Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik III

Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2019/20 -

Dr. Günter Kniesel

Übungsblatt 3

Abgabe: 25.10.2019, 16:00 Uhr

Bitte fangen Sie **frühzeitig** mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können. Checken Sie die Lösungen zu den Aufgaben bitte in Ihr Repository ein, „Erklärungen" bitte als Textdatei. Fragen zu Übungsaufgaben respektive zur Vorlesung können Sie auf der Mailingliste swt-tutoren@lists.iai.uni-bonn.de, bzw. swt-vorlesung@lists.iai.uni-bonn.de stellen.

Praktische Aufgaben

Aufgabe 1. *Klassendiagramm: Assoziation, Aggregation und Komposition* **25,5 Punkte**

Zeichnen Sie Klassendiagramme, die folgende Sachverhalte modellieren. Notieren Sie auch die Multiplizitäten.

Sie können das Diagramm per Hand anfertigen (gut als Vorübung für die Klausur!). Falls Sie lieber ein UML-Tool nutzen, empfehlen wir Astah UML (frei für Studenten). Ihr Ergebnis bitte als png oder pdf abgeben.
(siehe auch https://sewiki.iai.uni-bonn.de/teaching/lectures/se/2019/infrastruktur/casetool)

a) **(10,5 Punkte)** Ein Fernsehtisch besteht aus

- einer Tischplatte
- vier Tischbeinen
- einer weiteren Platte unter der Tischplatte

Auf dem Tisch kann eine oder auch mehrere Tischdecken liegen. Um nicht übermäßig zu dekorieren, begrenzen wir die Anzahl der Tischdecken auf drei Stück. Außerdem liegt auf der unteren Platte traditionellerweise eine Fernbedienung.

b) **(14 Punkte)** In einer Schule gilt folgendes:

- Die Schule hat als Lehrende eine nicht-leere Menge an Lehrern sowie eine evtl. leere Menge an Referendaren.
- Einer der Lehrer erfüllt die Rolle des Direktors, ein anderer die des stellvertretenden Direktors.
- Referendare sind keine Lehrer (da sie erst noch ihr Examen ablegen müssen)
- Jeder Referendar wird von einem anderen Lehrer betreut.
- Die Schule hat eine Menge an Schülern
- Alle Lehrenden unterrichten jeweils eine Teilmenge der Schüler.

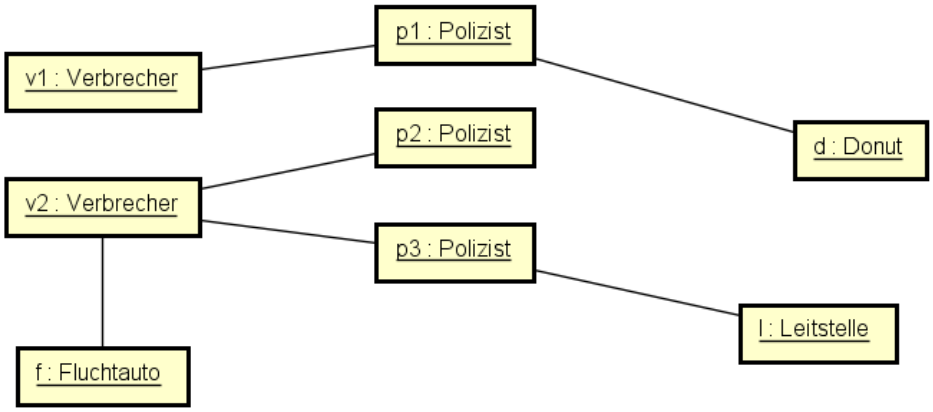
Aufgabe 2. *Klassendiagramm zu Objektdiagramm* **11 Punkte**

a) **(9 Punkte)** Geben Sie nun zu jedem der beiden Beispiele aus (a) und (b) ein *Objektdiagramm* an, dass sich aus dem Klassendiagramm ergeben könnte.

b) **(2 Punkte)** Geben Sie nun zu jedem der beiden Beispiele aus (a) und (b) ein Objektdiagramm an, dass **nicht** zu den Klassendiagrammen passt.

Aufgabe 3. *Objektdiagramm zu Klassendiagramm* **14,5 Punkte**

Setzen Sie das folgende Objektdiagramm in ein strukturell passendes Klassendiagramm um. Überlegen Sie sich, über welche Assoziationen, Rollen und Kardinalitäten die beteiligten Klassen verbunden sein könnten.



Reichen Sie Ihre Lösung in zwei Versionen ein:

- Eine, in der Sie genau den Sachverhalt des Objektdiagramms wiedergeben.
- Eine, in der Sie auf Basis eigener Vermutungen das Modell so generalisiert haben, dass auch weitere, Ihnen plausibel erscheinende Szenarien mit abgedeckt sind.

Geben Sie zu Ihrer Generalisierung eine Frage an, die Sie Ihrem Auftraggeber stellen würden, um zu überprüfen, ob Ihre Vermutung zutrifft.

Theoretische Aufgaben

Aufgabe 4. *Funktionen von Git-Befehlen* **3,5 Punkte**

a) **(1 Punkt)** Was ist der Unterschied zwischen einem harten und einem weichen Reset?

b) **(2 Punkt)** Zeichnen Sie beispielhaft die interne Darstellung für die folgenden Befehle:
   1) git reset --soft commit3
   2) git cherry-pick commit4

c) **(0,5 Punkte)** Sind unvollständige Zustände eines Repositories unter Git möglich?

Aufgabe 5. *UML Grundlagen* **10,5 Punkte**

a) **(1,5 Punkt)** Erklären Sie die Begriffe *Forward Engineering*, *Reverse Engineering* und *Round-Trip Engineering*.

b) **(2 Punkte)** Ordnen Sie alle Diagrammtypen den Kategorien *Strukturdiagramme*, *Verhaltensdiagramme* und *Interaktionsdiagramme* zu.

c) **(1 Punkt)** Wie werden Instanzen und Typen in UML dargestellt?

d) **(2 Punkte)** Welche Komponenten gehören alles zu einer Klasse (in UML)?

e) **(1,5 Punkte)** Wie werden die Sichtbarkeiten *public, private* und *protected* markiert?

f) **(0,5 Punkte)** Finden Sie (z.B. im Internet) heraus, ob es eine spezielle Notation für die Sichtbarkeit „package-private“ gibt.

g) **(1 Punkt)** Wie werden Klassenvariablen und -methoden dargestellt?

h) **(1 Punkt)** Wie werden abstrakte Klassen und Methoden dargestellt?

∑ 65 Punkte