

Aufgaben zur Klausur **Softwaredesign** im SS 2013 (BInf v310, BMinf v300, BWInf v310, BWInf-23)

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.

Nutzen Sie die Rückseiten der Klausur zur Entwicklung der Lösungen und übertragen die fertigen Lösungen in das Aufgabenblatt.

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 10 Seiten.

Aufgabe 1:

In dieser Aufgabe geht es darum, ein Datenmodell für JSON zu entwickeln. Auf json.org findet man folgende Definition:

JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, ...

JSON baut auf zwei Strukturen auf:

Name/Wert Paare In verschiedenen Sprachen wird dies realisiert als ein Objekt (object), Satz (record), Struktur (struct), Wörterbuch bzw. Verzeichnis (dictionary), Hash-Tabelle (hash table), Schlüssel-Liste (keyed list) oder als ein assoziatives Array (associative array).

Eine geordnete Liste von Werten In den meisten Sprachen wird das als Array (array), Vektor (vector), Liste (list) oder Sequenz (sequence) realisiert.

In JSON gibt es:

Objekt Ein Objekt ist eine ungeordnete Menge von Name/Wert Paaren. Ein Objekt beginnt mit '{' (geschwungene Klammer auf) und endet mit '}' (geschwungene Klammer zu). Jedem Namen folgt ein ':' (Doppelpunkt) gefolgt vom Wert und die einzelnen Name/Wert Paare werden durch ',' (Komma) voneinander getrennt.

Array Ein Array ist eine geordnete Liste von Werten. Arrays beginnen mit '[' (eckige Klammer auf) und enden mit ']' (eckige Klammer zu). Werte werden durch ',' (Komma) voneinander getrennt.

Wert Ein Wert kann ein Objekt, ein Array, eine Zeichenkette (string), eine Zahl oder einer der Ausdrücke 'true', 'false' oder 'null' sein. Diese Strukturen können ineinander verschachtelt sein.

Entwickeln Sie aus dieser umgangssprachlichen Beschreibung ein Datenmodell in abstrakter Syntax in Haskell-Notation. Verwenden Sie dabei für Zeichenketten den vordefinierten Datentyp *String* und für Zahlen den Datentyp *Double* (Lösung auf der folgenden Seite).

Definieren sie aus den obigen Beschreibung einen Datentyp *Wert*:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Aufgabe 2:

Gegeben seien die folgende Datentypen:

- .0 **data** *Expr* = *Const Value*
- .1 | *UnExpr Op1 Expr*
- .2 | *BinExpr Op2 Expr Expr*
- .3 **type** *Value* = ...
- .4 **type** *Op1* = ...
- .5 **type** *Op2* = ...

Entwickeln Sie für den Datentyp *Expr* gleichwertige Klassendefinitionen in Java.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Aufgabe 3:

Gegeben sei die folgende Datenstruktur in Haskell-Notation:

```
.0 data Tree a = T0 a
.1           | T1 a (Tree a)
.2           | T2 a (Tree a) (Tree a)
.3           | T3 a (Tree a) (Tree a) (Tree a)
.4           | T4 a (Tree a) (Tree a) (Tree a) (Tree a)
```

Welche Strukturmuster sind in dieser Datenstruktur enthalten?

- 1)
- 2)
- 3)
- 4)

Vereinfachen (verallgemeinern) Sie diese Datenstruktur so, dass nicht mehr mit den vier verschiedenen Varianten (in OOP/OOD: Nicht mehr mit Vererbung) gearbeitet werden muss und dass mehr als 4 Nachfolger möglich werden. Bitte benutzen Sie pro Typdefinition nur einen Typkonstruktor.

Das vereinfachte Datenmodell:

.....

.....

.....

.....

Aufgabe 4:

Welche Entwurfsmuster dienen dazu, Methoden als Parameter oder in Daten zu verwenden?

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

Aufgabe 5:

Gegeben sei das folgende Datenmodell für eine einfache Programmiersprache:

```
.0 data Stmt      = Assign Id Expr
.1                | Sequence [(Maybe Lab, Stmt)]
.2                | While Expr Stmt
.3                | If Expr Stmt (Maybe Stmt)
.4 data Expr      = Const Int
.5                | Null
.6                | Var Id
.7                | Bin Op Expr Expr
.8 type Id        = String
.9 type Lab       = String
.10 data Op       = Plus | Minus
```

Welche Strukturmuster kommen in diesem Modell vor?
Geben Sie jeweils den Musternamen und die beteiligten Datentypen an.

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

Welche Datentypen könnte man sinnvollerweise in einer Java-Implementierung als Fliegengewichte verwalten?

.....
.....
.....

Welche Datentypen könnte man sinnvollerweise in Java als Singletons implementieren?

.....
.....
.....

Aufgabe 6:

Modellieren Sie ein einfaches elektronisches Photoalbum.

Ein Photoalbum soll aus einem Titel und einer Folge von Einträgen bestehen. Ein Eintrag ist entweder ein Bild oder ein Kapitel. Ein Kapitel besitzt wieder einen Titel und eine Liste von Einträgen.

Ein Bild wird durch einen URL identifiziert. Außerdem soll es möglich sein, zu einem Bild beliebige Attribute zu speichern, wie zum Beispiel Bildtitel, Datum, Ort, technische Daten usw.

Entwickeln sie hierfür als Datenmodell eine abstrakte Syntax. Verwenden Sie für die Datentypen die Namen aus der Aufgabenstellung. Benutzen Sie pro Datentyp nur einen Typkonstruktor. Die einfachen Datentypen seien dabei alle implizit als *String* deklariert.

Das Datenmodell in Form einer abstrakten Syntax:

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

Welche Strukturmuster kommen in diesem Modell vor? Nennen Sie den jeweiligen Musternamen, die beteiligten Klassen und die beteiligten Methoden und/oder Referenzen.

1. Musternamen, beteiligte Klassen und Referenzen

.....
.....
.....

2. Musternamen, beteiligte Klassen und Referenzen

.....
.....
.....

3. Musternamen, beteiligte Klassen und Referenzen

.....
.....
.....

4. Musternamen, beteiligte Klassen und Referenzen

.....
.....
.....

