

# Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2014/2015 -

Dr. Günter Kniesel

## Übungsblatt 06

Zu bearbeiten bis: 28.11.2014

Bitte fangen Sie **frühzeitig** mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können. Checken Sie die Lösungen zu den Aufgaben in Ihr Repository ein, „Erklärungen“ bitte als Textdatei.

Fragen zu Übungsaufgaben/Vorlesung können Sie auf der Mailingliste [swt-tutoren@lists.iai.uni-bonn.de](mailto:swt-tutoren@lists.iai.uni-bonn.de), bzw. [swt-vorlesung@lists.iai.uni-bonn.de](mailto:swt-vorlesung@lists.iai.uni-bonn.de) stellen.

### **Aufgabe 1.** Anforderungs-Erhebung (7 Punkte)

*Hinweis: Bei dieser Aufgabe handelt es sich um eine ehemalige Klausuraufgabe. Versuchen Sie zur Übung daher zunächst, das Diagramm formal korrekt und vollständig auf Papier zu erstellen, bevor Sie die Lösung entwerfen und einchecken.*

Ziel dieser Aufgabe ist es, das interne Angestellten-Verwaltungs-System einer großen Firma funktional zu modellieren.

- Angestellte sollten in der Lage sein, ihre Flüge über das System zu buchen. Das System soll sowohl für dienstliche, als auch für private Flüge verwendet werden können.
- Des Weiteren sollen die Angestellten ihre Dienststunden mit dem System verwalten können. Flugstunden von dienstlichen Flügen sollen automatisch als Dienststunden angerechnet werden.

Die Eingabe von ungültigen Werten für die Dienststunden soll als Ausnahmefall betrachtet werden.

- a) Zeichnen Sie ein Anwendungsfalldiagramm, um Ihre Lösung für das angegebene Problem darzustellen. Wenn Sie Abhängigkeiten zwischen Anwendungsfällen modellieren, begründen Sie diese.
- b) Welche nicht-funktionalen Anforderungen sind Ihrer Meinung nach für dieses System wichtig? Nennen Sie zwei passende nichtfunktionale Anforderungen und begründen Sie kurz Ihre Wahl.

## Aufgabe 2. Analysephase (12 Punkte)

Eine Firma hat den Auftrag erhalten eine Kasse zu entwerfen. Die Kasse soll folgende Elemente enthalten:

- Ein Scanner für die Erkennung der Artikelnummern.
- Eine Taste zum Berechnen des Endpreises.
- Eine Listen-Anzeige für den Kassierer/die KassiererIn mit einer Auflistung der Artikel des aktuellen Einkaufs (jeweils Bezeichnung und Preis).
- Eine Einzel-Anzeige auf Seite des Kunden, die den Preis des letzten eingegeben Artikels bzw. den Endpreis anzeigt.

Die Durchführung eines Einkaufs wird durch folgende Sequenz von Ereignissen beschrieben:

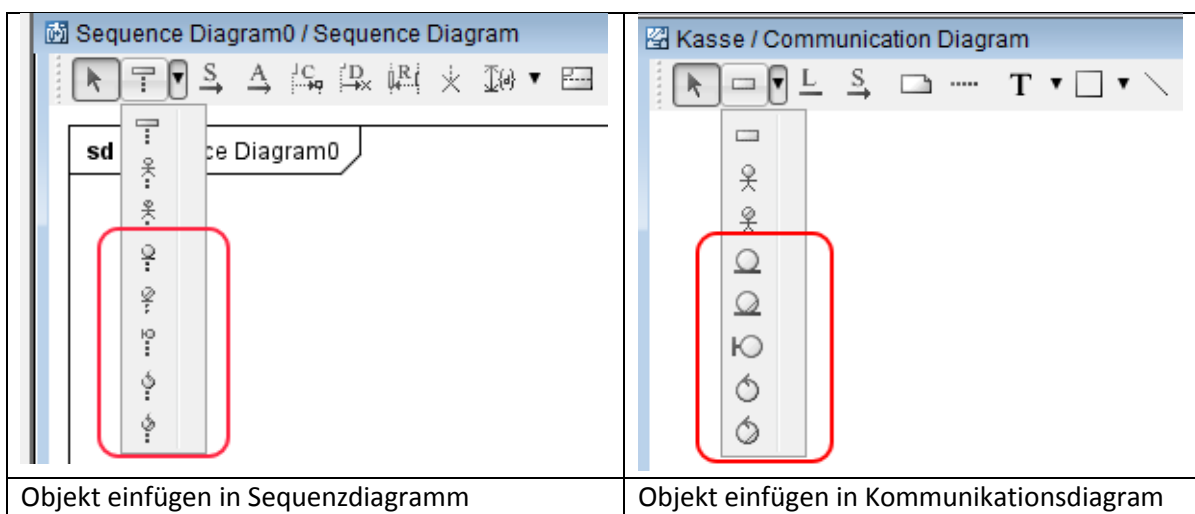
- Der Kassierer/die KassiererIn scannt einen Artikel ein.
- Daraufhin werden alle Anzeigen aktualisiert.
- Dieser Vorgang kann beliebig oft wiederholt werden, bis der Kassierer/die KassiererIn die Taste zur Berechnung des Endpreises drückt, woraufhin auf beiden Anzeigen der Endpreis angezeigt wird.

a) Zeichnen Sie ein **Klassendiagramm** für die Elemente der Kasse. Überlegen Sie sich Klassen, Operationen und Attribute. Fügen Sie Assoziationen ein, wo es Ihnen sinnvoll erscheint. Markieren Sie Boundaries, Controller und Entities.

b) Zeichnen Sie ein **Sequenzdiagramm** für obige Sequenz von Ereignissen.

c) Zeichnen Sie ein **Kommunikationsdiagramm** für obige Sequenz von Ereignissen.

Nutzen Sie auch im Sequenz- und Kommunikationsdiagramm die Symbole für Boundary, Controller und Entity. Falls Sie z.B. mit Astah arbeiten, finden Sie sie hier:



**Aufgabe 3.** Systementwurf (7 Punkte)

Die Softwarefirma entscheidet sich, eine **Model-View-Controller-Architektur** für die Modellierung der Kasse aus Aufgabe 2 zu verwenden. Die Anzeigen (views) sollen sich im Rahmen ihrer Initialisierung beim Modell anmelden, um Aktualisierungs-Benachrichtigungen zu erhalten.

- a) Überlegen Sie sich eine sinnvolle Gruppierung der Analyseklassen in Komponenten (Subsysteme) und definieren Sie Dienste. Zeichnen Sie ein passendes **Komponentendiagramm**.
- b) Diskutieren Sie, wie die Wahl der **Model-View-Controller-Architektur** folgende Entwurfsziele erfüllt oder verletzt:
- Erweiterbarkeit (z.B. neue „Views“)
  - Reaktionszeit (Zeit zwischen einer Benutzereingabe und dem Abschluss der Aktualisierung aller Views)
  - Änderbarkeit (z.B. die Erweiterung des Modells um zusätzliche Attribute)
  - Zugriffs-Kontrolle (d.h. die Sicherstellung, dass nur berechtigte Benutzer auf bestimmte Teile des Modells zugreifen können)