

Matrikelnr.

--	--	--	--	--	--	--	--

Name

Vorname

KLAUSUR: Entwurf und Implementierung von Informationssystemen (32561)
TERMIN: 08.09.2017, 09:00–11:00 Uhr
PRÜFER: Univ.-Prof. Dr. rer. pol. habil. S. Strecker

Aufbau und Bewertung der Klausur

Aufgabe	1	2	3	Summe
Maximal erreichbare Punktzahl	28	46	26	100
Erreichte Punktzahl				

DATUM:

NOTE:

Allgemeine Hinweise



Tragen Sie bitte jetzt Ihre Matrikelnummer, Ihren Namen und Vornamen auf dem Deckblatt ein. Versehen Sie bitte zusätzlich jede Seite mit Ihrer Matrikelnummer (oben rechts).

Hinweise zur Bearbeitung

Für die Bearbeitung der insgesamt 3 Klausuraufgaben auf den folgenden 13 Seiten dieser Klausur stehen Ihnen 120 Minuten zur Verfügung.

1. Außer Schreibgeräten sind keine Hilfsmittel zugelassen.
2. Die Lösungen müssen in den vorgesehenen Raum auf den Aufgabenblättern eingetragen werden.
3. Notizen können auf den Rückseiten der Aufgabenblätter gemacht werden. Diese Anmerkungen werden in die Bewertung *nicht* einbezogen.
4. Bei Beendigung der Klausur müssen das Deckblatt und die Aufgabenblätter abgegeben werden. Trennen Sie bitte einzelne *nicht* Blätter ab.

Viel Erfolg!

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Aufgabe 1 (Objektorientierter Entwurf)**(28P)**

a) Erläutern Sie in eigenen Worten, was unter der im Lehrbrief dargestellten „abstrakten Operation“ zu verstehen ist. Erörtern Sie anschließend in eigenen Worten die im Lehrbrief dargestellten Konzepte „abstrakte Klasse“ und „konkrete Klasse“. Verdeutlichen Sie zusätzlich den Zusammenhang von abstrakten Operationen und abstrakten Klassen mit Hilfe einer Skizze. **(8P)**

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

b) Geben Sie an, ob die nachfolgend aufgeführten Aussagen zutreffen oder nicht. Tragen Sie hierzu jeweils in dem vorgegebenen Kreis ein **"R"** für richtig oder ein **"F"** für falsch ein. Für diese Aufgabe gibt es maximal 20 Punkte. Die Bestehensgrenze dieser Aufgabe von 10 Punkten erreichen Sie mit 6 richtigen Antworten. **(20P)**

Aus dem Framework-Ansatz resultiert, dass in einer Anwendung zwar eventuell mehrere Frameworks, jedoch nur eine Klassenbibliothek verwendet wird.

Gilt für die Schichten einer Anwendung das Prinzip der asymmetrischen Kooperation und der unidirektionalen Abhängigkeit, so gilt dies auch für die Komponenten innerhalb einer Schicht untereinander.

Die Architektur einer Anwendung beschreibt ihre Struktur, d. h. ihre Gliederung in Komponenten und deren Beziehungen.

Innerhalb eines Prozesses können mehrere Threads ablaufen, die jeweils auf einen separaten Speicherbereich für Code und Daten zugreifen.

Eine zentrale Aufgabe des objektorientierten Entwurfs einer Anwendung ist die Spezifikation ihrer Architektur.

Unter der detaillierten Spezifikation einer Assoziation im objektorientierten Entwurf versteht man ihre vollständige Modellierung im Klassendiagramm, wozu unter anderem die Festlegung der Kardinalitäten und der Navigation gehören.

Charakteristisch für den objektorientierten Ansatz ist, dass bei dem Übergang von der Analyse zum Entwurf kein konzeptueller Bruch auftritt.

Das Proxy-Muster bewirkt, dass der Zugriff auf ein Objekt mittels eines zugehörigen Stellvertreter-Objekts gesteuert wird.

Die Einsparung der Versendung von Nachrichten/Botschaften stellt ein entscheidendes Mittel zur Erhöhung der Laufzeiteffizienz einer Anwendung dar.

Bidirektionale Assoziationen sind aufgrund geringerer Kohäsion einfacher zu implementieren und weniger fehleranfällig als unidirektionale Assoziationen.

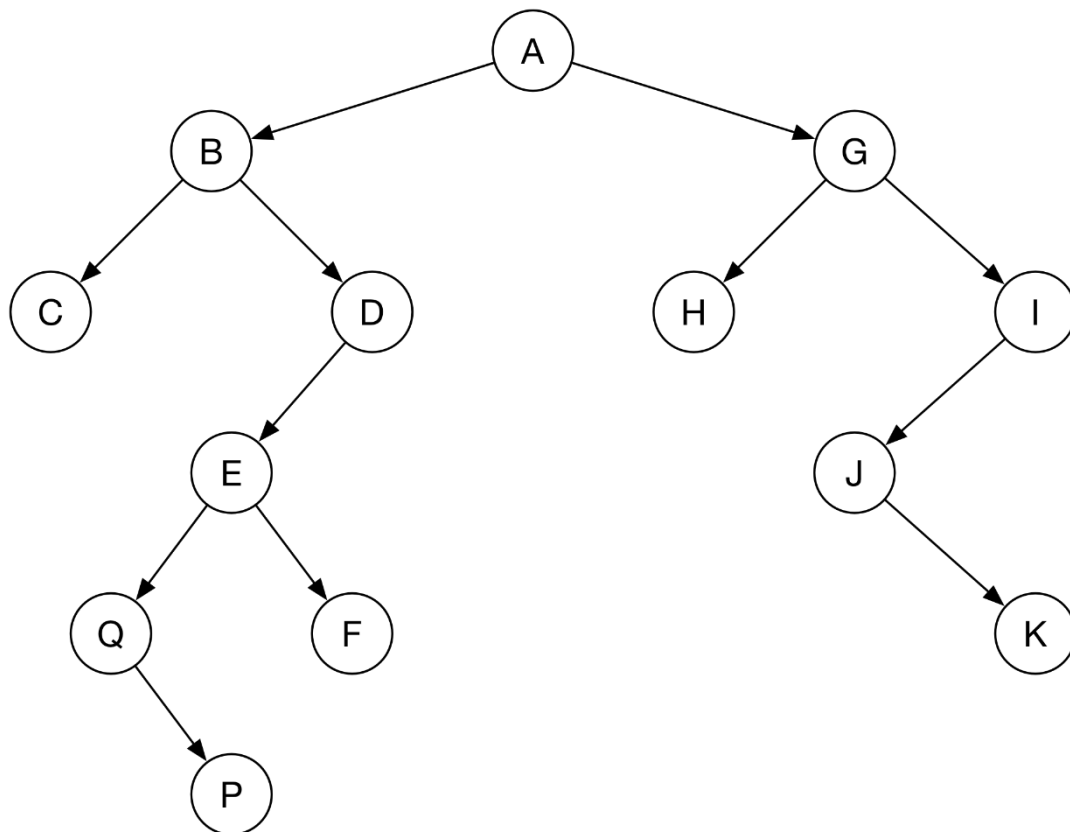
Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Aufgabe 2 (Algorithmen & Datenstrukturen)**(46P)**

In der gesamten Aufgabe 2 wird von Ihnen erwartet, dass Sie die im Lehrbrief dargestellte, an PASCAL angelehnte Pseudocode-Notation *ausnahmslos* anwenden. Für das Algorithmieren mit diesem Pseudocode stehen damit die spezifischen Konzepte von PASCAL zur Verfügung, d. h. verschiedene einfache und zusammengesetzte Datentypen, Konstrukte der strukturierten Programmierung und das Prozedurkonzept. Alle Teilaufgaben sind als Codefragmente in der im Lehrbrief dargestellten Pseudocode-Notation zu erstellen. Andere Pseudocode-Notationen oder Programmiersprachen werden *nicht* bewertet.

a) Mit dem Begriff „Traversieren“ wird das Durchlaufen sämtlicher Knoten eines Baumes in einer bestimmten Reihenfolge bezeichnet. In der Regel wird mit dem Traversieren die Bearbeitung vieler oder aller Knoten bzw. Datenobjekte verbunden sein. Für Binärbäume eignen sich insbesondere auch rekursive Traversierungsalgorithmen.

a1) Geben Sie die Reihenfolge des Ansprechens der Knoten für den Fall des Traversierens in (1) *Postordnung*, (2) *Präordnung* und (3) *Symmetrischer Ordnung* an. **(12P)**



(1)	
(2)	
(3)	

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

a2) Für einen weiteren Binärbaum wurde die Reihenfolge des Ansprechens der Knoten für den Fall des Traversierens in *Präordnung* bereits durch einen Kommilitonen gelöst. Rekonstruieren Sie den zugehörigen Binärbaum aus der gegebenen Reihenfolge: **(6P)**

Reihenfolge: {A, B, C, D, E, Q, F, P, L, M, N, O, R}

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

b) Was ist unter dem FIFO-Prinzip zu verstehen und welche Alternative zum FIFO-Prinzip können Sie nennen und anhand der Funktionsweise kurz erläutern? **(4P)**

c) Erläutern Sie in eigenen Worten, welche Funktion die DISPOSE-Anweisung hat. **(2P)**

d) Bei der sequentiellen Speicherung einer Schlange (engl. „Queue“) belegen die verwalteten Objekte einen zusammenhängenden Bereich eines Feldes. Je ein Index markiert das *front*- und das *rear*-Element. Im Folgenden sollen Prozeduren zum Einfügen, Entfernen und Lesen von Auftragsdaten (ordnern, company, reviser) entworfen werden. Die Aufträge werden in der Reihenfolge ihres Eingangs abgearbeitet. Sortier- und Suchvorgänge sind ausgeschlossen. Zugrunde gelegt seien folgende Datenvereinbarungen:

```
DATA
  CONST n = 200;
  TYPE
    ORDER = RECORD
      orderno : ARRAY [1..8] OF CHAR;
      company : ARRAY [1..20] OF CHAR;
      reviser : ARRAY [1..6] OF CHAR;
    END;
  QUEUE = ARRAY [1..n] OF ORDER;
  INDEX = [0..n];
VARIABLE
  qqueue : QUEUE;
  front, rear : INDEX;
  entries : INTEGER;
  newelement, frontelement : ORDER;
  full, empty : BOOLEAN;
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

d1) Für die Prozedur `append` (Einfügen) steht Ihnen folgender Pseudocode zur Verfügung. Der einzufügende Auftrag soll mit dem Parameter `neuelement` übergeben werden. Ein Einfügen ist nur möglich, wenn die Schlange weniger als n Objekte enthält. Weist jedoch bei nicht voller Schlange der Index `rear` auf das n -te Element, so müssen sämtliche Objekte erst zur `front`-Seite hin verschoben werden, bevor das Einfügen erfolgen kann. **(10P)**

```
PROCEDURE append ( neuelement : ORDER;  
                  VARIABLE front, rear : INDEX;  
                  VARIABLE entries : INTEGER;  
                  VARIABLE full : BOOLEAN;  
                  VARIABLE qqueue : QUEUE;
```

```
DATA
```

```
    VARIABLE i, length : INDEX;
```

```
BEGIN
```

```
END append;
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

d2) Entwerfen Sie die Prozedur `remove` (Entfernen) zum Entfernen eines Elements in der Schlange. Eine Entnahme ist nur bei nichtleerer Schlange möglich. Nach der Entnahme eines Elements wird der Index auf das neue *front*-Element umdirigiert und die Anzahl der Einträge wird um Eins reduziert. **(6P)**

```
PROCEDURE remove ( VARIABLE frontelement : ORDER;  
                   VARIABLE front : INDEX;  
                   VARIABLE entries : INTEGER;  
                   VARIABLE empty : BOOLEAN;  
                   VARIABLE qqueue : QUEUE);
```

```
BEGIN
```

```
END remove;
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

d3) Entwerfen Sie die Prozedur `read` (Lesen). Beachten Sie, dass diese Prozedur lediglich dem Lesen des *front*-Elements und nicht dem Entfernen dienen soll. **(6P)**

```
PROCEDURE read ( VARIABLE frontelement : ORDER;  
                 VARIABLE empty : BOOLEAN;  
                 front : INDEX;  
                 entries : INTEGER;  
                 qqueue : QUEUE);
```

```
BEGIN
```

```
END read;
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Aufgabe 3 (Programmieren in C)**(26P)**

a) Eine Gruppe Studierender möchte zum Zweck des Sparens von Versandkosten gemeinsam bei einem Online-Händler bestellen. Erweitern Sie die untenstehende Vorgabe so, dass jeder Aufruf der `addItem()`-Funktion den Artikel in einer einfach verketteten Liste speichert. Diese Liste soll am Ende des Programms auf dem Bildschirm formatiert ausgegeben werden, nachdem alle Artikel in der Liste gespeichert sind. Die Reihenfolge der Ausgabe ist irrelevant. In einem ersten Schritt sollen Sie die `struct`-Definition der Listenelemente festlegen und Hilfsfunktionen zum Einfügen in die Liste und zum Ausgeben der Liste am Ende programmieren. **(10P)**

```
#include <stdlib.h>

typedef struct litem_t
{

} litem;

typedef struct llist_t
{
    struct litem_t *item;
    struct llist_t *next;
} llist;

llist* addItem(llist* alist, char* ordeno, char* name, float price, int count, float sum)
{

}

}
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

```
void main(int argc, char** argv)
{

llist *alist;
alist = addItem(NULL, „AT 1001 C“, „Pear iPhone Quad 64GB“, 499.00, 1, 499.00);
alist = addItem(alist, „ZT 1012 D“, „Pear iPhone Octa 128GB“, 799.00, 1, 799.00);

}

void dumpList(llist* alist)
{

}

}
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

b) Welche Ausgaben sind bei den formatierten Ausgaben mit printf() zu erwarten? **(3P)**

Anweisung	Ausgabe
Printf(„%d“, 20);	
Printf(„%2d“, 2000);	
Printf(„%.2f“, 20.0f);	
Printf(„%6.2f“, 123.4567);	
Printf(„%e“, 20.0f);	
Printf(„%7.2f“, 20.0f);	

c) Welche Ausgabe liefert das folgende Programm? **(3P)**

```
#include <stdio.h>
```

```
struct person
{
    char name[30];
    long int knr;
};
```

```
void main ()
{
    static struct person kunde[4] = {"Schmitz", 20123, "Mueller", 82765, "Adams", 98761};

    printf("\n%s", kunde[0].name);
    printf("\n%c", kunde[2].name[0]);
    printf("\n%i", strlen(kunden[1].name));
    printf("\n%li", kunde[1].knr);
    printf("\n%li", kunde[3].knr);
    printf("\n%li", kunde[1].name[5]);
}
```

Ausgabe:	
----------	--

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

d) Entwerfen Sie ein Programm zur Berechnung des Binomialkoeffizienten zweier natürlicher Zahlen n und k . Die Funktion `binomial(int n, int k)` soll die Binomialkoeffizienten *rekursiv* berechnen. Der Binomialkoeffizient ist allgemein folgendermaßen definiert: **(10P)**

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Für den Fall, dass $k > n$ ist, gilt:

$$\binom{n}{k} = 0$$

Falls $n = 0$ oder $k = 0$ ist, gilt

$$\binom{n}{k} = 1$$

Aufgrund der Umformung mit dem Pascalschen Dreieck gilt:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Diese mathematische Funktion kann für die Erstellung des C-Programms verwendet werden. Nachdem Sie die Funktion `binomial(int n, int k)` entworfen haben, geben Sie den Binomialkoeffizienten in der Hauptroutine des C-Programmes aus und lesen Sie die Übergabewerte für die Funktion ein. Die Bedingungen sollen dem Benutzer über eine Bildschirmausgabe angezeigt werden.

```
#include <stdio.h>
```

```
void main(void)
```

```
{
```

```
}
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

```
int binomial(int n, int k)
{
```

```
}
```

Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.