
Aufgaben zur Klausur **Softwaredesign** im SS 2012 (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.

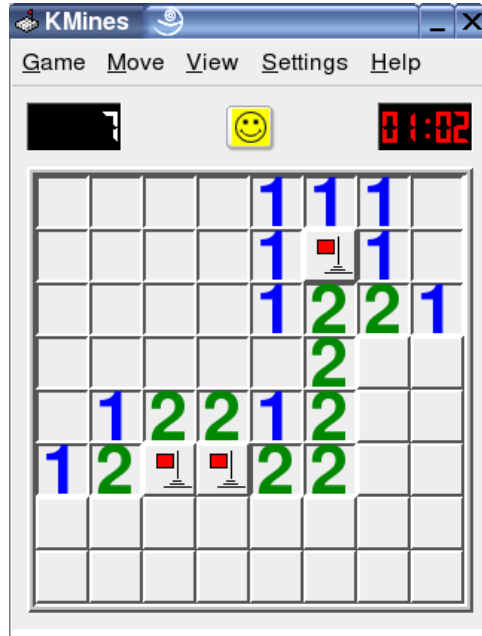
Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg!

Diese Klausur besteht einschließlich dieses Deckblattes aus 7 Seiten.

Aufgabe 1:

Für das Spiel MineSweeper (und Varianten davon) ist ein Datenmodell für die Spielzustände auf dem Feld zu modellieren. Das Bild zeigt eine Spielsituation:



In diesem Spiel geht es darum, auf einem $n \times m$ großen Spielfeld sogenannte Bomben zu finden, ohne diese versehentlich aufzudecken. Der Zustand eines Spiel kann aufgeteilt werden in die Information, die vor dem Start des Spiels berechnet wird und in die Daten, die vom Spieler eingegeben werden.

Der erste Teil enthält pro individuelm Feld die Information, ob eine Bombe darin plziert wurde oder, wenn nicht, wie viele Bomben in den angrenzenden Feldern gelagert sind.

Im zweiten Teil wird jedem einzelnen Feld zugeordnet, ob dieses Feld aufgedeckt worden ist, ob es noch nicht aufgedeckt wurde, oder ob es mit einer Fahne als Bombenfeld markiert worden ist.

Entwickeln Sie ein Datenmodell in abstrakter Syntax in Haskell Notation für die Beschreibung eines Spielzustands. Verwenden Sie bitte pro Typdefinition nur einen Typkonstruktor.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Aufgabe 2:

Gegeben seien die folgende Datentypen:

- .0 `type NodeId` = ...
- .1 `type NodeLabel` = ...
- .2 `type EdgeLabel` = ...
- .3 `type Graph` = `Map NodeId Succs`
- .4 `type Succs` = `Set NodeId`

Mit dem Datentyp *Graph* werden gerichtete Graphen mit Werten aus *NodeId* modelliert.

Entwickeln Sie aus diesem Model einen Datentypen *LabGraph* für gerichtete Graphen, deren Knoten markiert sind mit einem Wert aus *NodeLabel*.

.....

.....

.....

Entwickeln Sie einen Datentypen *Lab2Graph* für gerichtete Graphen, deren Kanten markiert sind mit einem Wert aus *EdgeLabel*.

.....

.....

.....

Entwickeln Sie einen Datentypen *Lab3Graph* für gerichtete Graphen, bei denen sowohl die Knoten als auch die Kanten markiert sind.

.....

.....

.....

Aufgabe 3:

Für die Verarbeitung von Baumstrukturen wird typischerweise das Kompositum–Muster eingesetzt. Geben Sie das OMT–Diagramm für die grundlegende Struktur des Kompositum–Musters an.

Mit welchen Mustern verarbeitet man Kompositum–Objekte typischerweise

1)

2)

3)

Aufgabe 4:

Klassifizieren Sie das Fabrikmethoden-Muster.

.....
.....

Welches ist der Zweck des Musters?

.....
.....

Wo ist das Muster typischerweise anwendbar?

- 1)
- 2)
- 3)

Geben Sie das Klassendiagramm für das Fabrikmethoden-Muster an:

Aufgabe 5:

Entwickeln Sie ein Datenmodell in Form einer abstrakten Syntax für ein hierarchisches Dateisystem.

Ein Dateisystem in dieser Aufgabe ist ein Verzeichnis, in dem Namen Einträge zugeordnet sind. Einträge können von unterschiedlicher Art sein. Eine Ausprägung sind einfache Dateien. Eine zweite Form ist ein (Unter-)Verzeichnis, eine dritte Form ein symbolischer Verweis (symbolic link), der aus einem absoluten Pfad besteht. Ein Pfad bestimmt einen Eintrag in dem hierarchischen Dateisystem.

Alle Einträge besitzen zusätzlich eine Menge von Attributen, wie zum Beispiel Zugriffsrechte und Zeitstempel. Diese Menge von Attributen soll nicht fest vorgegeben sein, sondern es soll eine variablen Anzahl von Attributen möglich sein.

Das Datenmodell in Form einer abstrakten Syntax in Haskell:

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