

Softwaretechnik und Programmierparadigmen WiSe 2014/2015

Prof. Dr. Sabine Glesner

Joachim Fellmuth

joachim.fellmuth@tu-berlin.de

Dr. Thomas Göthel

thomas.goethel@tu-berlin.de

Lydia Mattick

lydia.mattick@tu-berlin.de

Tutoren

Übungsblatt 5

Ausgabe: 13.11. (Besprechung: 17.11. und 18.11.)

1. Erweiterung des Klassendiagramms

Die Modellierungssprache des Klassendiagramms erlaubt weitaus mehr Ausdrucksformen. Das in der Vorlesung vorgestellte **Systemklassendiagramm** teilt die Klassen in disjunkte **Kategorien** ein und separiert Klassen innerhalb des Systems von außen mittels der **Systemgrenze**.

- **<<Entity>>**: eine Klasse, welche ausschließlich zur **Datenspeicherung** verwendet wird; dessen Methoden beschränken sich meist auf **get-** und **set-**Methoden.
- **<<Control>>**: beinhaltet komplexe **Methoden**, welche u.a. funktionale Requirements erfüllen; hat Zugriff auf alle relevanten **Entity**-Klassen.
- **<<Boundary>>**: eine **Schnittstelle** (außerhalb der Systemgrenze) für einen Akteur zu den benötigten **Control**-Klassen; wird selten explizit erwähnt.

2. OCL: Ausdrücke

Die Object Constraint Language erlaubt es, komplexe Datensätze formal, standardisiert und logisch auszudrücken. Formalisiert die folgenden Ausdrücke oder denkt euch eigene aus:

- a) Der Name des Kunden **harry**
- b) Die Menge aller Motorräder von **harry**

- c) Der alleinige Besitzer des Motorrads **harley**
- d) Im Kontext des Admins:
 - a) Die Menge aller Bestellungen
 - b) Die Menge aller Aufträge, welche im November 2014 eingegangen sind
 - c) Die Menge aller Datumsangaben aller Aufträge, welche im November 2014 eingegangen sind
 - d) Die Menge aller Aufträge und zusätzlich der Auftrag **tune404**
 - e) Die Menge aller Motorräder, die einem noch nicht abgeschlossenen Auftrag im System zugeordnet sind oder dem Kunden **hardy** gehören

3. OCL: Bedingungen

Logische Ausdrücke sind statisch verifizierbar und werden auf verschiedene Weise eingesetzt. Formalisiert die folgenden logischen Aussagen oder denkt euch eigene aus:

- a) Die Namen von **harry** und **hardy** sind nicht gleich
- b) Die **harley** gehört **harry**
- c) Die Anzahl der Motorräder des Auftrages **tune404** ist 1
- d) Im Kontext des Admins:
 - a) Die Motorräder von **harry** sind auch dem Admin bekannt
 - b) Mindestens ein Tuningauftrag ist noch nicht beendet
 - c) Alle Reparaturaufträge sind beendet
 - d) Alle Reparaturaufträge aller Motorräder von **hardy** sind beendet

4. OCL: Invarianten

Invarianten sind Bedingungen von Klassen, die zu jeder Zeit gelten sollen. Formalisiert die folgenden Invarianten oder denkt euch eigene aus:

- a) Die Daten jedes Auftrages sind früher als die Variable **TODAY**
- b) Jeder Auftrag hat genau ein Motorrad
- c) Alle beendeten Aufträge sind einem Mitarbeiter zugeordnet
- d) Alle Motorräder mit Aufträgen haben auch einen Besitzer
- e) Alle Motorräder haben unterschiedliche Kennzeichen