Prof. Dr. V. Linnemann
Ute Masermann
Medizinische Universität Lübeck
Institut für Informationssysteme

Lübeck, 07. Januar 1997

Nonstandard-Datenbanksysteme
Wintersemester 1996/97
3. Übungsblatt

Das voliegende Übungsblatt beschäftigt sich weiterhin mit O_2, jetzt geht es um Behandlung von Methoden und um komplexere Anfragen.

Um die Anfragen auf einer geeigneten Datenbasis ausführen zu können, sollten nach dem Setzen des Schemas und der Basis diese Daten geladen werden. Sie stehen in der Datei
`http:www.ifis.mu-luebeck.de/~ifis/lehre/ws9697/ndb/uebung/ueb3.o2`.

Um eine solche Datei einzuladen (und damit alle darin stehenden O_2-Befehle auszuführen), wird in der O2Shell folgendes Kommando abgesetzt: #"<*dateiname*>",
z.B. #"/home/maserman/uebung.o2"

Achtung: Sowohl die Klassen- als auch Attribut- und Instanzennamen müssen mit denen übereinstimmen, wie sie in der Kurzeinführung und der 2. Übung vorgegeben wurden (z.B. Klasse `UebungsHiWi`), sonst tritt ein Laufzeitfehler auf. Überprüfen Sie gegebenenfalls Ihr Schema und benennen Sie die Klassen um oder editieren Sie die einzuladende Datei!

Aufgabe 1: Methoden

a) **Einführung**
Um Methoden zu erzeugen bzw. zu ändern, gibt es in dem Klassenbrowser die Spalte *Methoden*. Hier werden alle klassenspezifischen Methoden angezeigt bzw. über `Methods:Show Inherited` alle geerbten.

Jetzt soll für die Klasse `UebungsHiWi` die Methode `Monatsgehalt` angelegt werden: Mit `Method:Create` wird der Name definiert, anschließend öffnet sich der Methodeneditor. Diese Methode soll *public* sein, d.h. andere Klassen dürfen darauf zugreifen. Deshalb wird bei `Visibility` der `Public`-Button angeclickt.

Als nächstes wird die Signatur festgelegt. In diesem Fall wird der „Stundenlohn" von Typ `real` als Parameter übergeben, als Ergebnis soll ebenfalls eine `real`-Zahl

zurückgegeben werden. Die Signatur wird wie folgt eingegeben:
`(Stundenlohn:real):real`

Die O_2C-Syntax lehnt sich sehr stark an C an, insbesondere gilt dies für die if/then/else-Konstruktion, Boolesche Ausdrücke und Zeigernotation (wie auch schon in der Kurzeinführung gezeigt). Ein Nichtinitialisierter Zeiger hat den Wert `nil`. Durch `self` wird immer das Objekt identifiziert, das die Methode ausführt.

Der Methoden-Body fällt in diesem Beispiel recht kurz aus: Es muß lediglich die Stundenzahl mit dem Parameter des Stundenlohns multipliziert werden. Er lautet deshalb

```
return(self->Stundenzahl * Stundenlohn);
```

Nach dem Compilieren mit `Method:Compile` ist es z.B. möglich, diese mit

```
select h.Monatsgehalt(16.25)
from h in HiWis
```

aufzurufen. (1 Punkt)

b) **Redefinition**
Bei dem Anzeigen (mit `Display Values`) von beispielsweise `Personen` wird als Titel über die Methode `title` jeweils der Klassenname angezeigt. Das reicht uns nicht aus, deshalb soll diese von `Object` geerbte Methode in der Klasse `Person` so verändert (d.h. überschrieben) werden, daß sie den Namen dieser Person anzeigt. Testen Sie diese Methode mit Instanzen von verschiedenen Subklassen von `Person`. (3 Punkte)

c) **Programmierung**
Ergänzen Sie die Klasse `Student` um die parameterlose Methode `Semesterberechnung`, die aus dem Immatrikulationsdatum die Semesterzahl berechnet. Es darf vorausgesetzt werden, daß keine Urlaubssemester in Anspruch genommen wurden und daß Immatrikulationen am 01.10. jeden Jahres stattfinden. Achten Sie auf korrekte Fehlerbehandlung.

Hinweise zur Klasse `Date` sind Kapitel 1.1 des *O_2Kit User Manual* zu entnehmen. Mehrere Kopien dieses Kapitels liegen im Rechnerraum des IfIS aus. (4 Punkte)

Aufgabe 2: Anfragen

Um mehrere Attribute einer Instanz anzuzeigen, muß in O_2 – anders als z.B. in relationalen DBMS – ein Typ konstruiert werden, der diese Attribute geeignet aufnehmen kann.

Beispiel:
```
select tuple(Name: s.Name, Semester_Attribut: s.Semester,
             Semester_Methode: s.Semesterberechnung)
from s in Studenten
```
liefert eine Menge von Tupeln zurück. Aggregatfunktionen (z.B. `count`) funktionieren analog zu RDBMS.

Der aus der Vorlesung bekannte `flatten`-Operator (Kapitel 2.2, S. 8) wird in O_2 wie folgt angewandt (bezogen auf das Beispiel in Kapitel 2.1, S. 27 der Vorlesung): Es sollen die Namen aller Angestellten herausgesucht werden, die bei Niederlassungen in Boston arbeiten.
```
select x.Name
from x in flatten (select y.Angestellte from y in Niederlassung
                   where y.Sitz.Ort="Boston")
```
`y` durchläuft hier die Menge aller Niederlassungen; mit `select y.Angestellte...` entsteht eine Menge von Mengen, die dann „flachgeklopft" werden muß.

a) Wieviele Veranstaltungen besucht der Student (aus der Instanz `Studenten`) mit Namen „Neuling"? Geben Sie ein OQL-Statement dafür an. (1 Punkt)

b) Geben Sie zwei mögliche OQL-Statements für die Aufgabe 1.c) des ersten Übungsblattes an und führen Sie sie aus (Hinweis: Hier kann z.B. der `flatten`-Operator angewandt werden!) (4 Punkte)

c) Zusätzlich zu dem Dozentennamen soll nun auch der Titel der vorausgesetzten Veranstaltung herausgesucht werden. (1 Punkt)

d) Schreiben Sie ein OQL-Statement, das herausfindet, ob der UebungsHiwi „Hiwilein" (in der Instanz `HiWis`) an den Gebäuden, in denen die von ihm betreuten Veranstaltungen stattfinden, das Fahrrad ordnungsgemäß abstellen kann.(1 Punkt)

Abnahmetermin: Nach Terminabsprache bis spätestens Montag, 20.01.97 im Rechnerraum IFIS, Osterweide 8. Fragen können am Montag, 13.01.07, 14-16h sowie freitags, 11-13h im IfIS gestellt werden.