



Klausur zu „Einführung in die Softwaretechnik“

Sommersemester 2007
Datum: 02.08.2007

Hinweise:

- Jedes Blatt ist in der Kopfzeile mit Ihrem Namen und Ihrer Matrikelnummer zu versehen.
- Die Aufgaben sind in den dafür vorgesehenen Freiräumen oder auf den Rückseiten der Blätter zu lösen. Sollte der Platz nicht ausreichen, so können Zusatzblätter verwendet werden. Diese sind dann ebenfalls mit Namen und Matrikelnummer zu kennzeichnen.
- Außer (dokumentenechten) Schreibgeräten (kein Bleistift o.ä.; außerdem kein Rot oder Grün) sind keine weiteren Hilfsmittel erlaubt.
- Die Klausur besteht aus insgesamt 4 Aufgaben auf 12 Seiten.
- Die Klausur ist bestanden, wenn 29 von 58 Punkten erreicht wurden.
- Die Bearbeitungszeit beträgt 120 Minuten.

Viel Erfolg!

Name: (in Druckschrift)	_____
Matr.Nr.:	_____
Unterschrift:	_____

Aufgabe	Max. Punkte	Korrektur		Einsicht	
		Punkte	Kürzel	Punkte	Kürzel
1	8				
2	18				
3	14				
4	18				
Σ	58				



Name:

Matr.Nr.:

Aufgabe 1

Wissensfragen

8 Punkte

- (a) Nennen Sie drei verschiedene Produktmodellierungsprinzipien und erläutern Sie diese kurz (max. 3 Sätze).

- (b) Nennen Sie drei Qualitätseigenschaften von Programmsystemen und erläutern Sie diese kurz (max. 3 Sätze).

--



Name:

Matr.Nr.:

- (c) Erläutern Sie die Begriffe „funktionale Abstraktion“ und „Datenabstraktion“, und nennen Sie die Eigenschaften von funktionalen und Datenabstraktionsmodulen.

**Aufgabe 2****Anforderungsspezifikation****18 Punkte**

Die Online-Auktionsplattform *HudsonBay* bietet seinen Kunden die Möglichkeit Artikel über das Internet zu versteigern.

(a) Entity-Relationship-Diagramm

Erstellen Sie ein Entity-Relationship-Diagramm zur Beschreibung des Datenmodells der *HudsonBay* Auktionsplattform. Sie entnehmen die folgenden Angaben aus dem bereits erstellten Lastenheft:

- Zu einem Kunden werden Login-Name, Passwort und Bankverbindung gespeichert.
- Zu den einzelnen Angeboten werden die Bezeichnung und eine Beschreibung des Artikels gespeichert.
- Ein Angebot wird von genau einem Kunden erstellt. Ein Kunde kann beliebig viele Angebote erstellen.
- Ein Kunde kann beliebig viele Angebote beobachten. Ein Angebot kann von beliebig vielen Kunden beobachtet werden.
- Ein Kunde kann auf beliebig viele Angebote bieten. Dabei wird die Höhe des jeweiligen Gebots gespeichert.
- Endet ein Angebot, für das mindestens ein Gebot abgegeben wurde, so wird für das Höchstgebot ein Kaufvertrag abgeschlossen. Zu diesem Kaufvertrag werden Datum und Zahlungsweise abgespeichert.
- Zu dem Kaufvertrag wird gespeichert, welcher Kunden den Artikel kauft, und welcher Kunde den Artikel verkauft.
- Wurde ein Kaufvertrag abgeschlossen, so können zu diesem Vorgang Bewertungen erstellt werden. Beide Vertragspartner können optional den jeweils anderen bewerten. Zu jeder Bewertung werden die abgegebene Punktzahl und eine Erläuterung gespeichert.

10 Punkte



Name:

Matr.Nr.:

Lösung Aufgabe 2a:



(b) Datenflussdiagramm

Erstellen Sie ein Datenflussdiagramm zur Beschreibung des Funktionsmodells der *HudsonBay* Auktionsplattform. Sie entnehmen die folgenden Angaben aus dem bereits erstellten Lastenheft:

- Ein Kunde, der einen Artikel versteigern möchte, kann ein neues Angebot erstellen. Dabei muss er die Artikelangaben in das System eingeben. Es wird ein Datensatz im Datenspeicher für Angebotsdaten erstellt. Dabei werden die Daten des Kunden aus dem Datenspeicher für Kundendaten gelesen.
- Um gewünschte Artikel unter den verfügbaren Angeboten zu finden, steht den Kunden eine Suchfunktion zur Verfügung. Der Kunde gibt Suchkriterien ein und erhält aus den vorhandenen Angeboten ein den Kriterien entsprechendes Suchergebnis.
- Auf ein mit der Suchfunktion gefundenes Angebot kann der Kunde ein Gebot abgeben. Im Datenspeicher für Angebotsdaten werden zu dem Angebot die Höhe des Gebots sowie der Kunde, welcher das Gebot abgegeben hat, gespeichert.
- Endet ein Angebot, und entsteht ein Kaufvertrag zwischen dem anbietenden Kunden und dem höchstbietenden Kunden, so werden die Vertragsdaten im Datenspeicher für Angebotsdaten abgelegt. Die beteiligten Kunden werden über den Vertragsabschluss benachrichtigt.
- Nach Abschluss eines Kaufvertrags haben die beteiligten Kunden die Möglichkeit sich gegenseitig zu bewerten. Es wird ein Datensatz im Datenspeicher für Kundendaten erstellt, wobei die abgegebenen Bewertungsdaten und die Daten des zugehörigen Kaufvertrags berücksichtigt werden.

8 Punkte



Name:

Matr.Nr.:

Lösung Aufgabe 2b:

**Aufgabe 3****Entwurfsspezifikation****14 Punkte**

Das online Poker-Portal „Best Bluff“ bietet die Möglichkeit Poker Turniere im Internet auszutragen. Die Anwendung speichert die Spielerdaten während eines Turniers in einem Array. Die Elemente des Arrays enthalten jeweils Daten zu einem Spieler. Das Array ist sortierbar.

(a) Teilsystementwurf:

Erstellen Sie für den beschriebenen Teil des Softwaresystems (Hauptanwendung und sortiertes Array) eine Softwarearchitektur. Fassen Sie dabei alle notwendigen Module für das sortierte Array in einem Teilsystem zusammen, das von der Hauptanwendung verwendet wird. Geben Sie für jedes Modul die Art des Moduls an. Definieren Sie die erforderlichen Beziehungen zwischen den Modulen bzw. zwischen dem Teilsystem und den Modulen. Überlegen Sie dabei unter anderem, welche Schnittstellen von enthaltenen Modulen zu der Schnittstelle des Teilsystems beitragen. Verwenden Sie die Notation für Architekturen aus der Vorlesung. Geben Sie für jeden Baustein des Teilsystems und für das Teilsystem selbst die Entwurfsentscheidungen in einem Satz an.

8 Punkte



(b) Algebraische Spezifikation für das Array:

Arrays enthalten Elemente des generischen Typs `elem`. Die Indizes eines Arrays sind durch eine untere und eine obere Schranke beschränkt (diese können durch die Operationen `first` und `last` ermittelt werden). Der Zugriff auf einzelne Elemente eines Arrays erfolgt über die Funktion `read`, die einen ganzzahligen Index als Parameter erhält. Wird versucht, auf ein Element außerhalb des Wertebereichs für die Indizes zuzugreifen, ist der Rückgabewert `undefined`. Die Funktion `create` erhält die Schranken für die Indizes als Parameter und liefert ein Array zurück, dessen Werte mit `undefined` initialisiert sind. Die Funktion `assign` weist dem Element an dem angegebenen Index den angegebenen Wert zu.

Vervollständigen Sie die unten angegebene algebraische Spezifikation, indem Sie sechs Axiome definieren. Verwenden Sie dabei die angegebenen Variablen.

6 Punkte

```
type ARRAY
functions
  create: integer × integer → array
  assign: array × integer × elem → array
  first: array → integer
  last: array → integer
  read: array × integer → elem
axioms for a: array; x,y,n,m: integer; v: elem let

end type ARRAY
```

**Aufgabe 4****Qualitätssicherung****18 Punkte**

Gegeben sei die folgende Implementierung des Suchalgorithmus `BinarySearch` zur Suche in einem sortierten Feld mit `int`-Werten ohne doppelte Einträge. Bei der Division wird grundsätzlich abgerundet. Die Indizierung des Arrays reicht von 0 bis `length(array)-1`.

- (a) Erstellen Sie den Kontrollflussgraph des hier angegebenen Algorithmus auf der folgenden Seite. Die Nummerierung der Zeilen entspricht der Einteilung in Basic Blocks.

4 Punkte

```
BinarySearch (array, value) {  
01      low = 0  
      high = length(array) - 1  
02      while ( low <= high ) {  
03          p = (high + low) / 2  
04          if ( array[p] > value )  
05              high = p - 1  
06          else if ( array[p] < value )  
07              Low = p + 1  
08          Else  
              return p  
      }  
09      return not_found  
}
```

Verwenden Sie für Ihre Lösung die nachfolgende Seite.

- (b) Geben Sie zu Aufgabenteil (a) Testfälle an, so dass Sie eine vollständige Zweigüberdeckung erreichen. Notieren Sie in der untenstehenden Tabelle zu jedem Testfall das Feld (es genügen drei Komponenten), den Suchwert und den sich daraus ergebenden Programmpfad. Die Anzahl der Zeilen entspricht nicht der Anzahl der benötigten Testfälle.

6 Punkte

Array	Value	Pfad



Name:

Matr.Nr.:

Lösung Aufgabe 4a:

00

10



Wir betrachten nun Black-Box-Tests für ein anderes Beispiel. Gegeben sei die Funktion `encrypt(string passphrase, int offset)` zur einfachen Verschlüsselung von Passwörtern. Als zu verschlüsselnde Passwörter sind nur Zeichenfolgen bestehend aus 1 bis 8 Kleinbuchstaben des lateinischen Alphabets (ohne Umlaute und ‚ß‘) erlaubt, wie z.B. „abc“. Der zweite Parameter ist ein `integer`-Wert zwischen 1 und 25, der angibt, um wie viele Stellen (modulo 26) im Alphabet jeder Buchstabe verschoben wird. Mit dem Wert 3 würde das obige Passwort zu „def“ verschlüsselt.

- (c) Geben Sie 7 Äquivalenzklassen für den Black-Box-Test an. Nummerieren Sie die Klassen durch.

4 Punkte

	Gültige Äquivalenzklassen	Ungültige Äquivalenzklassen
Eingabestring passphrase		
Integer Wert offset		

- (d) Geben Sie Testfälle an mit denen Sie alle Äquivalenzklassen überdecken.

4 Punkte

Eingabestring passphrase	Integer-Wert offset	Überdeckte Äquivalenzklassen