

Softwaretechnik und Programmierparadigmen WiSe 2014/2015

Prof. Dr. Sabine Glesner

Dr. Thomas Göthel

thomas.goethel@tu-berlin.de

Lydia Mattick

lydia.mattick@tu-berlin.de

Tutoren

Übungsblatt 4

Ausgabe: 6.11. (Besprechung: 10.11. und 11.11.)

1. Klassendiagramm

Klassendiagramme bieten eine schnelle Übersicht über die Struktur der Software. Im einfachen **Klassendiagramm** werden zusätzlich zu den Attributen der einzelnen Klassen auch dessen Beziehungen untereinander dargestellt.

Auf Basis der auf dem letzten Übungsblatt modellierten Klassen soll nun das Klassendiagramm vervollständigt werden. Benutzt und bespricht dazu nach Bedarf folgende Designelemente:

- Assoziationen, Aggregationen und Kompositionen mit
 - Bezeichnung mit Leserichtung,
 - sinnvollen Multiplizitäten,
 - Navigierbarkeit,
- Generalisierung / Spezialisierung.

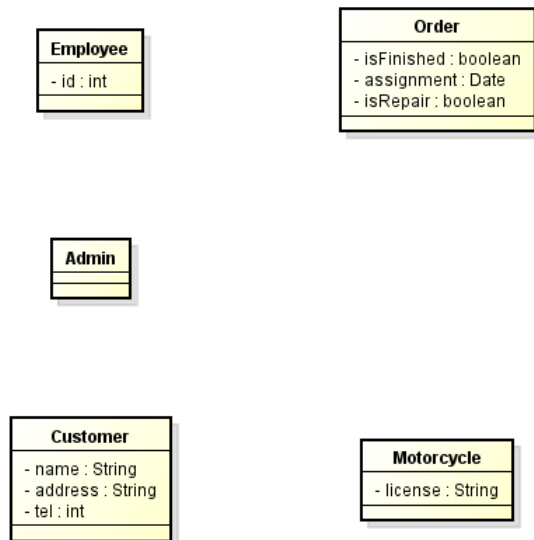
Anforderungen (Wiederholung):

Motorsport Gomez möchte die Abfertigung der Aufträge mit einer Software verwalten.

Ein Kunde kann eine Reparatur oder ein Tuning für sein Motorrad vom Verwaltungspersonal in Auftrag geben lassen. Dazu wird das Kennzeichen des Motorrads im System vermerkt und der Auftrag erhält automatisch einen Datumsstempel. Wenn ein Auftrag

beendet ist, kann ein Mitarbeiter diesen als beendet markieren.

Das Verwaltungspersonal soll die Möglichkeit haben, Kundendaten aufzunehmen sowie Mitarbeiterkontos zu erstellen. Kundendaten beinhalten Name, Adresse und Telefonnummer. Bei der Erstellung eines Mitarbeiterkontos wird die automatisch generierte Mitarbeiter-ID zurückgegeben.



2. Objektdiagramm

Modelliert nun einige Beispiel-Objektdiagramme für das soeben modellierte Klassendiagramm.

Achtet dabei insbesondere auf die Einschränkungen durch Multiplizitäten.

3. Erweitertes Sequenzdiagramm / Objektinteraktionsdiagramm

Im Gegensatz zum einfachen Sequenzdiagramm werden nun die Interaktionen der Objekte innerhalb des Systems genauer modelliert. Im Laufe der Modellierung werden oft neue Methoden benutzt, um welche dann das Klassendiagramm erweitert werden kann.

Folgende Anwendungsfälle sollen modelliert werden:

- a) „neuen Auftrag erstellen“
Der Auftrag soll nur erstellt werden, wenn ein existierender Kunde und Motorrad übergeben wird.
- b) „alle Aufträge eines Mitarbeiters ausgeben“
Überlegt euch was für Überprüfungen sinnvoll sind.

4. Struktur- und Verhaltensdiagramme

- a) Nennt einige UML-Struktur- bzw. Verhaltensdiagramme. Was sind wesentliche Unterschiede der beiden Kategorien?
- b) Welche Vorteile hat die Nutzung von UML-Diagrammen?