbung (Ja/Nein)  
Preis

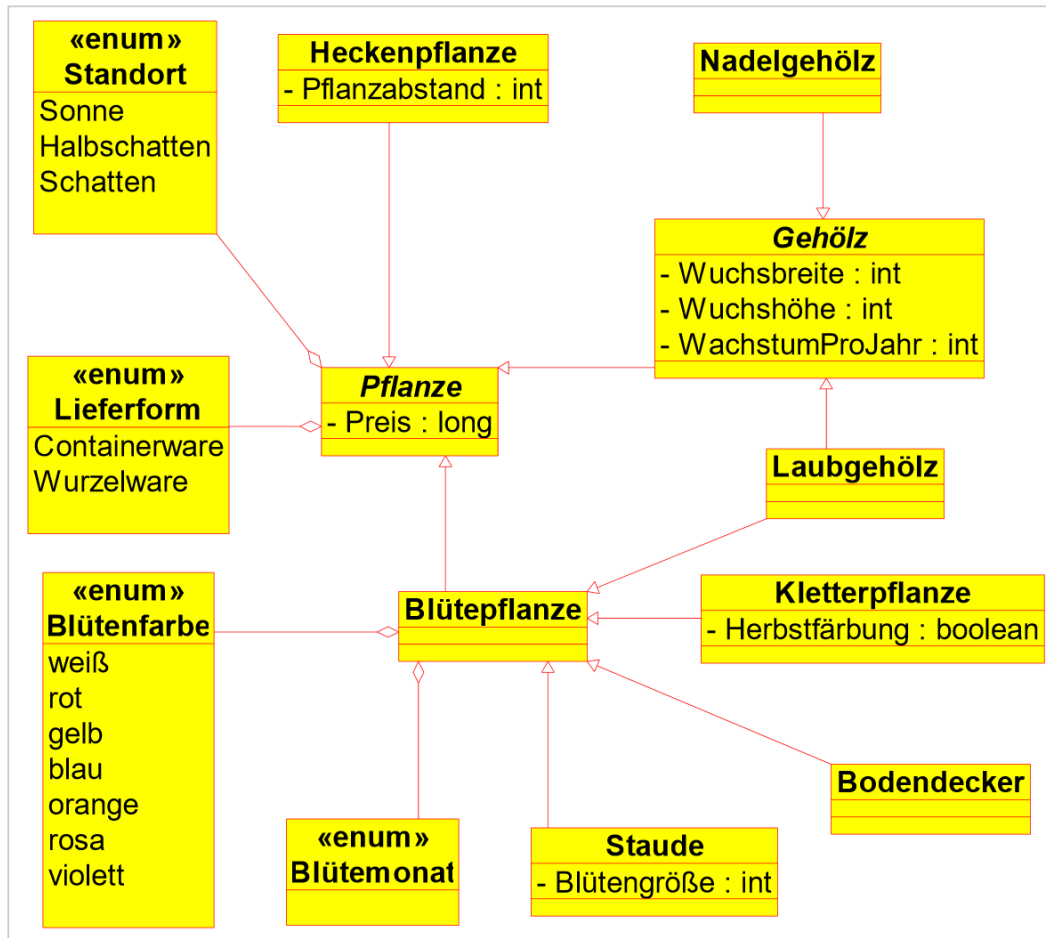
4. Bodendecker werden beschrieben durch:  
Lieferform (Containerware oder Wurzelware)  
Standort (Sonne, Halbschatten oder Schatten)  
Blütenfarbe (weiß, rot, gelb, blau, orange, rosa, violett)  
Blütemonat  
Preis

5. Heckenpflanzen werden beschrieben durch:  
Lieferform (Containerware oder Wurzelware)  
Standort (Sonne, Halbschatten oder Schatten)  
Pflanzabstand in cm  
Preis

6. Stauden werden beschrieben durch:  
Lieferform (Containerware oder Wurzelware)  
Standort (Sonne, Halbschatten oder Schatten)  
Blütenfarbe (weiß, rot, gelb, blau, orange, rosa, violett)  
Blütemonat  
Blütengröße in cm  
Preis

Hinweise:

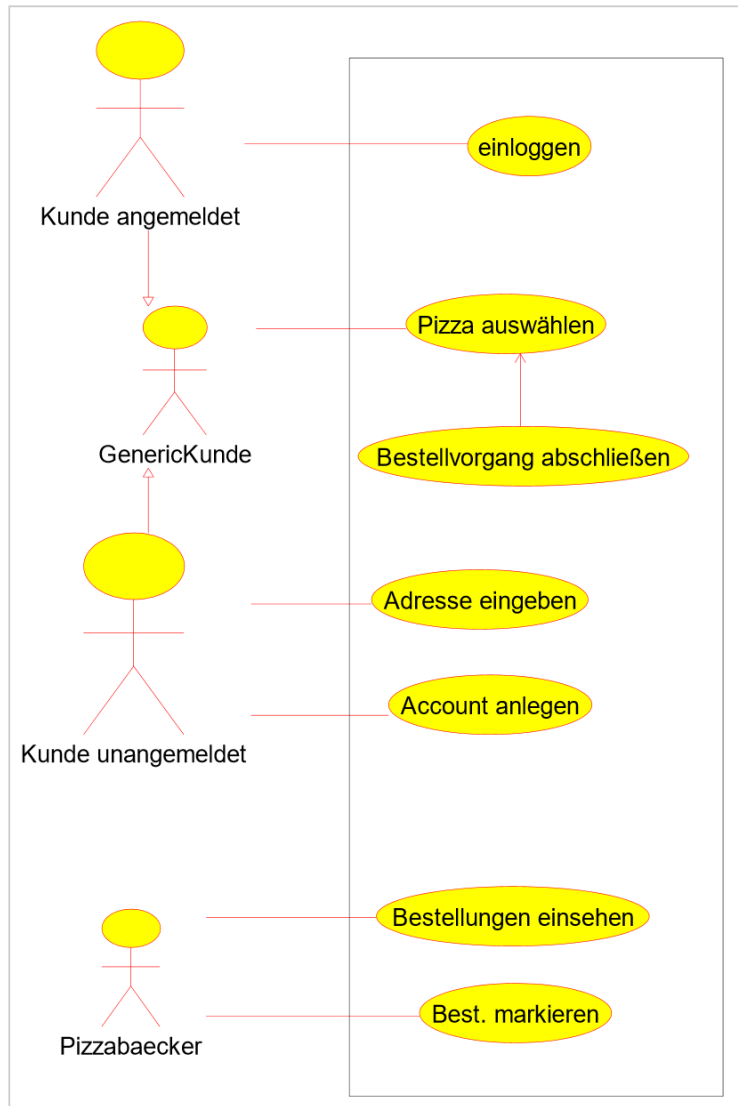
- Entwerfen Sie eine geeignete Vererbungshierarchie.
- Um Missverständnisse zu vermeiden: Kennzeichnen Sie alles farbig was im Diagramm kursiv gedruckt werden soll.
- Methoden brauchen Sie nicht zu modellieren.
- Modellieren Sie alle notwendigen Assoziationen. Die Attribute, mit denen diese Assoziationen implementiert werden, brauchen Sie nicht zu modellieren.



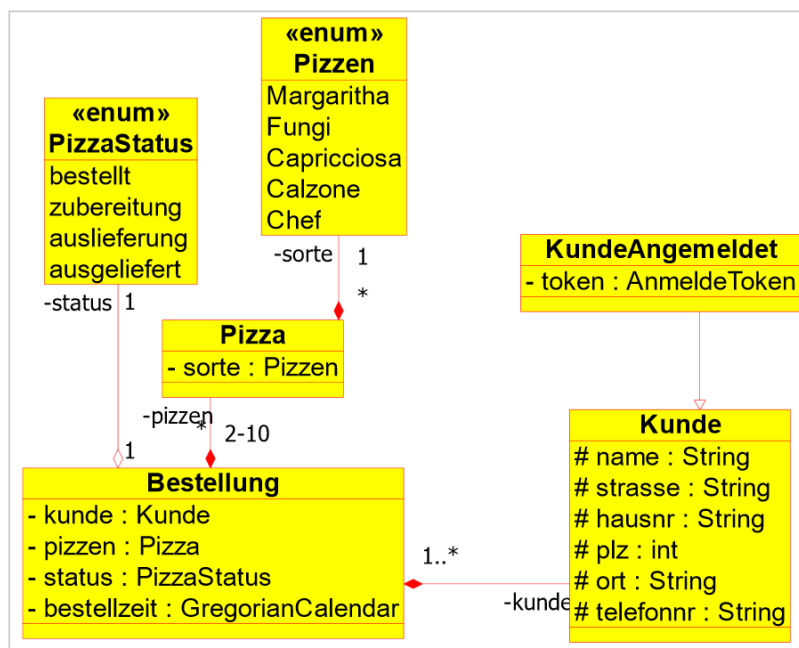
=== Unbekannte Herkunft ===

1. Für eine Pizzadienst in Ludwigshafen soll ein Bestell-Webseite eingerichtet werden. Die Anforderungsanalyse hat ergeben:

- Unangemeldeten Kunden können Pizzen auswählen und bestellen. Dazu müssen sie ihren Namen, Straße, Hausnummer, PLZ, Ort und Telefonnummer angeben
- Eine Bestellung besteht aus zwei bis zehn Pizzen
- Eine Pizza ist einer der Sorten Margaritha, Funghi, Capricciosa, Calzone oder Chef zugeordnet
- Kunden können sich einen Account anlegen. Ihre Daten müssen sie dann nicht immer erneut eingeben, wenn sie bestellen.
- Der Pizzabäcker wird auf seinem Smartphone über eingegangene Bestellungen informiert.
  - Er kann die Preise für verschiedene Pizzasorten pflegen
  - Er markiert jede Bestellung als erledigt, nachdem sie ausgeliefert ist.
- Zeichnen Sie ein Use-Case-Diagramm



b. Zeichnen Sie ein Klassendiagramm mit Klassen und Attributen sowie allen Assoziationen



- c. Die UML beinhaltet 13 [+Profildigramme] Diagrammarten. Die beiden meistverwendeten sind das Klassendiagramm und das Use-Case-Diagramm. Nennen Sie 6 weitere Diagrammarten der UML.
- Sequenzdiagramm
  - Objektdiagramm
  - Komponentendiagramm
  - Zustandsdiagramm
  - Aktivitätsdiagramm
  - Verteilungsdiagramm

## [MO] 2012 SS

Dienstag, 24. Mai 2016

10:17

### 1. Erläutern Sie die folgenden Begriffe

- **Model Driven Architecture**

Konzept der OMG, bei dem die Zielplattform wird bei Modellierung außen vor gelassen wird. Die Entwicklung erfolgt als Plattform independent model (PIM)

- **Generalisierung**

Ist eine Klassendiagramm, bei der Pfeile von einer speziellen auf eine allgemeine Klasse zeigen. Diese Beziehungen können in Generalisierungsgruppen zusammengefasst werden, welche eigene Eigenschaften besitzen können.

Im Use-Case-Diagramm können Generalisierungen zwischen Akteuren und Use Cases auftreten.

- **OMG**

Objects Management Group, Standardisierungs-Gremium der UML

### 2. Modellieren Sie die Funktionalität der "Pause"-Taste |> auf der Fernbedienung eines DVD-Players als UML-Sequenzdiagramm

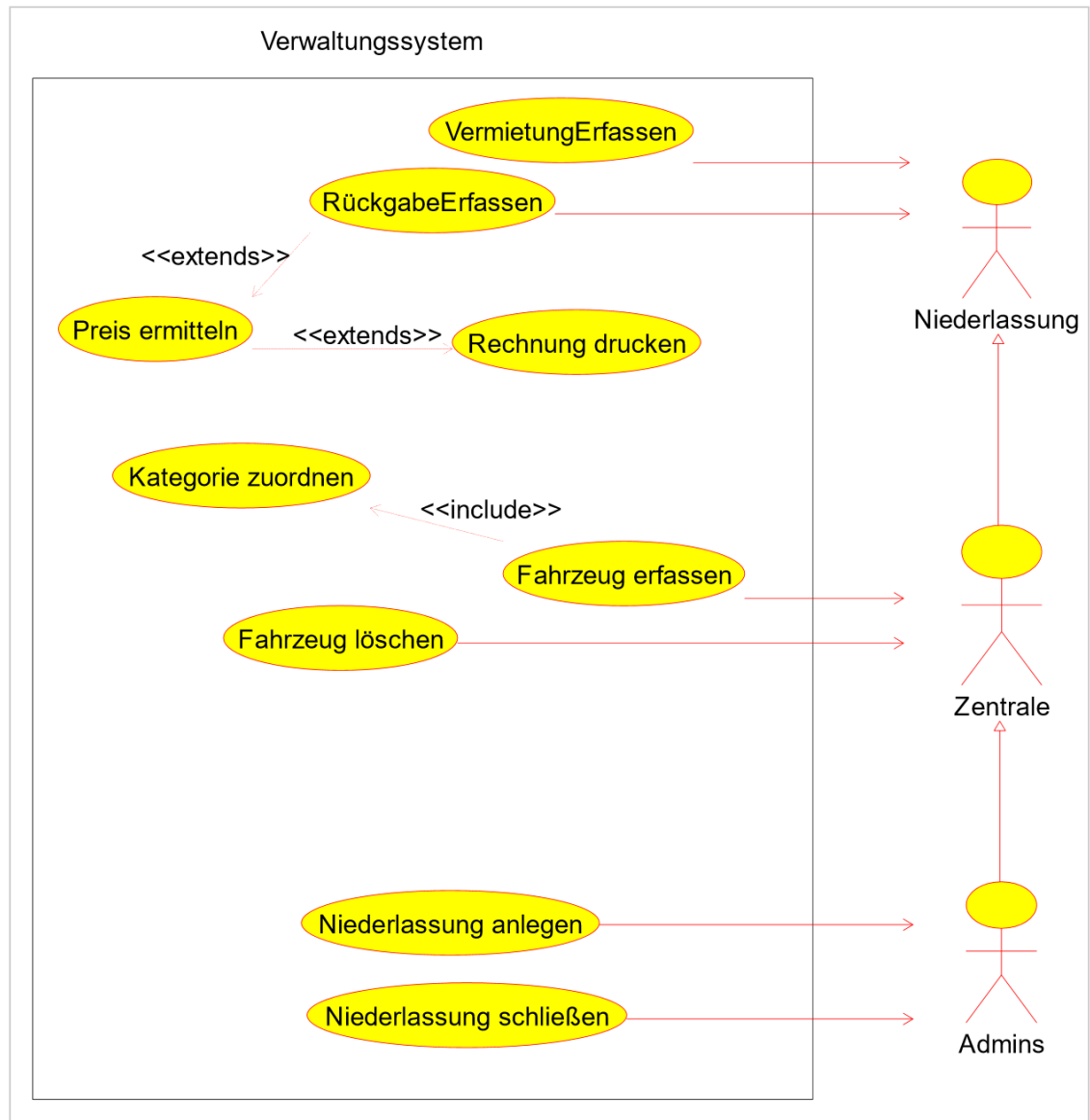
- Lebenslinien für Bediener + Gerät
- Wichtiger Punkt: Asynchrone Kommunikation vs. Synchrone Kommunikation (Ist Synchron)

### 3. Eine Autovermietung möchte ihre Fahrzeuge und Kunden mit einem Client-Server-System verwalten. Die Anforderungsanalyse hat ergeben:

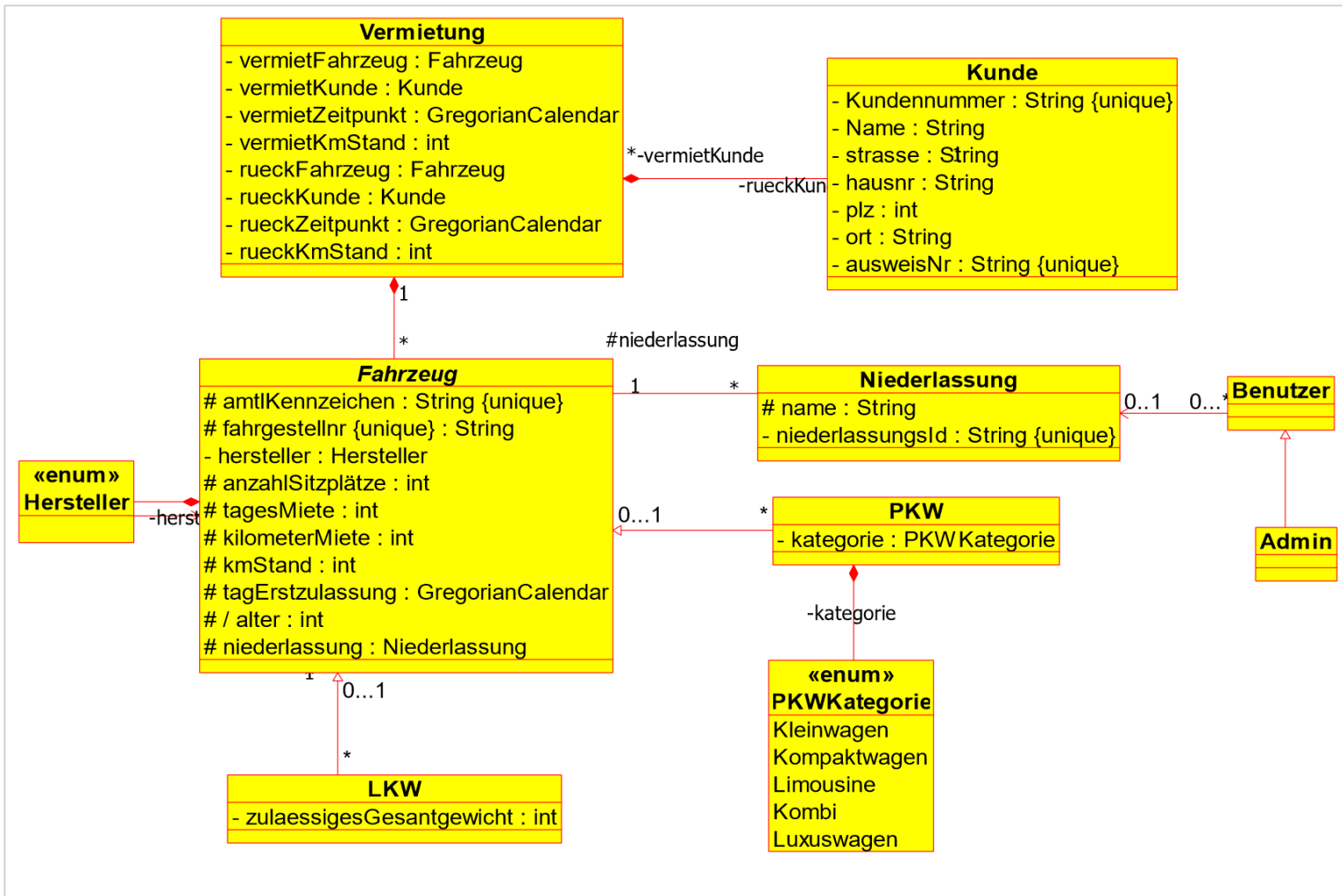
- Die Niederlassungen in verschiedenen Städten sollen Vermietungen und Rückgaben registrieren können.
- Die Zentrale soll zugekaufte Fahrzeuge neu anlegen und verkaufte Fahrzeuge aus dem Bestand löschen können.
- Administratoren sollen neue Niederlassungen anlegen und vorhandene schließen können
- Fahrzeuge können nicht bei einer Niederlassung gemietet und bei einer anderen Niederlassung zurückgegeben werden. Sie sind also eindeutig auf Dauer einer Niederlassung zugeordnet.
- PKW werden beschrieben durch amtliches Kennzeichen, Fahrgestellnummer, Hersteller, Anzahl Sitzplätze, Mietpreis pro Tag, Mietpreis pro Kilometer, Kilometerstand, Tag der Erstzulassung, Alter in Jahren.
- PKW werden einer der Kategorien Kleinwagen, Kompaktwagen, Limousine, Kombi, Luxuswagen zugeordnet.

- LKW werden keiner Kategorie zugeordnet. Für sie wird zusätzlich zu den genannten Attributen das zulässige Gesamtgewicht gespeichert.
- Kunden werden durch eine eindeutige Kundennummer, Namen, Straße, Hausnummer, PLZ, Ort und Personalausweisnummer beschrieben.
- Bei einem Vermietungsvorgang werden das Datum, die Uhrzeit, der Kunde, das Fahrzeug und dessen Kilometerstand erfasst, ebenso bei seiner Rückgabe,
- Nach der Rückgabe kann die Niederlassung sich vom System den Preis anhand von Kilometerstand und Datum ermitteln und die Rechnung drucken lassen.

a. Erstellen Sie ein Use-Case-Diagramm



b. Erstellen Sie ein Klassendiagramm



## [SE] 2012/2013 WS

Mittwoch, 8. Juni 2016

21:49

### 1. Erläutern Sie die folgenden Begriffe

- **Refactoring**

Beim Refactoring wird die Struktur des Quellcodes verändert, ohne dabei dessen Verhalten zu verändern. Ziel ist meist Verbesserung von Stil, Flexibilität und Erweiterbarkeit des Codes.

- **Echtzeitsystem**

Softwaresystem, welches besonderen zeitlichen Anforderungen gerecht werden muss. Um korrekt zu funktionieren muss sowohl Ergebnis als auch die Dauer seiner Berechnung stimmen.

- **WSDL**

WSDL steht Web Services Design Language und definiert, wie ein Service beschrieben wird. Sie spezifiziert Services.

**2. Grenzen Sie sequentielle, hierarchische und hinzufügende Zusammenstellung von Komponenten gegeneinander ab.**

- **sequentiell:** die Teilkomponenten werden nacheinander in der zusammengesetzten Komponente ausgeführt
- **hierarchisch:** eine Komponente ruft die Dienste einer anderen auf
- **hinzufügend:** eine neue Komponente erhält die Schnittstellen aller Teilkomponenten

**3. Erläutern Sie kurz, wie, wann und warum es zur Entstehung der Wissenschaftsdisziplin Software Engineering kam.**

Als Grundstein des *Software Engineering* gilt die NATO-Konferenz 1968, mit dem Ziel, ingenieurwissenschaftliche Methoden auf die Softwareentwicklung anzuwenden. Grund hierfür war die sog. *Softwarekrise*, eine Bezeichnung für mit der Rechenleistung und Komplexität der Software gewachsenen Probleme der Softwareentwicklung und -Methoden.

**4. Aspektorientierte Entwicklung**

- **Erläutern Sie die Vorteile aspektorientierter Entwicklung**  
Aspektorientierte Entwicklung versucht das Problem aufzulösen, dass Separation of Concerns zwar besser für die Wartung eines Systems ist, aber teilweise schwer umzusetzen. Mit aspektorientierter Entwicklung wird entsprechende Funktionalität in Aspekten gekapselt, welche in gängigen Programmiersprachen vor bzw. während der Kompilierung in das Programm eingewoben werden
- **Ein Aspekt besteht aus einem Advice und einem Pointcut, der seine Join Points festlegt. Erläutern Sie diese drei Begriffe.**  
Ein Advice ist der Quellcode, welcher an einem Join Point eingewoben wird. Join Points sind z.B. vor Methoden- oder Konstruktoraufrufen. Die Definition, an welchen Join Points der Code eingefügt werden soll, wird Pointcut genannt.
- **Welche drei grundsätzlichen Möglichkeiten gibt es zum Einweben von Aspekten? Erläutern Sie sie.**  
Die erste Möglichkeit ist die Vorverarbeitung des Quellcodes durch den Aspect-Weaver, welcher vor dem compilieren den Quellcode bearbeitet. Die zweite Möglichkeit ist ein Compiler, welcher während des Compilierens die Aspekte einbaut.  
Die dritte Möglichkeit ist das Ausführen der Advice zur Laufzeit, bei welchem die Join Points im Hintergrund überwacht werden.

**5. Beschreiben Sie die Rolle des Scrum Master.**

Der Scrum Master ist für die Einhaltung der Scrum-Regeln zuständig. Er ist weder Teamleiter noch -Mitglied und ist der Ansprechpartner für Schwierigkeiten der Teammitglieder. Er trägt die im *Daily Scrum* genannten Blockaden in den *Impediment Backlog* ein und ist für deren Beseitigung verantwortlich. Hält er die Ziele eines Sprints für unerreichbar oder die Hürden für zu hoch, kann er den Sprint abbrechen.

**6. Beschreiben Sie ausführlich das Spiralmodell nach Barry Boehm**

Das Spiralmodell ist ein inkrementell-iteratives Modell, welches seine Zyklen in vier Quadranten aufteilt: Zieldefinition, Risikoabschätzung, Implementierung und Test, Planung des nächsten Zyklus.

Hat wie alle inkrementell-iterativen Methoden den Vorteil, dass Fehler früh erkannt werden können und frühe Prototypen zur Verfügung stehen.



Klausur Modellierung WS 2012/2013  
Bearbeitungszeit: 60 Minuten  
Maximale Punktzahl: 60  
Zugelassene Hilfsmittel: Keine

**a. Erläutern sie die folgenden Begriffe (12 Punkte)**

- **Kapselung**

OO-Prinzip, nach dem Informationen vor direktem externen Zugriff abgeschottet werden. Der Zugriff darf - sofern erlaubt- nur über explizit definierte Schnittstellen erfolgen.

Wird bei *Separation of Concerns* eingesetzt, um eine modular unabhängige Struktur zu erreichen.

Kann in Java durch access level modifier, und öffentliche Schnittstellen implementiert werden.

- **n-äre Assoziation**

Beziehung im Klassendiagramm zwischen mehr als zwei Klassen. Wird im Klassendiagramm als Raute dargestellt. Da sich die Multiplizitäten auf Kombinationen mit allen anderen Objekten beziehen ist eine andere Untergrenze als 0 meist nicht sinnvoll.

- **Asynchrone Nachricht**

Sender wartet nicht auf Antwort von Empfänger, als Pfeil im Diagramm gezeichnet. Es gibt keinen Antwortpfeil.

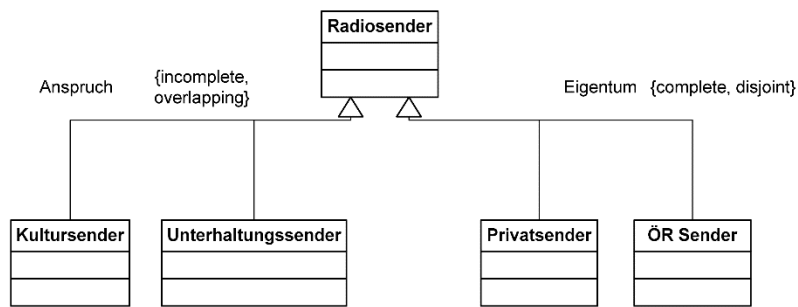


**2. Für einen Schulhausmeister soll ein Softwaresystem zur Verwaltung der Lehrmittel und ihrer Zuordnung zu den Klassenzimmern eingerichtet werden. Zeichnen sie ein Klassendiagramm. Geben Sie dabei auch sinnvolle Multiplizitäten an (23 Punkte).**

Beachten Sie:

- Es gibt vier Arten von Lehrmitteln: Beamer, Overheadprojektoren, Flipchart-Tafeln und Pinnwände (sogenannte Metaplantafeln)
- Ein Klassenzimmer wird eindeutig durch seine Raumnummer beschrieben.
- Ein Lehrmittel kann - muss aber nicht - einem Klassenzimmer zugeordnet sein.
- Ein Lehrmittel wird eindeutig durch seine Inventarnummer beschrieben. Außerdem werden für alle Lehrmittel gespeichert: Modellbezeichnung, Hersteller und Beschaffungsdatum
- Für Beamer wird außerdem die Leuchtstärke gespeichert.
- Für Flipchart-Tafeln werden außerdem die Länge in cm, die Breite in cm und die Papierart gespeichert.
  - Es gibt vier Papierarten: Liniert, Kariert, Blanko und Notenlinien

**3. Betrachten Sie folgenden Ausschnitt aus einem Klassendiagramm:**



Welche der folgenden Aussagen lassen sich aus diesem Diagramm ableiten? Bitte kreuzen Sie das entsprechende Kästchen an, wenn sich die Aussage in der Zeile aus dem Diagramm ableiten lässt. (10 Punkte)

Aussage	lässt sich aus dem Diagramm ableiten
Es gibt keinen Sendern, der nicht entweder Privatsender oder ÖR-Sender ist.	Ja
Es gibt keinen Privatsender, der Kultursender ist.	Nein
Es gibt Sender, die sowohl Privatsender als auch ÖR-Sender sind.	Nein
Es gibt keinen Sender, der sowohl Privatsender als auch ÖR-Sender ist.	Ja
Es kann Sender geben, die weder Kultursender noch Unterhaltungssender sind.	Ja
Es kann Sender geben, die sowohl Kultursender als auch Unterhaltungssender sind.	Ja
Jeder Kultursender ist ein Radiosender.	Ja
Zwischen den Klassen Radiosender und Unterhaltungssender besteht eine Aggregation.	Nein
Jedes Objekt der Klasse Unterhaltungssender hat ein Attribut namens Anspruch.	Nein
Es gibt mindestens einen Privatsender, der Unterhaltungssender ist.	Nein

4. Beschreiben Sie jeweils kurz Gegenstand und Einsatzzweck der folgenden UML-Diagramme: (6 Punkte)

a. Objektdiagramm

Ähnlich einem Klassendiagramm, sind aber Momentaufnahme von Objekten zur Laufzeit. Sie werden v.a. ergänzend zu Klassendiagrammen eingesetzt.

b. Paketdiagramm

Beschreibt die Struktur des modellierten Systems und umfasst die Beziehungen der Pakete.

Paketdiagramme können zur Unterteilung der Software in Module eingesetzt werden.

c. Zustandsdiagramm

Ergänzen v.a. Aktivitätsdiagramme und beschreiben, unter welchen Bedingungen sich ein Objekt ändert. Können zur Definition von Kommunikationsprotokollen eingesetzt werden

5. Zeichnen Sie ein Sequenzdiagramm für das Senden und Annehmen einer Freundschaftsanfrage in Facebook (9 Punkte)

# [SE] 2013

Dienstag, 19. April 2016  
20:33

## 1. Erläutern Sie folgende Begriffe

- **Daily Scrum**  
Tägliches Meeting im Entwicklungsmodell Scrum, bei welchem jedes Teammitglied folgende Themen anspricht: Was habe ich seit gestern erreicht? Was will ich morgen erreichen? Was blockiert meine Arbeit? Die Blockaden werden vom Scrum Master in den Impediment Log eingetragen.
- **P2P**  
In einer Peer-to-Peer-Architektur ist das System dezentral organisiert. Die Rechenoperationen können auf jedem Knoten im Netzwerk durchgeführt werden. Auf jeden Knoten sind die P2P-Protokolle in einer extra installierten Anwendung implementiert. Jeder Knoten ist gleichzeitig auch Switch, der Daten von und zu anderen Knoten leitet.  
Diese Architektur ist hochgradig redundant und ausfalltolerant, aber ineffizient aufgrund Mehrübertragungen.
- **Defensive Programming**  
Aufnahme von Überprüfungsmechanismen in den Code, welche eigentlich für das Funktionieren des Algorithmus nicht erforderlich wären. Es wird sich nicht darauf verlassen, dass Eingaben und Umgebung der Spezifikation entsprechen.
- **Regressionstest**  
Ein Regressionstest ist ein Software-Test, in dem nach Änderung von Code auch bereits getestete Code-Teile erneut getestet werden, um Seiteneffekte, sog. Regressionen, zu finden.
- **Softwarekrise**  
Mitte der 1960er sind Software-Kosten erstmals größer als Hardware-Kosten. Erste Anwendungsprojekte scheitern an Software. Man erkannte, dass die Methoden der Softwareentwicklung nicht mit der Technik schrittgehalten hatten. [ref: Skript/wikipedia]
- **Joint Point**  
Join Points sind Punkte in der Ausführung des Systems, wie z.B. Methodenaufrufe, wo das von Aspekten gelieferte Verhalten kombiniert wird. Ein Join Point ist ein Punkt in der Ausführung des Programms, welcher verwendet wird um die dynamische Struktur eines cross-cutting concern (CCC) zu definieren. [wikipedia]  
--> CCC: Querschnittliche Belange, die durch herkömmliche Modularisierungsansätze nicht einfach modularisiert werden können. [wikipedia]

## 2. Erläutern Sie, wie Entwurfsmuster (Design Patterns) bei der Softwareentwicklung von Nutzen sein können

Design Pattern sind abstrakte Konzepte, welche erfolgreiche Software-Designs beschreiben. Sie können immer wieder verwendet werden und mit Erfahrungen aus jedem weiteren Projekt verbessert werden.

In gängigen Projekten können Design Patterns sehr effektiv sein; es besteht dabei jedoch die Gefahr, durch Verwendung zu vieler Entwurfsmuster ineffektiv zu arbeiten.

## 3. Grenzen Sie die Begriffe Verifikation und Validierung voneinander ab.

### **Verifikation**

Entspricht das Produkt den Anforderungen?

"Are we building the product right?"

Überprüfen, ob die Software ihre funktionalen und nicht-funktionalen Anforderungen erfüllt.

### **Validierung**

Stellt das Produkt die Anwender in ihrer Arbeitsumgebung zufrieden?

"Are we building the right product?"

Sicherstellen, dass die Software die Erwartungen des Kunden erfüllt.

[ref: Sommerville, 2011]

Die Validierung gestaltet sich dabei deutlich schwieriger, da sich Nutzerzufriedenheit und -Bedürfnisse je nach Nutzergruppe stark unterscheiden können.

**4. Als wesentliche Eigenschaft von Frameworks gilt das Prinzip *Inversion of Control*. Erläutern Sie dieses.**

Frameworks agieren wie das eigentliche Hauptprogramm indem sie Methoden der Anwendung aufrufen, was als inversion of control bezeichnet wird.[Skript]

Frameworks reagieren auf Ereignisse und rufen daran angehängte Methoden des Programms auf. Das Programmverhalten wird also vom Framework gesteuert.[Sommerville, 2011]

**5. Erläutern Sie Argumente, die aus Sicht eines anwendenden Unternehmens für den Kauf einer Standardsoftware und die Argumente, die für die Entwicklung einer Individualsoftware sprechen.**

#### **Standardsoftware**

- **Kosten besser abschätzbar**
- **Ausgereifte Software**
- **Prüfung des Systems vor der Entscheidung möglich**
- **Wartungsaufwand abschätzbar**
- **Service, Dokumentation, Schulung, Entwicklung aus einer Hand**
- **Einheitliche GUI über alle Module, Ergonomie**

#### **Individualsoftware**


- Individuell gestaltete und unterstützte Prozesse bringen Vorteile im Wettbewerb
- In der Regel einfachere Legacy-Integration
- Flexibler als Standardsoftware, bessere Abbildung der Prozesse
- Nicht überladen wie Standardprodukte
- Von Performance/Hardwarebedarf her "schlanker"
- Keine aufwendige Anpassung
- Standardlösung oft nicht auf neuestem technischen Stand

# [MO] 2013 SS

Dienstag, 19. April 2016

20:40

## 1. Erläutern Sie folgende Begriffe:

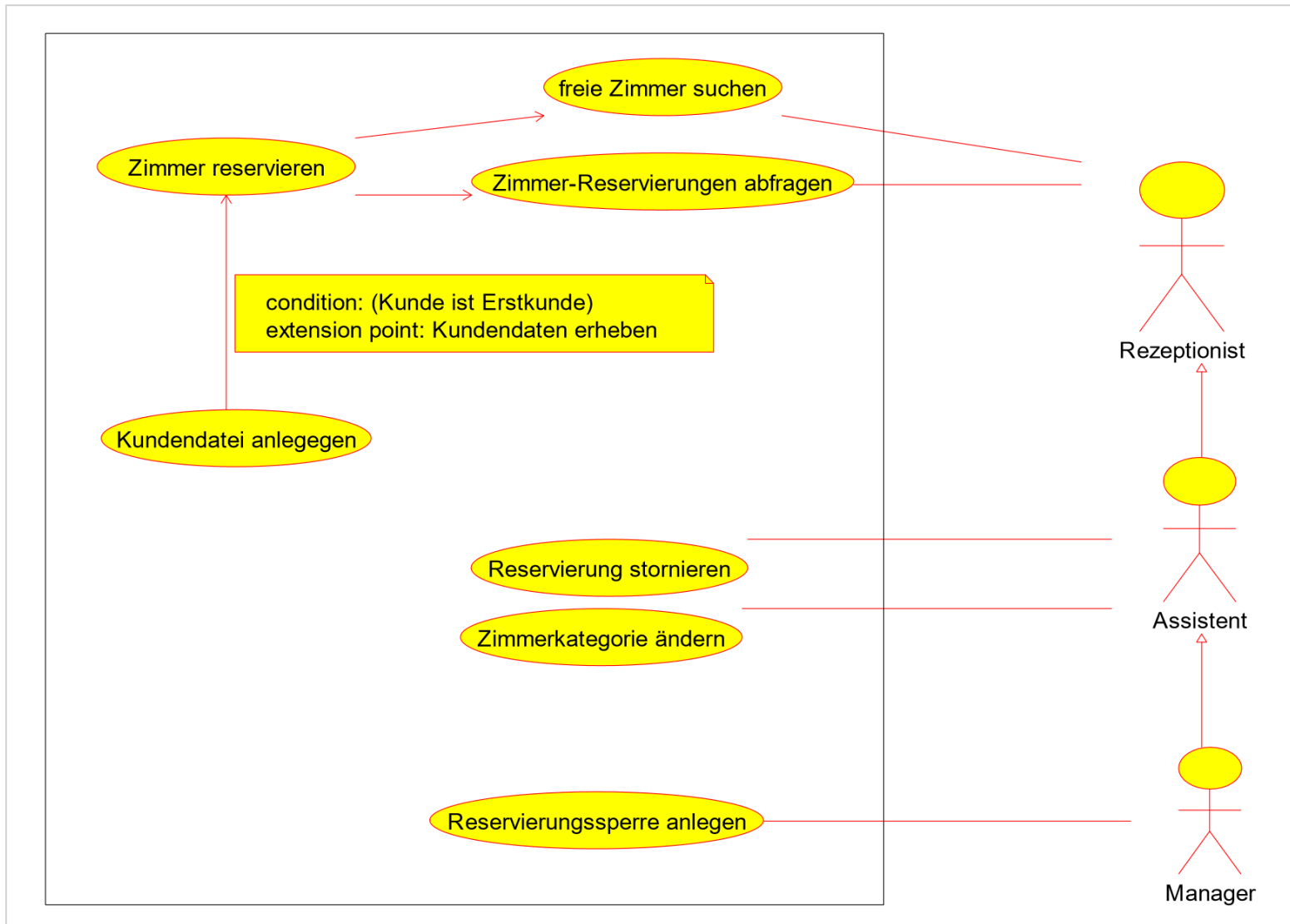
- **Komposition**  
Eine Beziehung zwischen zwei Klassen im Klassendiagramm. Ist interpretierbar als *teil-von*, das Teil kann nicht ohne sein dazugehöriges Objekt existieren.  
Eine Aggregation, bei der ein Objekt untrennbar mit einem Ganzen verbunden ist.  
Wenn das Ganze aufhört zu existieren endet auch die Existenz der Teile.  
Implementierung: Das Ganze und seine Teile sind voneinander unabhängig. Die Klasse des Ganzen erzeugt ihre Teile selbst.  
Java: keine Übergabe als Parameter sondern selbstständig mit new-Operator
- **Entität**  
Entitäten sind Gegenstände aus der Realwelt, welche durch die Datenbank abgebildet werden sollen. Sie werden zu Entitätstypen zusammengefasst.  
Sie werden im Entity-Relationship-Modell als Rechteck mit Bezeichnung des Entitätstyps dargestellt.
- **Asynchrone Nachricht**  
Sender wartet nicht auf Antwort von Empfänger, als Pfeil im Diagramm gezeichnet  
  
Notationselement im Sequenzdiagramm

## 2. Ein Hotel möchte seine Zimmer mit Hilfe eines objektorientiert entwickelten

Anwendungssystem verwalten. Die Anforderungsanalyse hat ergeben:

- Im Augenblick ist keine Reservierung durch Kunden via Internet geplant. Die Applikation soll auf dem Computer an der Rezeption und auf den PCs des Managers und seines Assistenten installiert werden.
- Die Rezeptionisten sollen telefonische oder persönliche Buchungsanfragen sofort beantworten können. Sie müssen also wissen, wie viele Zimmer einer bestimmten Kategorie zu einem bestimmten Termin frei sind.
- Manchen Kunden reicht das nicht. Sie wollen wissen, ob ein bestimmtes Zimmer zu einem Termin noch frei ist. Auch das kann der Rezeptionist in Erfahrung bringen.
- Wenn ein Kunde zum ersten Mal reserviert, werden seine Kundendaten vom Rezeptionisten erhoben.
- Wenn sich der Kunde für die Buchung entscheidet, wird das Zimmer sofort für ihn reserviert.
- Der Rezeptionist kann telefonische Anfragen nach bestimmten Gästen einfach beantworten: Er fragt das System, welches Zimmer sie haben und kann dann dorthin durchstellen.
- Der Assistent des Managers kann alle Aufgaben der Rezeptionisten übernehmen. Außerdem kann er Reservierungen stornieren und Zimmer einer anderen Kategorie zuweisen, wenn z.B. eine Renovierung erforderlich ist.
- Der Manager kann alle Aufgaben der Rezeptionisten und seines Assistenten übernehmen. Außerdem kann er für bestimmte Zimmer eine Reservierungssperre anlegen, so dass diese nicht mehr vergeben werden dürfen.
- Für die Kunden werden hinterlegt: Name, Vorname, Straße, Hausnummer, PLZ, Ort und Land
- Einige Kunden gelten als VIP-Kunden. Für sie kann außerdem hinterlegt werden: Bevorzugte Zimmerkategorie, Geburtstag und ein Freitextfeld für besondere Vorlieben (z.B. das Lieblingsfrühstück)

- Für die Zimmer werden hinterlegt: Zimmernummer, Anzahl der Betten, zugeordnete Kategorie, Größe in m2 und Himmelsrichtung
- Es gibt folgende Zimmerkategorien: Standard, Luxus, Präsidentensuite
- Erstellen Sie ein Use-Case-Diagramm



- b. Erstellen Sie ein Klassendiagramm. Methoden brauchen Sie nicht zu modellieren. Berücksichtigen Sie aber alle nötigen Attribute, Klassen, die Beziehungen zwischen diesen Klassen, Multiplizitäten und ggf. Navigierbarkeiten.

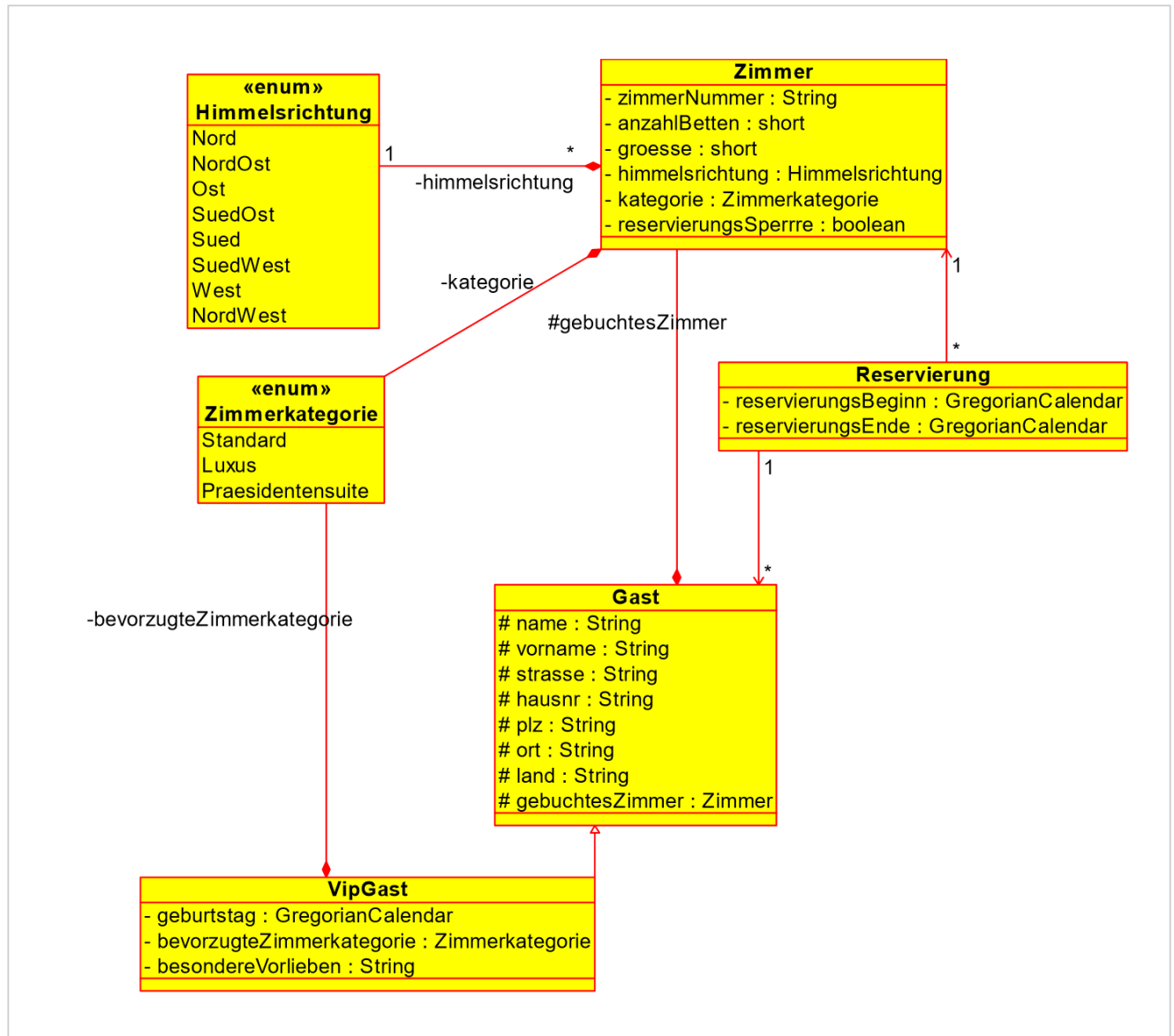
- c. Welche Vor- und Nachteile hat die Modellierung der Zimmerkategorien als Subklasse im Vergleich zur Modellierung als Enumerationsklasse?

**Enum:**

Einfacher und schneller zu implementieren, aber unflexibel für zusätzliche Attribute/Verhalten

**Subklasse:**

Höherer Aufwand, aber dafür möglich, für Zimmertypen zusätzliche Attribute und Methoden zu programmieren





# [MO] 2013/2014 WS

Donnerstag, 12. Mai 2016

11:13

## 1. Erläutern Sie die folgenden Begriffe (12 Punkte)

### a. Modell

Ein Modell ist ein vereinfachtes Abbild der Wirklichkeit, welches deren Komplexität reduziert um für einen bestimmten Kontext relevante Informationen zu liefern.

### b. Aggregation

Eine Form der Assoziation, bei der sich ein Objekt einer Klasse u.a. aus einem Objekt einer anderen Klasse zusammensetzt. Die zusammensetzenden Objekte werden unabhängig erzeugt und können auch wieder entfernt werden.

### c. synchrone Nachricht

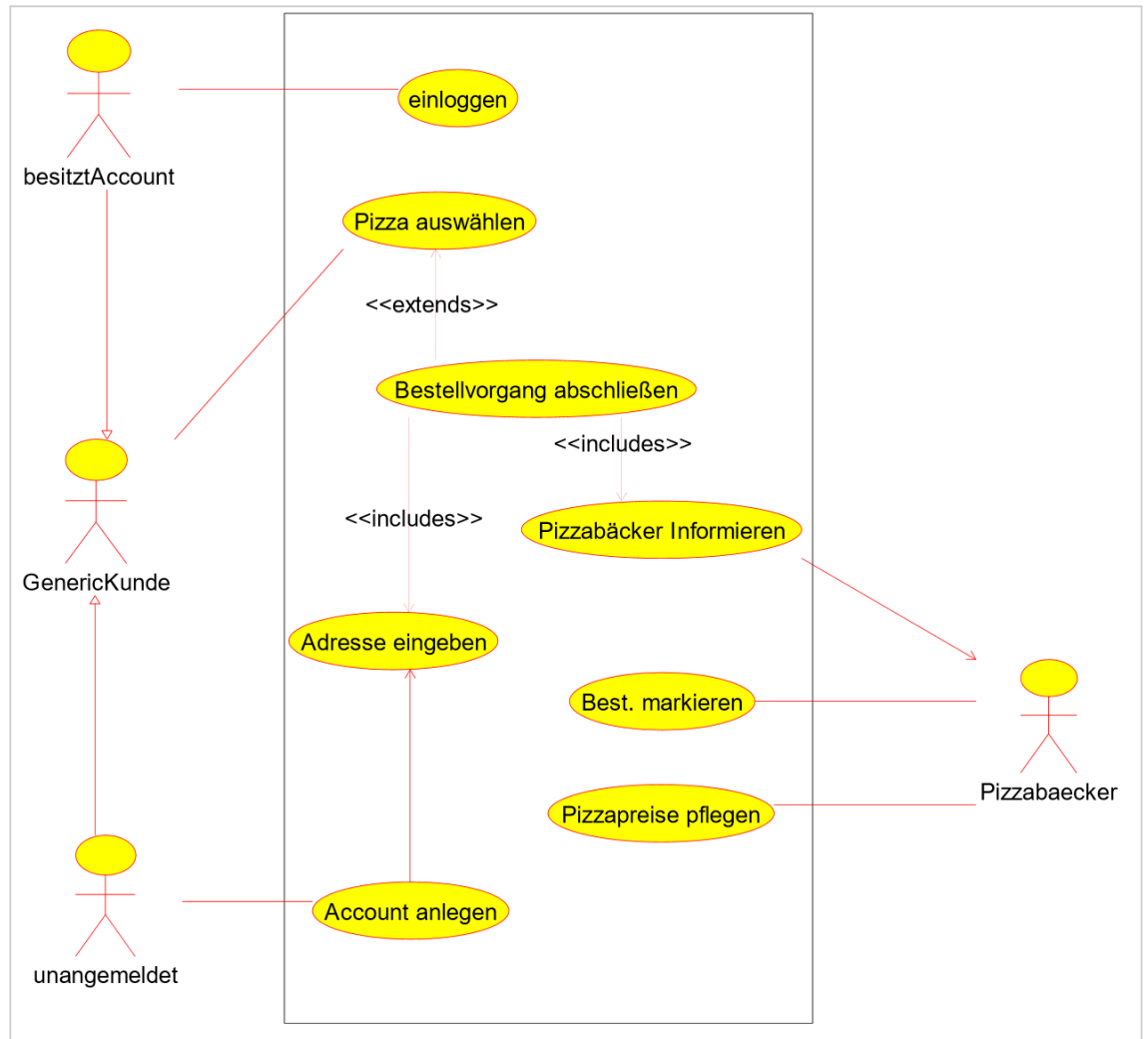
Nachrichtentyp im Sequenzdiagramm, bei dem eine Antwort erforderlich ist. Die synchrone Nachricht wird als Pfeil, deren dazugehörige Antwort mit unterbrochener Linie gezeichnet



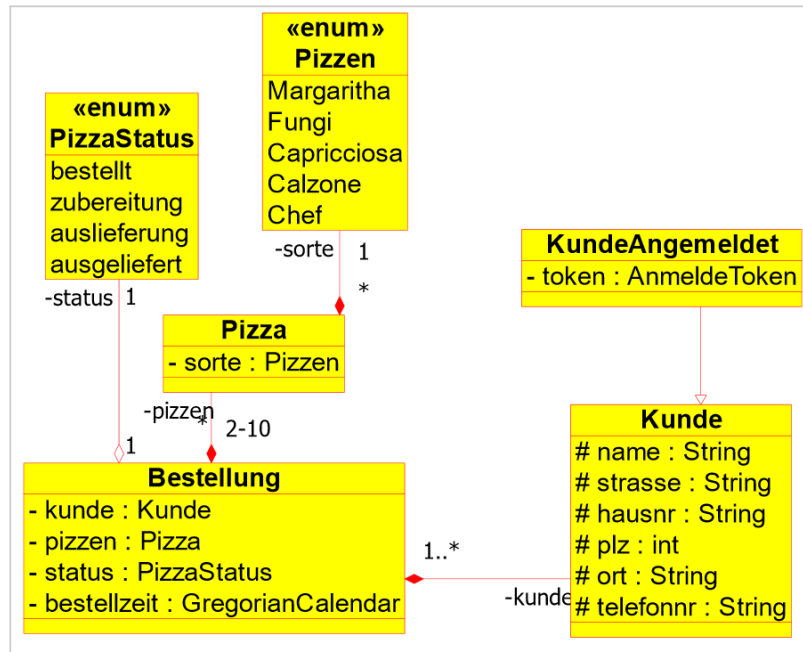
## 2. Zeichnen Sie ein Sequenzdiagramm für das Senden und Annehmen einer Nachricht in Facebook (8 Punkte)

## 3. Für eine Pizzadienst in Ludwigshafen soll ein Bestell-Webseite eingerichtet werden. Die Anforderungsanalyse hat ergeben:

- Unangemeldeten Kunden können Pizzen auswählen und bestellen. Dazu müssen sie Ihren Namen, Straße, Hausnummer, PLZ, Ort und Telefonnummer angeben
- Eine Bestellung besteht aus zwei bis zehn Pizzen
- Eine Pizza ist einer der Sorten Margaritha, Fungi, Capricciosa, Calzone oder Chef zugeordnet
- Kunden können sich einen Account anlegen. Ihre Daten müssen sie dann nicht immer erneut eingeben, wenn sie bestellen.
- Der Pizzabäcker wird auf seinem Smartphone über eingegangene Bestellungen informiert.
  - Er kann die Preise für verschiedene Pizzasorten pflegen
  - Er markiert jede Bestellung als erledigt, nachdem sie ausgeliefert ist.
- a. Zeichnen Sie ein Use-Case-Diagramm



2. Zeichnen Sie ein Klassendiagramm mit Klassen und Attributen sowie falls erforderlich auch mit Assoziationen, Kompositionen, Aggregationen, Generalisierungen, Multiplizitäten und Navigierbarkeiten. Methoden brauchen Sie nicht anzugeben (25 Punkte)



# [SE] 2014/2015 WS

Dienstag, 19. April 2016

15:21

## 1. Erläutern sie folgende Begriffe:

### • **Batchverarbeitende Systeme**

Verarbeiten Daten am Stück ohne Eingriff des Benutzers. Sie eignen sich um ähnliche Operationen auf große Datensätze anzuwenden, wie z.B. der automatisierte tägliche Rechnungsdruck aus einem ERP-System. Sie bestehen meistens aus Komponenten für Ein- und Ausgabe sowie Verarbeitung.

### • **Scrum Master**

Der Scrum-Master ist weder Teamleiter noch Teammitglied und überwacht die Einhaltung der Scrum-Regeln. Er ist der Ansprechpartner für Teammitglieder bei Schwierigkeiten.

Die vorgetragenen Blockaden aus dem Daily Scrum werden vom S. in den Impediment Log eingetragen. Der S. ist ebenfalls für deren Beseitigung verantwortlich.

Der S. kann Sprints abbrechen, wenn er die Ziele für unerreichbar oder Hürden für unüberwindbar hält.

### • **Defensive Programming**

Aufnahme von Überprüfungsmechanismen in den Code, die eigentlich für das Funktionieren des Algorithmus nicht erforderlich sind. Es wird sich nicht darauf verlassen, dass Eingaben und Umgebung der Spezifikation entsprechen.

### • **Betatest**

Softwaretest, bei dem eine erste Version einer COTS-Software an einen (meist eingeschränkten) Benutzerkreis ausgegeben wird, um Feedback zu sammeln. Der Betatest folgt auf den Alphatest. Danach folgen Release Candidate, Release to Manufacturing und General Availability als Letzte Phase.

--> COTS: Commercial off-the-shelf software, also Standardsoftware

### • **Struts**

Quelloffene Implementierung von MVC für EE. Seit 2000 unter Apache Foundation entwickelt. Weit verbreitet, Weiterentwicklung von Version 1 eingestellt, dafür Nachfolgeversion 2. Zusätzliche Funktionen zur Internationalisierung, Eingabevalidierung und Fehlerbehandlung.

### • **Stakeholder**

Bei Scrum: Stakeholder sind die Kunden und finanzieren das Projekt. Sie bekommen im Sprint Review Meeting die Ergebnisse des letzten Sprints präsentiert.

Allgemein SE: Im Allgemeinen sind Stakeholder eines Softwareprojekts oder Systems diejenigen Personen, Personengruppen oder Organisationen, welche ein Interesse am System besitzen. Dies schließt Benutzer des Systems mit ein.

## 2. Begründen Sie, für welche der folgenden Softwareprojekte Sie aspektorientierte Entwicklung empfehlen, für welche und warum:

### a. Das Kontoführungssystem einer Bank

### b. Eine App für Mensch-Ärgere-Dich-Nicht unter Android

### c. Eine Software, die an Touchscreens im Kassensbereich Eintrittskarten für ein Hallenbad verkauft

## 3. Client/Server-Systeme

### a. Beschreiben Sie verbal und mit einer Skizze, was man unter einem 3-Tier-Client-Server-System versteht

In einem dreistufigen System sind Datenmanagement, Anwendungsprozesse und Client getrennt.

Ein Datenbankserver sorgt für Persistenz, Applikationsserver führen Anwendungslogik aus und Clients übernehmen Ein/Ausgabe sowie Darstellung.

[ Client ] <---> [ Anwendungs-Server ] <---> [ Datenmanagement-Server ]

**b. Welche Vorteile hat ein 3-Tier-Client-Server-System?**

- einfachere Skalierbarkeit bei vielen Anwendern durch mehr Anwendungsserver
- keine Aufspaltung von Anwendungslogik zwischen Datenbankserver und Anwendung
- einfachere Integration von Daten aus unterschiedlichen Quellen

**4. Beschreiben Sie Eigenschaften, Vorteile und Nachteile des Wasserfallmodells [10 Min.]**

Das Wasserfallmodell ist ein streng sequentielles Entwicklungsmodell, in welchem die Phasen Anforderung, Design, Implementierung, Test, Inbetriebnahme hintereinander durchlaufen werden.

Vorteilhaft ist die unmittelbare Verständlichkeit und Qualitätskontrolle an jedem Phasenende. Nachteile sind der Mangel an Prototypen und damit die Möglichkeit, vor Fertigstellung zu testen, die Annahme, dass die Anforderungen im Detail klar sind und sich nicht mehr ändern und das Fehlen des iterativen Entwicklungsprozesses.

Es ist deshalb v.a. für kleine, interne Projekte geeignet, bei denen weder mit Änderung der Anforderungen noch größeren Missverständnissen zu rechnen ist.

**5. Ergänzen Sie folgende Tabelle mit Arten der Benutzerinteraktion - Stichpunkte genügen**

Art der Interaktion	Vorteil	Nachteil	Anwendungsbeispiel
Direkte Manipulation	• schnell + intuitiv	• ggf. schwierige Implementierung	• PC-Spiel
Menüauswahl	• verhindert Tippfehler	• langsam für erfahrene Nutzer	• allg. Systeme
Ausfüllen einer Eingabemaske	• einfache Dateneingabe	• großer Platzbedarf auf Bildschirm	• Bestandskontrolle
Befehlssprache	• leistungsfähig	• hoher Lernaufwand	• Betriebssysteme

**6. Eine städtische Musikschule möchte ihre Schüler und Lehrer mit Hilfe eines Objektorientiert entwickelten Anwendungssystems verwalten.**

Die Anforderungsanalyse hat ergeben:

- Ein Schüler kann für ein bis maximal zwei Instrumente Unterricht erhalten.
- Ein Lehrer unterrichtet genau ein Instrument
- Lehrer können festangestellte Lehrer oder Honorarkräfte sein
- Festangestellte Lehrer haben zwischen 5 und 20 Schüler
- Honorarkräfte haben 2 bis 10 Schüler
- Es gibt ausschließlich Einzelunterricht, d.h. jede Unterrichtseinheit besteht aus einem Lehrer, einem Instrument und einem Schüler. Sie wird durch Wochentag, Anfangszeit und Dauer beschrieben.
- Schüler werden beschrieben durch Name, Vorname und eine eindeutige Schülernummer.
- Festangestellte Lehrer werden beschrieben durch Name, Vorname, eindeutige Personalnummer und Monatsgehalt.

- Honorarkräfte werden beschrieben durch Name, Vorname, eindeutige Personalnummer und Stundenhonorar.
- Die Schüler zahlen einen festen Monatsbeitrag, der für jedes Instrument festgelegt ist, unabhängig davon, welcher Lehrer sie unterrichtet.
- Die Sekretärin kann neue Schüler aufnehmen, ausscheidende Schüler löschen und die Daten von Schülern pflegen.
- Sie kann Unterrichtseinheiten planen und dafür Schüler, Instrument, Lehrer, Wochentag und Uhrzeit einpflegen.
- Der Leiter der Musikschule kann all das und außerdem Lehrer einstellen, entlassen und Ihre Daten pflegen. Außerdem kann er sich die Gesamtzahl der Schüler anzeigen lassen.
- Für die Administration des Systems wird ein Fachmann der Stadtverwaltung mit 10% seiner Arbeitszeit betraut. Er kann alle Funktionen nutzen und auch Instrumente anlegen und löschen sowie die für die Instrumente zu zahlenden Monatsbeiträge ändern.
- Ein Berater schlägt vor, das System nach dem Prinzip der Diversity im *n-Version-Programming* zu entwickeln. Begründen Sie ihre Meinung dazu.

Eine Entwicklung nach n-Version-Programming würde den Aufwand in der Entwicklung je nach Versionsanzahl erheblich erhöhen und es damit erschweren, zu konkurrenzfähigen Preisen zu entwickeln.

Zusätzlich wäre der Nutzen von n-version-Programming auf Schutz vor Implementationsfehlern bzw. auf Redundanz begrenzt, nicht aber auf Spezifikationsfehler.

Dieser erhöhte Aufwand ist nicht zu rechtfertigen, da kurzzeitige Ausfälle des Systems wahrscheinlich ohne Bedrohung der Geschäftsgrundlage überstanden werden können.

- Er schlägt eine zweistufige Client/Server-Architektur für das System vor. Begründen Sie ihre Meinung dazu

Eine zweistufige Client-Server-Architektur wäre hier durchaus sinnvoll, bei der die Daten auf einem Datenbankserver gespeichert werden und die Anwendung zur Verwaltung der Daten auf den Clients selbst läuft.

Als Alternative zu Fat Clients könnte die Anwendung auch als Webapplikation entwickelt werden.

# [MO] 2014/2015 WS

Montag, 25. April 2016  
10:52

1. Erläutern Sie die folgenden Begriffe und zeichnen Sie die dazugehörigen Symbole aus der UML.
  - **Komposition**  
Eine Aggregation, bei der ein Objekt untrennbar mit einem Ganzen verbunden ist. Die einzelnen Teile sind voneinander abhängig. Die Klasse des Ganzen erzeugt ihre Teile selbst. Wenn das Ganze aufhört zu existieren endet auch die Existenz seiner Teile.
  - **Aggregation**  
Eine Form der Assoziation, bei der sich ein Objekt der einen Klasse u.a. aus einem (oder mehreren) Objekten einer anderen Klasse zusammensetzt. Das Ganze und seine Teile sind voneinander unabhängig. Die Klasse erzeugt ihre Teile nicht selbst, sie können auch entfernt werden.
  - **Gabelung (im Aktivitätsdiagramm)**  
Eine Gabelung ist ein Element im Aktivitätsdiagramm, bei der der Kontrollfluss in mehrere parallele Kontrollflüsse ausgespalten wird.
2. Ein Fußballverein möchte seine Mitglieder mit Hilfe eines objektorientiert entwickelten Anwendungssystems verwalten.  
Die Anforderungsanalyse hat ergeben: [48 Min.]
  - Es gibt mehrere Mannschaften (A-, B-, C-, D-Jugend, Herren, Damen, AH)
  - Mitglieder können aktive Mitglieder, passive Mitglieder oder Jugendliche sein.
  - Ein aktives Mitglied gehört zu einer Mannschaft.
  - Ebenso gehört ein jugendliches Mitglied zu genau einer Jugendmannschaft.
  - Für jede Mannschaft gibt es genau einen Trainer. Trainer sind immer auch Mitglieder des Vereins.
  - Mitglieder werden beschrieben durch Name, Vorname und eindeutige Mitgliedsnummer.
  - Trainer werden beschrieben durch Name, Vorname, eindeutige Mitgliedsnummer und monatliches Entgelt.
  - Mannschaften werden beschrieben durch Bezeichnung, einen Tabellenplatz und den Namen der Liga, in der sie spielen.
  - Die Mitglieder zahlen einen festen Monatsbeitrag. Dieser unterscheidet sich für aktive Mitglieder, passive Mitglieder und Jugendliche.
  - Die Sekretärin kann neue Mitglieder aufnehmen, ausscheidende Mitglieder löschen und die Daten von Mitgliedern pflegen.
  - Sie kann Trainingseinheiten planen und dafür Mannschaft, Wochentag und Uhrzeit einpflegen. Eine Mannschaft hat 1 bis 4 wöchentliche Trainingseinheiten.
  - Der Vereinsvorstand kann all das und außerdem Mannschaften anlegen, Mannschaften löschen, einen Trainer einer Mannschaft zuweisen, einem Trainer sein Amt entziehen und sein monatliches Entgelt pflegen. Außerdem kann er sich die Gesamtzahl der Mitglieder anzeigen lassen.
  - Der Jugendleiter kann alle Funktionen des Vereinsvorsitzenden nutzen, aber nur für Jugendliche, Jugendmannschaften und deren Trainer.  
  - Spezifizieren Sie den Use Case *Neues Mitglied aufnehmen* in einem Use-Case-Dokument. Berücksichtigen Sie dabei alle wichtigen Inhalten, die in einem Use-Case-Dokument enthalten sein sollten.[12 Min.]

**Use Case:** Neues Mitglied aufnehmen

**Description:** Aufnahme eines neuen Mitglieds im Fußballverein-Mitglieder-Verwaltungs-Anwendungssystem (nachfolgend FVMVAS) durch berechnigte Benutzergruppe

**Actors:** berechtigter Benutzer

**Basic Flow**

1. Nutzer betätigt Schaltfläche "neues Mitglied aufnehmen"
2. Nutzer erhält Eingabemaske mit Eingabefelder für folgende Daten:
  - **Name und Vorname (freie Texteingabe)**
  - **Adresse (Auswahl aus Liste, alternativ: freie Texteingabe)**
  - **Art der Mitgliedschaft (Auswahl aus Liste)**
  - **Wenn aktiv: Mannschaft (Auswahl aus Liste)**
  - **Wenn jugendlicher: Jugendmannschaft (Auswahl aus Liste)**
3. System auto-speichert Eingaben während dem Eingeben
4. Benutzer bestätigt Eingaben durch Betätigung einer Schaltfläche
5. System Überprüft und speichert Daten
6. 6a) Daten in Ordnung: System schließt Eingabemaske und gibt Erfolgsmeldung aus  
6b) Fehler in Daten: System gibt Fehlermeldung aus und markiert fehlerhafte Eingabefelder  
6c) Anderer Fehler: System gibt Fehlermeldung aus, Versucht Datenverlust zu vermeiden und Fehlerbericht an Admin zu schicken

**Alternate Flow**

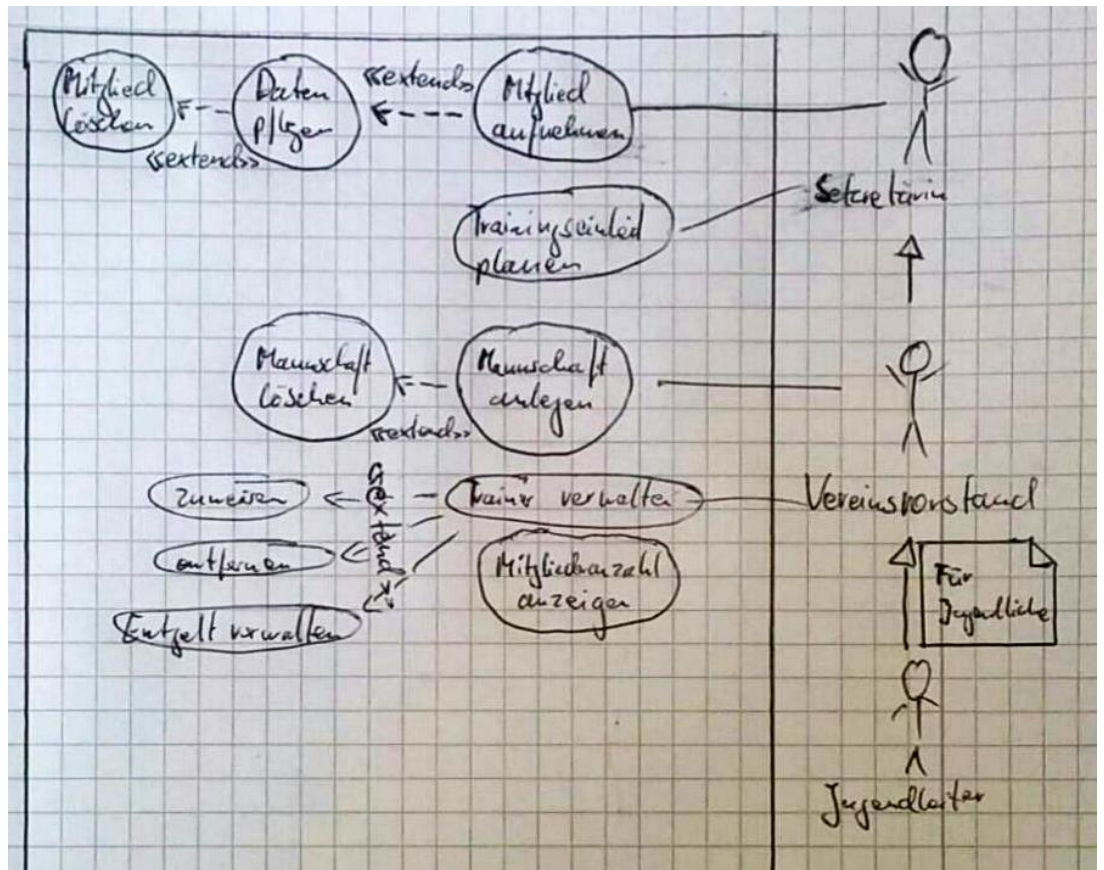
**Preconditions:**

- System betriebsbereit
- berechtigter Benutzer anwesend und besitzt Daten des aufzunehmenden Mitglieds bzw. kann diese unmittelbar in Erfahrung bringen
- Berechtigter Benutzer hat Benutzeroberfläche des FVMVAS geöffnet

**Postconditions:** Mitglied wurde ins FVMVAS aufgenommen

**2. Zeichnen Sie ein Use-Case-Diagramm des Systems [12 Min.]**





- c. Zeichnen Sie ein Klassendiagramm für das System. [24 Min.]  
 Hinweis: Modellieren Sie alle notwendigen Attribute. Methoden brauchen Sie nicht zu modellieren.

# [SE/MO] 2015 SS

Dienstag, 19. April 2016

13:56

Erläutern die folgenden Begriffe

- **Regressionstest [SE]**  
Wiederholung von Tests in bereits getesteten Softwareteilen um weitere Fehler durch Codeveränderungen auszuschließen
- **Komposition [MO]**  
Eine Aggregation, bei der ein Objekt untrennbar mit dem Ganzen verbunden ist. Wenn das Ganze aufhört zu existieren endet auch die Existenz der Teile.
- **Software Reengineering [SE]**  
Neuimplementierung eines bestehenden Systems zur Verbesserung der Wartbarkeit. Die Funktionalität bleibt gleich, Architektur und Dokumentation werden jedoch neu erstellt; ggf. auch Wechsel der Programmiersprache.  
Vorteil: geringere Kosten und Risiko als bei vollständiger Neuentwicklung. Meist aber Inkaufnahme höher Kosten um durch vollständige Neuentwicklung auch neue Funktionen berücksichtigen zu können.
- **Defensive Programming [SE]**  
Aufnahme von Überprüfungen in den Code, welche eigentlich nicht für das Funktionieren des Algorithmus notwendig wären. Es wird sich nicht darauf verlassen, dass Umgebung/Eingaben genau der Spezifikation entsprechen. ->wikipedia
- **Struts [SE]**  
Framework, welches Model-View-Controller (MVC) umsetzt. Seit 2000 unter Apache, weit verbreitet. Weitere Funktionalität in Bereich Internationalisierung, Eingabevalidierung, Fehlerbehandlung
- **SOAP**  
**AOSD**  
Aspektororientierte Entwicklung. Mittels Aspekten wird Funktionalität gekapselt. Im Quellcode wird spezifiziert, wie Aspekte in den Code eingewoben werden, um ein ausführbares Programm zu erzeugen.  
**SOA**  
Service-orientierte Architektur ist eine spezielle Art der Wiederverwendung von Komponenten mittlerer Größe. Die Komponenten sind eigenständige Dienste, die im Netzwerk verteilt ausgeführt werden  
**SOAP**  
Simple Object Access Protocol, definiert das XML-Format der Nachrichten, die ein Service enthält und verschickt.
- **Nennen Sie drei Rollen eines SCRUM-Projekts und deren Aufgaben [SE]**  
*Manager:* Sind Vorgesetzte des Teams, stellen Arbeitsumgebung zur Verfügung  
*Stakeholders:* finanzieren das Projekt, bekommen im Sprint Review die Ergebnisse des letzten Spring präsentiert  
*Das Projektteam:* schätzt Aufwände für Einträge im Product Backlog, wählt anhand der Prioritäten des Product Owners aus den Product Backlog-Einträgen die Ziele für den nächsten Sprint
- **Was ist ein Sprint? Wie lange dauert es? [SE]**  
Eine etwa ein Monat lange Entwicklungsphase. Die Ziele werden im Sprint Planning Meeting festgelegt. Es wird ein Sprint Backlog angelegt, welcher eine Auswahl der Ziele aus dem Product Backlog enthält. Im Sprint Review Meeting werden die Ergebnisse vorgestellt.
- **Was ist ein Scrum? Wie lange dauert es? [SE]**  
Im Daily Scrum beantwortet jedes Mitglied des Projektteams die Fragen: Was habe ich seit gestern erreicht? Was will ich morgen erreichen? Was blockiert meine Arbeit? Die Blockaden werden vom Scrum Master in den Impediment Log eingetragen.

- **Systemarchitektur [SE]**

Architekturentscheidungen bestimmen die Struktur des Systems. Die gewählte Systemarchitektur beeinflusst:

- Die System-Performance
- Die System-Angriffssicherheit (Security)
- Die System-Betriebssicherheit (Safety)
- Die System-Verfügbarkeit
- Die System-Wartbarkeit

- **Was sollen Entwickler bei Scrum zur Sprache bringen? [SE]**

Im Daily Scrum beantwortet jedes Mitglied des Projektteams die Fragen: Was habe ich seit gestern erreicht? Was will ich morgen erreichen? Was blockiert meine Arbeit?

Aufgabe ...

Dr. Carsten Dorrhauer

Klausur Modellierung und Software Engineering SS 2015

Die Fernbuslinie möchte eine objektorientierte Software entwickeln lassen, um ihre Linien zu verwalten. (75 Punkte)

Die Anforderungsanalyse hat ergeben:

- In allen angefahrenen Städten gibt es genau eine Haltestelle,
- Eine Buslinie startet in einer Stadt, endet in eine Stadt und fährt dazwischen 3 bis 10 weitere Städte an.
- Jede Buslinie fährt an mindestens einem Wochentag. Manche Buslinien fahren aber auch an mehreren Wochentagen, manche an allen sieben Wochentagen.
- Eine Buslinie wird beschrieben durch ihre Streckenlänge und die angefahrenen Städte.
- Sie hat eine eindeutige Nummer.
- Eine Busfahrt ist eindeutig einer der Buslinien zugeordnet und fährt an einem bestimmten Datum. Sie wird durch dieses Datum beschrieben. Sie kann einen der drei Status „noch nicht gestartet“, „unterwegs“ und „am Ziel angekommen“ haben.
- Jeder Busfahrt ist einer der Fahrer zugeordnet.
- Zu jedem Fahrer kennt das System Name, Vorname, Geburtsdatum, Geburtsort und eine eindeutige Personalnummer.
- Die Busse sind alle baugleich und haben gleich viele Sitzplätze.
- Ein Ticket ist eindeutig einer Busfahrt zugeordnet. Es gilt freie Platzwahl.
- Die Mitarbeiter in den Verkaufsstellen können online Tickets buchen, solange für die Busfahrt noch Plätze verfügbar sind. Sie können auch Tickets stornieren.
- Die Mitarbeiter in der Zentrale können darüber hinaus neue Linien anlegen, Linien bearbeiten und löschen, neue Städte anlegen, Städte bearbeiten und löschen, neue Fahrer anlegen, Fahrerdaten bearbeiten und Fahrer löschen sowie die Fahrer einer Fahrt zuweisen.
- Die Fahrer ändern mobil nach dem Start und nach der Ankunft den Status ihrer Fahrt.

a. **Spezifizieren Sie den Use-Case *Einen Fahrer einer Busfahrt zuordnen* in einem Use-Case-Dokument. berücksichtigen Sie dabei alle wichtigen Inhalte, die in einem Use-Case-Dokument enthalten sein sollten.**

Use Case: Einen Fahrer einer Busfahrt zuordnen

Beschreibung: Mitarbeiter weist Fahrer einer Busfahrt zu

Vorbedingungen: Mitarbeiter ist am Omnibusverwaltungssystem angemeldet und hat die Benutzeroberfläche geöffnet.

Ablauf im Normalfall:

1. Mitarbeiter klickt auf Schaltfläche 'Fahrten'. Eine Liste der Fahrten wird angezeigt.
2. Mitarbeiter öffnet einen Eintrag mit Doppelklick. Die Details der Fahrt werden angezeigt.

3. Mitarbeiter klickt auf Schaltfläche 'Fahrer bearbeiten'. Eine Liste der dieser Fahrt zugewiesenen Fahrer und eine Liste aller Fahrer (scrollbar) wird angezeigt. Über den Listen befindet sich jeweils ein Suchfeld, um schnell Fahrer zu finden.
4. Der Mitarbeiter klickt auf einem Eintrag in der allgemeinen Liste auf die Schaltfläche 'dieser Fahrt zuweisen'. Der Mitarbeiter wird (sofern nicht bereits der Fahrt zugewiesen) der Fahrt zugewiesen und erscheint nun in der 'zugewiesen-Liste'

Alternativablauf:

4a Mitarbeiter hat versehentlich falschen Mitarbeiter zugewiesen. Mit einem Klick auf 'von Fahrt entfernen' auf den Entsprechenden Eintrag in der Liste kann dies rückgängig gemacht werden.

Postconditions: Fahrer wurde Fahrt zugewiesen

**b. Zeichnen Sie ein Use-Case-Diagramm für das neue System**

**c. Zeichnen Sie ein Klassendiagramm für das System**

Hinweise:

- **Modellieren Sie alle notwendigen Attribute. Methoden brauchen Sie nicht zu modellieren.**

**d. Begründen Sie, welche Systemarchitektur Sie für das System vorschlagen**

Ich würde eine 2-Tier-Client-Server-Architektur vorschlagen, bei welcher Thin Clients auf einen zentralen Server zugreifen, welcher Datenbank + Anwendungslogik beinhaltet. Das vereinfacht Datenschutz, Datensicherung und Zugriffskontrolle.

# [SE/MO] 2015 WS 2016

Donnerstag, 23. Juni 2016

15:45

## 1. Erläutern Sie die folgenden Begriffe

- **Timingdiagramm**

Beschreiben die Kommunikation zwischen Objekten und betonen dabei den zeitlichen Ablauf der Kommunikation.

- **Swimmlane**

Swimlanes sind eine Notationselement im Aktivitätsdiagramm und stellen die organisationalen Zuständigkeiten dar. Sie werden graphisch meist als Spalte für jeden zuständigen Akteur dargestellt.

- **Schnittstelle**

Stellen im Allgemeinen detaillierte Vorgaben über die Interaktion zwischen einzelnen Komponenten dar.

Im Klassendiagramm bezeichnen Schnittstellen Vorgaben über Attribute und Methoden einzelner Klassen

- **Call-by-Value**

Ist eine Übergabeart für Methodenparameter, bei welcher der Übergebene Wert eine Kopie des Originalwerts ist. Je nach Programmiersprache ist diese Übergabeart für unterschiedliche Datentypen nur teilweise oder gar nicht verfügbar.

- **Hibernate**

Java-Framework für Objektpersistenz mittels Datenbanken. Hibernate bildet die Objekte einer Java-Applikation auf relationale Datenbanken ab.

- **Integrationstest**

Test des Gesamtsystems, bei dem die Tester Zugriff auf den Quellcode haben und dies beim Entwurf der Tests berücksichtigen. Von einem Systemtest wird ab der vollständigen Integration aller Komponenten gesprochen.

- **n-Version-Programming**

Programmierung von n Implementierungen der Software, um die Redundanz zu gegen Fehler in einzelnen Implementierungen zu erhöhen. Im Fall eines Fehlers stehen immer noch n-1 konsistente Ausgaben zur Verfügung. N-Version Programming ist aber dennoch anfällig für Spezifikationsfehler.

- **Echtzeitsystem**

Softwaresystem, an welches besondere zeitliche Anforderungen gestellt werden. Die korrekte Funktionsweise hängt auch von der Zeit ab, in der Ergebnisse erzeugt wurden.

- **Software Engineering**

Nach Sommerville: Eine technische Disziplin, die sich mit allen Aspekten der Softwareentwicklung beschäftigt, von den frühen Phasen der Systemspezifikation bis hin zur Wartung des Systems, nachdem sein Betrieb aufgenommen wurde.

## 2. In einem Softwareprojekt soll eine App zur Verwaltung und Überwachung von Ernährungsinformationen objektorientiert entwickelt werden. Die Anforderungsanalyse hat ergeben:

- Die Ernährungs-App verwaltet für ihren Benutzer Informationen. Der Benutzer gibt als Stammdaten Name, Vorname, Geburtsdatum, Körpergröße, Gewicht und Geschlecht an.
- Ein Administrator pflegt Gerichte und zu den Gerichten gehörende Nährwertlisten, die Angaben zu Brennwert und Gehalt an Vitaminen und Mineralstoffen enthalten
- Der Benutzer kann Gerichte zu seiner persönlichen Liste von Lieblingsgerichten hinzufügen und sie wieder entfernen
- Er kann sich diese Liste anzeigen lassen, auf eines der Gerichte klicken und bekommt dann dessen Nährwertliste gezeigt.

- Er kann bei einer Mahlzeit angeben, welche Gerichte er zu sich nimmt.
- Auf dieser Grundlage kann er sich die Summe der heute bereits zu sich genommenen Nährwerte anzeigen lassen.
- Für alle Nährstoffe kann er sich die durchschnittlich täglich zugeführten Werte der letzten Woche anzeigen lassen und diese mit den Sollwerten der Deutschen Gesellschaft für Ernährung vergleichen.
- ER kann sich seinen Body-Mass-Index berechnen lassen.
- Er kann sich BMI und Gewicht im Zeitverlauf als Grafik anzeigen lassen. Dazu muss er zunächst den gewünschten Zeitraum angeben.
- **Spezifizieren Sie den Use Case *Neues Gericht anlegen* in einem Use-Case-Dokument. Berücksichtigen Sie dabei alle wichtigen Inhalte, die in einem Use-Case-Dokument enthalten sein sollten.**

Use Case-Dokument: Neues Gericht anlegen

Beschreibung: Administrator legt neues Gericht an

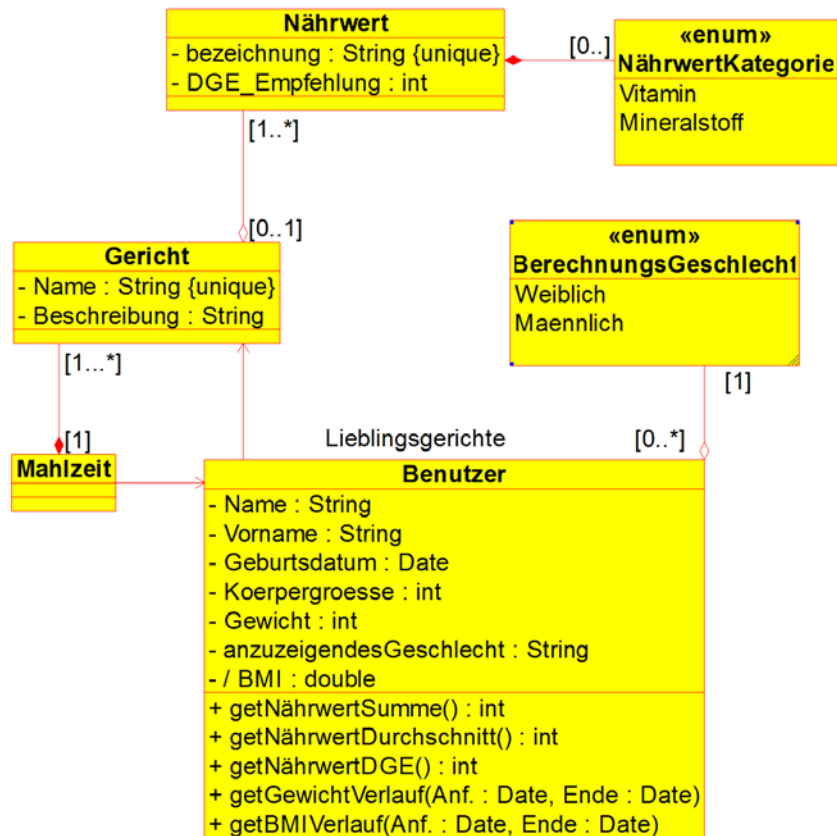
Preconditions: Administrator ist am System angemeldet

Basic Flow:

1. Administrator betätigt Schaltfläche "neues Gericht"
2. Administrator erhält Eingabemaske für die folgenden Informationen:  
Name, Beschreibung, Nährwerte (Brennwert, Vitamine, etc.)
  - a. Bei Eingabe des Namen überprüft das System selbstständig, ob der Name noch verfügbar ist und gibt visuell Rückmeldung, der Administrator kann den Namen anpassen, falls notwendig
  - b. Nährstoffe können per Instant-Suche gesucht und mit Mengenangaben versehen werden
  - c. Falls ein Nährstoff noch nicht existiert kann der Administrator diesen neu anlegen
3. Administrator bestätigt die Eingaben mit Klick auf "OK"
4. System gibt visuell Rückmeldung über Erfolg und Schaltfläche, die Aktion rückgängig zu machen die Details nochmal zu überarbeiten

Postconditions: Neues Gericht wurde angelegt

- **Zeichnen Sie ein Use-Case-Diagramm für das neue System**
- **Zeichnen Sie ein Klassendiagramm für das System.**  
Hinweise:
  - **Modellieren Sie alle notwendigen Attribute. Methoden brauchen Sie nicht zu modellieren**
  - **Die Unterscheidung zwischen mobil gehaltenen Daten und serverbasierten Daten dürfen Sie vernachlässigen. Zeichnen Sie ein einziges Klassendiagramm.**



**Begründen Sie, welche Systemarchitektur Sie für das System vorschlagen**

Ich würde eine zweistufige Client-Server-Architektur vorschlagen, bei der ein ThinClient die Dateneingabe und Präsentation übernimmt, während ein Server Anwendungslogik und Datenhaltung übernimmt. Bei einer großen Benutzerzahl wäre auch eine dreistufige Architektur denkbar, bei der Anwendungsserver und Datenbankserver aufgeteilt werden.