

Aufgaben zur Klausur **Softwaredesign** im SS 2006 (WI h252, WI h253, II h752, MI h403, MI h404, MI h405, BInf v310, BMinf v300, BWInf v310)

Zeit: 90 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 9 Seiten

---

### Aufgabe 1:

Entwickeln Sie zu einer gegebenen Anfragesprache für eine Freitextsuche in einem Dokumentenindex eine abstrakte Syntax. Die konkrete Syntax sei durch folgende in BNF-Notation gegebene kontextfreie Grammatik beschrieben. Dabei sind Terminalsymbole in ' gesetzt.

- .0  $Query ::= Query 'OR' Query_1 \mid Query_1$
- .1  $Query_1 ::= Query_1 'AND' Query_2 \mid Query_2$
- .2  $Query_2 ::= Query_3 'NEAR' Query_2 \mid Query_3$
- .3  $Query_3 ::= Query_4 Query_3 \mid Query_4$
- .4  $Query_4 ::= ' (' Query ' )' \mid Query_5$
- .5  $Query_5 ::= Query_6 \mid Query_6 '...'$
- .6  $Query_6 ::= Word$

In der Anfragesprache gibt es also Und- und Oder-Verknüpfungen, es gibt eine Kontext-Suche, in der ein Wort in der Nähe eines anderen gesucht werden soll (Operator *NEAR*), man kann nach Folgen von Wörtern suchen (1. Regel für *Query<sub>3</sub>*), nach einfachen Wörtern und nach Präfixen (Operator ...). Es gibt also vier binäre Operatoren und einen unären Operator in der Sprache. Wie ein Wort dabei aufgebaut sein soll, bleibt hier offen, ein Wort wird durch eine beliebige Zeichenreihe repräsentiert.

Entwickeln Sie für diese Sprache eine abstrakte Syntax für die interne Verarbeitung von Anfragen. Versuchen Sie durch Abstraktion ein möglichst einfaches Modell zu entwickeln.

- 1) .....
- 2) .....
- 3) .....
- 4) .....
- 5) .....
- 6) .....
- 7) .....
- 8) .....
- 9) .....
- 10) .....
- 11) .....
- 12) .....

Welche Strukturmuster findet man in diesem Datenmodell wieder?

1) .....

2) .....

3) .....

Welche Verhaltensmuster sind zur Verarbeitung dieser Strukturen geeignet?

1) .....

2) .....

3) .....



**Aufgabe 2:**

Entwickeln Sie ein Datenmodell für die Verarbeitung von XML und HTML. In diesem etwas vereinfachten Modell ohne Kommentare und Processing Instructions soll ein XML-Dokument aus einem (Wurzel-)Element bestehen. Ein Element setzt sich zusammen aus einem Elementnamen, einer Menge von Attributen und einem Inhalt. Der Inhalt ist eine Folge von einfachem Text und Elementen.

Verwenden Sie in dem Datenmodell pro Typdefinition nur einen Typkonstruktor. *String* sei ein vordefinierter Datentyp für Zeichenreihen.

Eine Abstrakte Syntax für ein *XMLDokument*:

- 1) .....
  - 2) .....
  - 3) .....
  - 4) .....
  - 5) .....
  - 6) .....
  - 7) .....
  - 8) .....
  - 9) .....
  - 10) .....
-

**Aufgabe 3:**

Welches sind die Vorteile des Besuchermusters gegenüber dem Interpretierermuster?  
Aussagekräftige Stichworte:

1) .....

2) .....

3) .....

4) .....

5) .....

Welches sind die Nachteile des Besuchermusters gegenüber dem Interpretierermuster?  
Aussagekräftige Stichworte:

1) .....

2) .....

3) .....

4) .....

5) .....

**Aufgabe 4:**

Welche Strukturmuster sind geeignet, um rekursive Datenstrukturen zu modellieren?

1) .....

2) .....

3) .....

4) .....

5) .....



### Aufgabe 5:

Gegeben sei der folgende Teil eines Datenmodells für einen CD-Katalog:

Das Modell in abstrakter Syntax in Haskell-Notation:

```
.0 data Sammlung           = Sammlung [Album] Attribute
.1 data Album             = Album [CD] Attribute
.2 data CD                 = CD [Stueck] Attribute
.3 data Stueck            = Stueck Attribute
.4 type Attribute         = Map Attr Wert
.5 type Attr              = String
.6 type Wert              = String
```

Hinweis: In den Gleichungen (.0–.3) ist der erste Name rechts vom Gleichheitszeichen der Konstruktorname.

In diesem Datenmodell ist die Hierarchie streng festgelegt: Es gibt immer genau vier Stufen in der Hierarchie. Dieses ist für die Strukturierung eines Kataloges unflexibel. Zum Beispiel kann man Sammlungen nicht in Teilsammlungen aufgliedern, eine Sammlung kann auch keine CDs und Alben gleichzeitig enthalten.

Entwickeln Sie ein flexibleres Datenmodell, in dem beliebige Hierarchien von Sammlungen, Alben, CDs und Stücken möglich sind, es aber nur diese vier Arten von Knoten gibt.

- 1) .....
- 2) .....
- 3) .....
- 4) .....
- 5) .....
- 6) .....
- 7) .....
- 8) .....

Welche zusätzlichen Konsistenzbedingungen (Datenstruktur-Invarianten) müssen für diese Datenstruktur eingehalten werden, damit ein sinnvoller CD-Katalog entsteht.

- 1) .....
- 2) .....
- 3) .....
- 4) .....
- 5) .....