



Formale Methoden

SS 2004 – Übungsblatt 3

19. Mai 2004

Ausgabe: 13. Mai 2004

Aufgabe 1. *Design eigener Klassen*

Konzipieren Sie eigene Klassen `Verein`, `Vorstand`, `Vorstandsmitglied`, `Mitglied` und `Person`. Leiten Sie dabei die Klassen `Vorstandsmitglied`, `Mitglied` und `Person` geeignet voneinander durch Vererbung ab.

Die konzipierten Klassen sollen für ein Softwarepaket „Vereinsverwaltung“ eingesetzt werden. Ergänzen Sie alle Ihnen für diesen Zweck nötig erscheinenden Attribute und Methoden.

Spezifizieren Sie innerhalb der einzelnen Klassen die Invarianten sowie die Vor- und Nachbedingungen der Methoden in OCL-Syntax. Beachten Sie dabei die Prinzipien des »Programming by Contract«, das heißt: spezifizieren Sie immer nur mit Hilfe von wenigen zentralen »basic queries«. In welche anderen Kategorien sollten Sie die Methoden einteilen?

Aufgabe 2. *Design von Klassen-Interdependenzen*

Ergänzen Sie die Klassen von Aufgabe 1 um die nötigen Assoziationen, Aggregationen und Kompositionen. Ergänzen Sie auch nötige Hilfsklassen wie zum Beispiel `Datum`, ohne jedoch deren Innenleben schon genauer zu konzipieren. Benutzen Sie unter anderem auch qualifizierte Assoziationen (`Mitgliedsnummer`) sowie Assoziationsklassen (`Eintrittsdatum`).

Aufgabe 3. *OCL Constraints*

Schreiben Sie in OCL-Syntax alle für die Vereinsverwaltungssoftware nötigen Abhängigkeiten der Klassen, der Klassenattribute und der Klassenmethoden von anderen Klassen.

Aufgabe 4. *Invarianten, Vor- und Nachbedingungen bei der Vererbung*

Referieren Sie kurz die Regeln, die bezüglich Klasseninvarianten, Vor- und Nachbedingungen bei der Vererbung eingehalten werden müssen.

Demonstrieren Sie diese Regeln an Beispielen der obigen Aufgaben.

Aufgabe 5. *OCL-Manual*

Lesen Sie die OCL-Spezifikation unter

<http://neptune.irit.fr/Biblio/01-09-77.pdf>

und beantworten Sie die folgenden Fragen:

- Wozu dient der Package-Kontext?
- Welche vordefinierten Standard-Datentypen und welche Operationen auf deren Exemplaren bietet OCL an?
- Welchem Zweck dienen «definition» Constraints?
- Erläutern Sie die Prioritäts-Regeln von OCL-Operationen.
- Welche vordefinierten Schlüsselwörter kennt OCL?
- Wann ist das Ergebnis einer binären Operation definiert, obwohl einer der beiden beteiligten Operanden undefiniert ist? Warum ist diese Regelung sinnvoll?
- Diskutieren Sie den Unterschied der Operationen `isUnique()` und `one()` (vergleiche Seite 6-38)!