

---

Aufgaben zur Klausur **Software Entwicklungs–Methoden** im SS 97 (WI 56)

Zeit: 75 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:

Gegeben sei die folgende hierarchische Datenstruktur

- .0 *System* = *Table*
- .1 *Table* =  $Key \xrightarrow{m} (Properties \times Arglist \times Type)$
- .2 *Properties* =  $Name \xrightarrow{m} Value$
- .3 *Arglist* =  $Value^*$
- .4 *Type* = ...
- .5 *Key* = ...
- .6 *Name* = ...
- .7 *Value* = ...

Entwerfen Sie für *System* eine gleichwertige Datenstruktur *System1*, die geeignet ist, in einem relationalen Datenbanksystem abgespeichert zu werden. Dabei wird angenommen, daß *Type*, *Key*, *Name* : und *Value* unstrukturierte Wertebereiche sind. In der Transformation werden kein nil-Werte benötigt.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Gibt es Konsistenzbedingungen für *System1*, so daß ein Wert auch einen Wert aus *System* repräsentiert? Wenn ja, welche sind dies?

1) .....

2) .....

3) .....

4) .....

Geben sie, wenn nötig, eine Invariante für *System1* an, mit der die Konsistenzbedingungen präzisiert werden:

.....

.....

.....

.....

.....

.....

.....

.....



**Aufgabe 2:**

Gegeben sei die folgende rekursive Datenstruktur für Ausdrücke

.0  $Expr = Op \times Args$

.1  $Op = Char^*$

.2  $inv-Op(op) \triangleq \text{len } op \leq 30$

.3  $Args = [ Expr ] \times [ Expr ]$

Entwerfen Sie für  $Expr$  eine gleichwertige Datenstruktur  $Expr_1$ , die geeignet ist, in einem relationalen Datenbanksystem abgespeichert zu werden.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Auf der Datenstruktur *Expr* seien folgende Funktionen definiert:

.0 *evalOp* : *Op* × *Value* × *Value* → *Value*

.1 *eval* : [*Expr*] → *Value*

.2 *eval*(*e*)  $\triangleq$

.3 if *e* = nil

.4 then []

.5 else

.6 let (*op*, *arg1*, *arg2*) = *e* in

.7 let *v1* = *eval*(*arg1*) in

.8 let *v2* = *eval*(*arg2*) in

.9 *evalOp*(*op*, *v1*, *v2*)

Über die Arbeitsweise von *evalOp* ist nichts weiter bekannt, außer daß das Resultat vom Typ *Value* ist, wobei hier angenommen werden kann, daß *Value* wie folgt definiert ist:

.10 *Value* = Char\*

Geben Sie eine gleichwertige Funktion *eval1* an, die auf dem neuen Datenmodell arbeitet:



### Aufgabe 3:

Gegeben seien folgende Datenstrukturen für ein hierarchisches Dateisystem:

- .0  $Fsys$  =  $RootInode \times InodeTable$
- .1  $RootInode$  =  $Inode$
- .2  $InodeTable$  =  $Inode \xrightarrow{m} Entry$
- .3  $Inode$  =  $\mathbb{N}_0$
- .4  $Entry$  =  $Data$
- .5  $Data$  =  $File \mid Dir$
- .6  $File$  =  $Byte^*$
- .7  $Byte$  =  $\{0, \dots, 255\}$
- .8  $Dir$  =  $Fname \xrightarrow{m} Inode$
- .9  $Fname$  =  $Char^*$
- .10  $Path$  =  $Fname^*$
- .11  $DateTime$  =  $\mathbb{N}_0$

Erweitern Sie die Datenstruktur so, daß auch *Softlinks* in dem System modelliert werden können.

Die geänderten und/oder zusätzlichen Datentypen:

.....

.....

.....

.....

.....

.....

.....

Erweitern Sie das Modell so, daß zu jedem Eintrag ein Attribut für das Datum des letzten Lesens und Schreibens vorhanden ist.

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

Schreiben Sie zwei Funktionen *searchFsys* und *searchTable*

.12 *searchFsys* : *Fsys* × *Path* → [*Inode*]

.13 *searchTable* : *Inode* × *InodeTable* × *Path* → [*Inode*]

die zu einem (absoluten) *Pfad* in einem *Fsys* den zugehörigen *Inode* berechnen oder im Fehlerfall nil zurückgeben. Verwenden Sie hierfür das Ausgangsdatenmodell.

Tip: die eigentliche Arbeit wird von der zweiten Funktion übernommen.

Beschreiben Sie als erstes umgangssprachlich mögliche Fehlersituationen:

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



