

Matrikelnr.

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

Name _____

Vorname _____

KLAUSUR: Entwurf und Implementierung von Informationssystemen (32561)
TERMIN: 10.03.2017, 09:00–11:00 Uhr
PRÜFER: Univ.-Prof. Dr. rer. pol. habil. S. Strecker

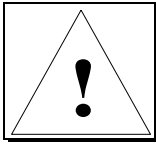
Aufbau und Bewertung der Klausur

Aufgabe	1	2	3	Summe
Maximal erreichbare Punktzahl	30	40	30	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 16 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 nicht einzelne Blätter ab.

Viel Erfolg!

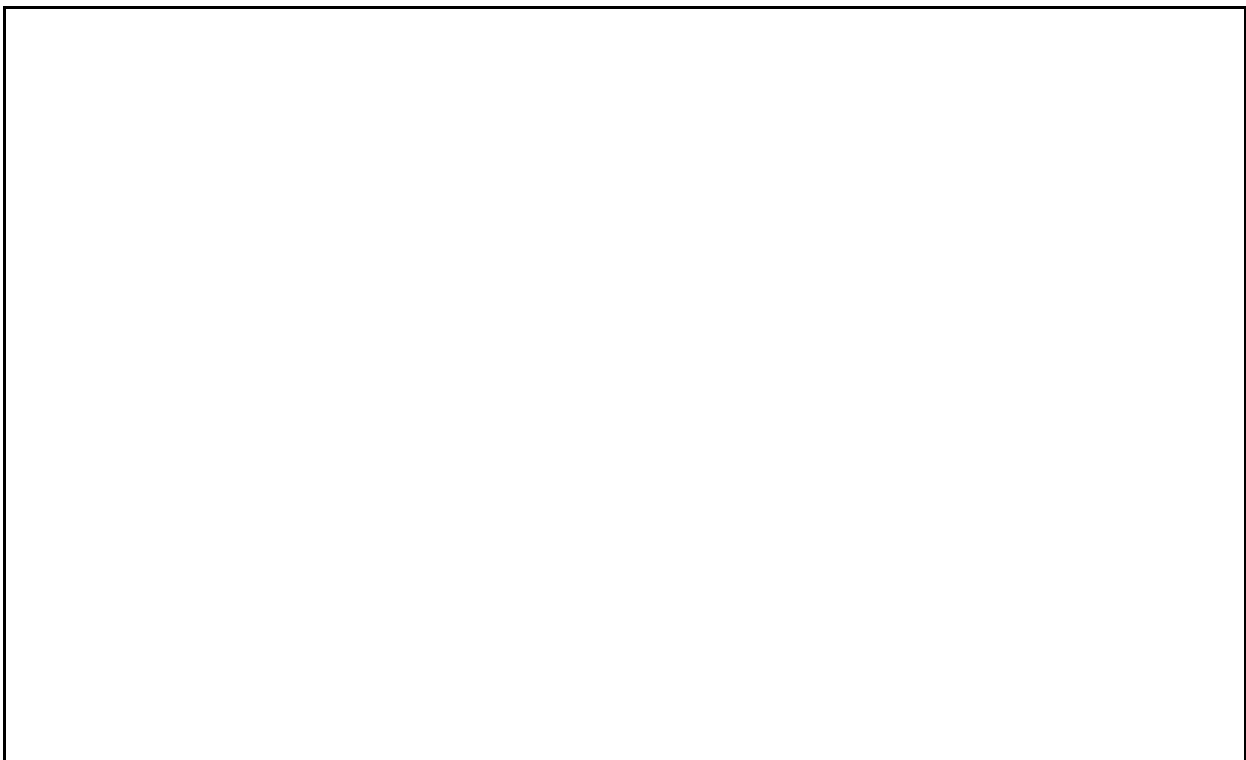
Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, 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)**(30P)**

a) Erläutern Sie in eigenen Worten, was unter dem im Lehrbrief dargestellten „Singleton-Muster (singleton)“ zu verstehen ist. Erörtern Sie dazu zunächst den Zweck und anschließend die Anwendbarkeit und Vorteile. **(5P)**



b) Erläutern Sie, warum der Einsatz des im Lehrbrief dargestellten „Beobachter-Musters (observer pattern)“ eine geeignete Alternative für die Verwendung zwischen GUI- und Fachkonzept-Schicht ist. **(5P)**



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.

c) 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)**

Bei einer Schichten-Architektur gestattet die Separierung des fachlichen Kerns einer Anwendung in einer Fachkonzept-Schicht die kontinuierliche Weiterentwicklung des OOA-Modells zum OOD-Modell der Fachkonzept-Schicht.

Eine Assoziation modelliert eine Gesamtheit von direkten Objektverbindungen zwischen zwei Objekten. Assoziationen bilden die Grundlage der Kooperation von Objekten verschiedener Klassen.

Enthält eine Klasse mindestens eine abstrakte Operation, so können von dieser Klasse Objekte erzeugt werden

Das Proxy-Muster ist ein objektbasiertes Erzeugungsmuster. Der Zugriff auf ein Objekt wird mittels eines zugeordneten Stellvertreter-Objekts, genannt Proxy-Objekt, gesteuert.

Bei einer Fünf-Schichten-Architektur werden die GUI-Schicht und die Fachkonzept-Schicht zusätzlich in je zwei Schichten zerlegt. Die GUI-Schicht ist damit ausschließlich für die Präsentation der Anwendungsdaten zuständig.

Charakteristisch für Schichten-Architekturen ist unter anderem, dass zwischen den Schichten einer Anwendung unidirektionale Abhängigkeitsbeziehungen bestehen.

Ein Garbage-Kollektor dient dem automatischen Schließen von Dateien, die bei der Ausführung eines Programmes nicht mehr benötigt werden.

Dynamischer Polymorphismus kann die Laufzeit im Vergleich zum statischen Binden in gewissem Umfang erhöhen.

Container-Klassen müssen für alle abstrakten Klassen angelegt werden.

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

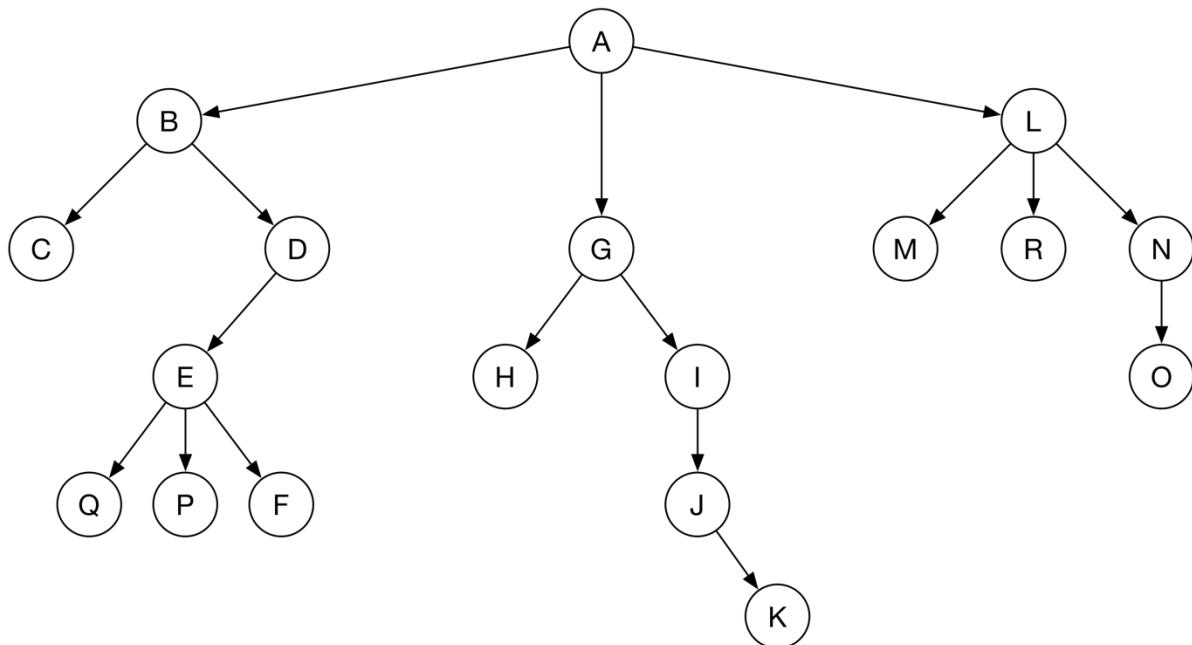
Diese Klausur ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, 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)**(40P)**

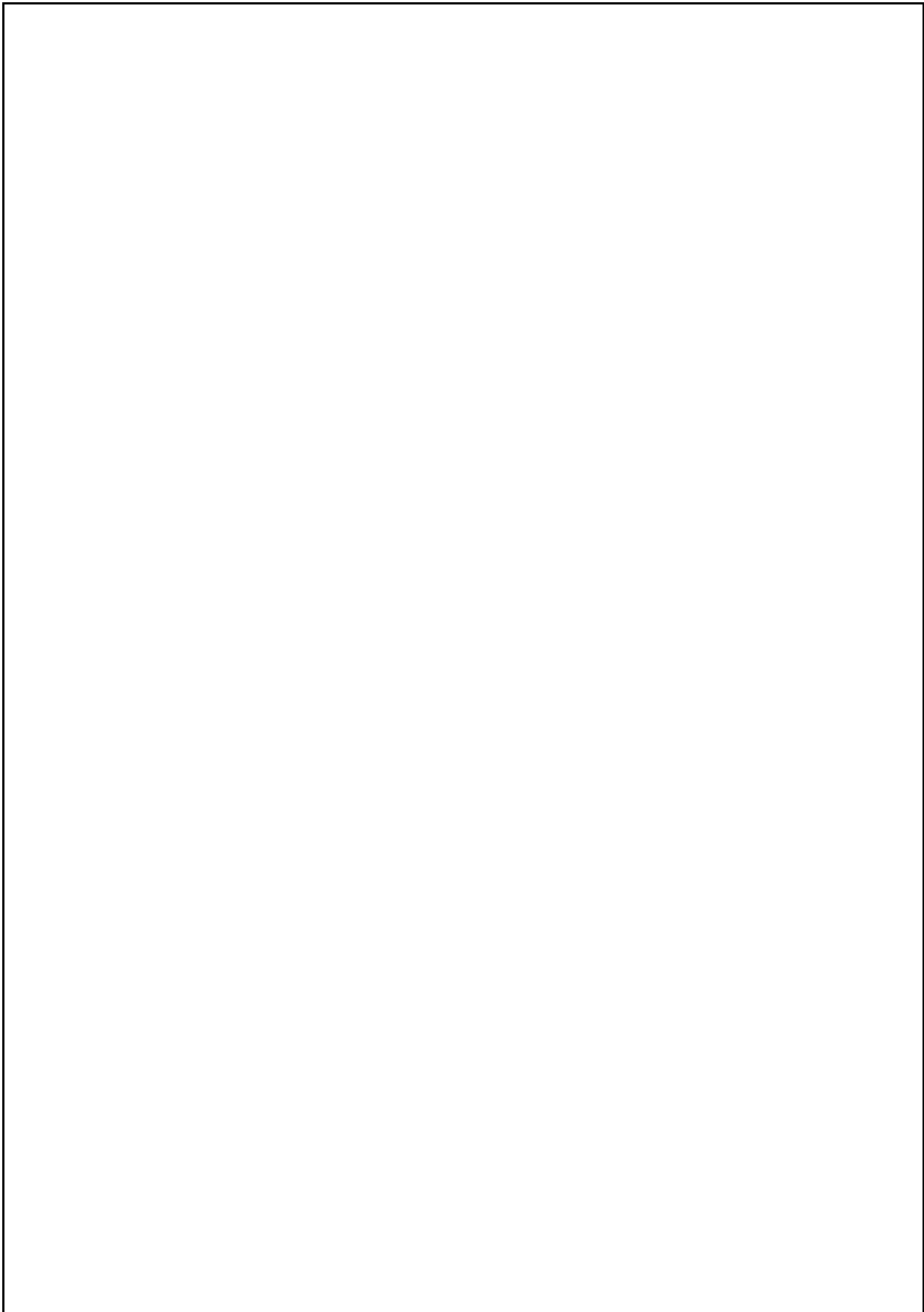
In der gesamten Aufgabe 2 wird von Ihnen erwartet, dass Sie die im Lehrbrief dargestellte, an PASCAL angelehnte Pseudocode-Notation *ausnahmslos* anwenden. Für das Algorithmieren mit diesem Pseudocode stehen damit die spezifischen Konzepte von PASCAL zur Verfügung, d. h. verschiedene einfache und zusammengesetzte 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.

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 Traversierungsverfahren.

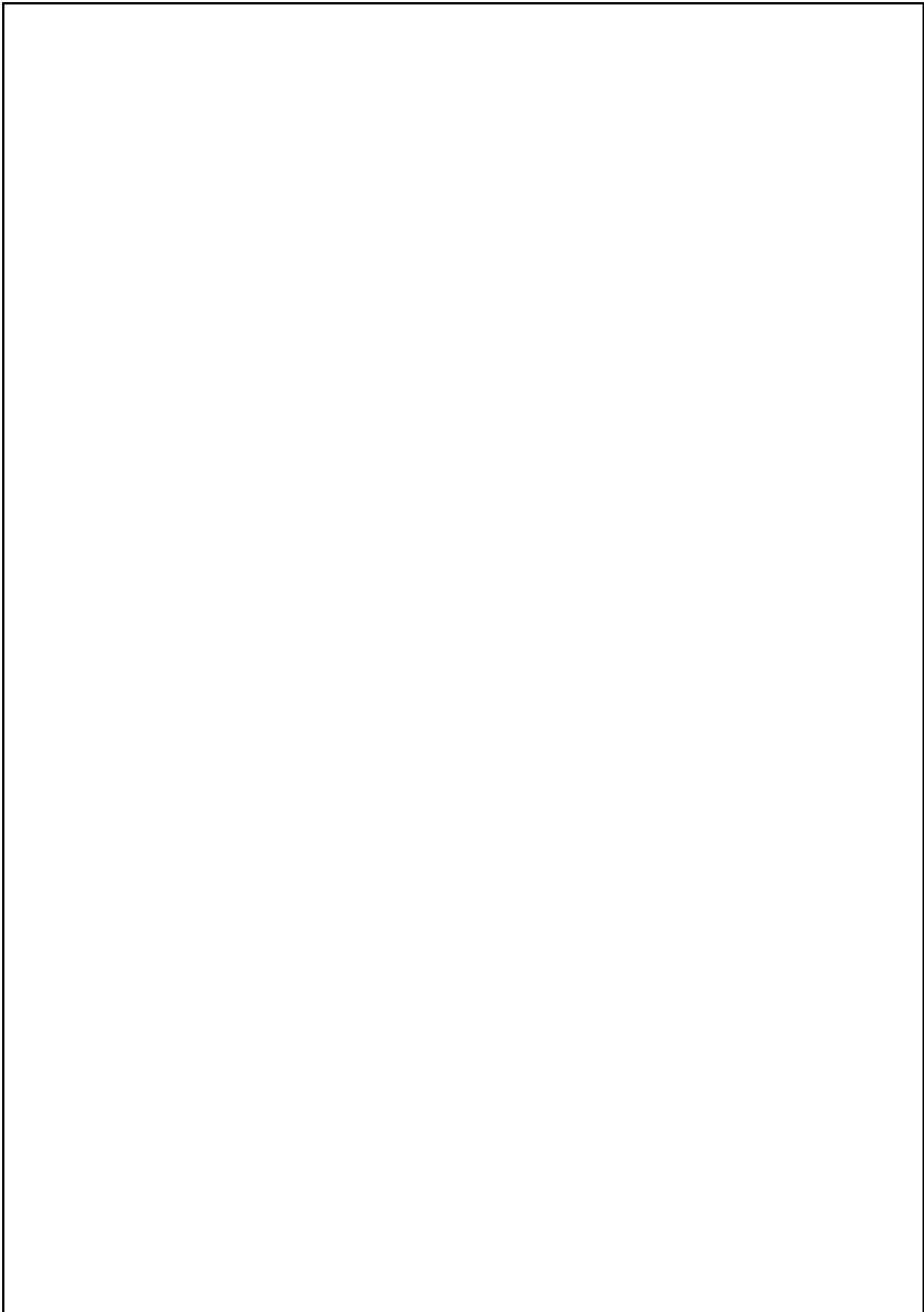
a) Wandeln Sie den nachfolgend dargestellten k-nären Baum, mit einer Ordnung von $k = 3$, in einen Binärbaum um. Anschließend sollen Sie die Reihenfolge des Ansprechens der Knoten für den Fall des Traversierens in *Symmetrischer Ordnung* angeben. **(20P)**



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.



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.



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) Betrachtet sei ein Sortieralgorithmus für sequentiell gespeicherte Objekte. Die Tabelle unten zeigt die Schritte eines solchen Algorithmus sowie die Ausgangs- und Zielfolge. **(10P)**

b1) Benennen Sie den Algorithmus und erläutern Sie das Vorgehen des Sortierens.

	Ausgangsfolge							
Ausgangszustand :	42	51	13	48	92	16	09	68
1. Schritt :	42	51	13	48	92	16	09	68
2. Schritt :	42	51	13	48	92	16	09	68
3. Schritt :	13	42	51	48	92	16	09	68
4. Schritt :	13	42	48	51	92	16	09	68
5. Schritt :	13	42	48	51	92	16	09	68
6. Schritt :	13	16	42	48	51	92	09	68
7. Schritt :	09	13	16	42	48	51	92	68
8. Schritt :	09	13	16	42	48	51	68	92

Zielfolge

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.

c) Welche Traversierungsform ist für die Auflösung eines binären (Such-)Baumes geeignet und was ist dabei zu beachten? Begründen Sie Ihre Antwort. **(4P)**

d) Mittels zusätzlicher Zeiger können weitere Verkettungsformen genutzt werden, wie beispielsweise die Ringverkettung, Doppelverkettung und Ankerkettung. Gegenüber der sequentiellen Speicherung linearer Datenstrukturen weist diese verkettete Speicherung eine Reihe von Vorteilen, aber auch Nachteilen auf. Der Einsatz der verketteten bzw. sequentiellen Speicherung hängt zudem von der jeweiligen Anwendung ab. Geben Sie im Folgenden die Vor- und Nachteile sowie Maßgaben der Anwendung an. **(6P)**

Vorteile der verketteten Speicherung:

Nachteile der verketteten Speicherung:

Aspekte der Nutzung einer Anwendung, die die Wahl der Speicherungsform (verkettet oder sequentiell) beeinflussen:

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)**(30P)**

In der gesamten Aufgabe 3 wird von Ihnen erwartet, dass Sie die im Lehrbrief dargestellte Programmiersprache C ausnahmslos anwenden. Für die Implementierung mit dieser Programmiersprache stehen Ihnen damit die spezifischen Sprachkonzepte von C zur Verfügung.

a) Für eine Folge von reellen Messwerten v_1, v_2, \dots, v_n mit $n \geq 1$, ist der Mittelwert \bar{m}_n gemäß

$$\bar{m}_n = \frac{1}{n} \cdot \sum_{i=1}^n v_i$$

zu berechnen. In dem folgenden Programmfragment finden Sie innerhalb der main-Funktion einen Vektor mit 5 Werten, deren Mittelwert zweimal bestimmt werden soll. Zum einen mit einer nicht-rekursiven Funktion `ermittle_mw()` und zum anderen mit einer rekursiven Funktion `ermittle_mw_rek()`. Für beide Funktionen existieren bislang nur die Funktionsköpfe. Ergänzen Sie den Funktionsrumpf für beide Funktionen an den dafür vorgesehenen Stellen. Hinweis: Nutzen Sie bei der Formulierung der rekursiven Berechnung die Beziehung: **(10P)**

$$\bar{m}_n = \frac{1}{n} \cdot (m_{n-1} \cdot (n-1) + v_n)$$

```
/* --- Programm zur Mittelwertbestimmung --- */

#include <stdio.h>
#define LAENGE 5

void main(void)
{
    double werte[LAENGE] = {12.3, 3.5, 15.0, 8.2, 7.9};
    double resultat;
    int laenge = 5;

    resultat = ermittle_mw(werte, laenge-1);
    printf("\nResultat 1: %.2Lf", resultat);

    resultat = ermittle_mw_rek(werte, laenge-1);
    printf("\nResultat 2: %.2Lf", resultat);
}
```

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.

c) Geben Sie stichwortartig jeweils zwei Unterschiede zwischen der `while`-Schleife und der `do-while`-Schleife in C an. **(4P)**

d) Ersetzen Sie in dem folgenden Programm die Kette von `if-else`-Anweisungen durch eine `switch-case`-Anweisung mit gleicher Bedeutung. **(4P)**

```
#include <stdio.h>

void main(void)
{
    char operator;
    float operand1, operand2;

    printf("\nOperation eingeben (+,-,*): ");
    scanf("%c", &operator);

    printf("\n1. float-Operand: ");
    scanf("%f", &operand1);
    printf("\n2. float-Operand: ");
    scanf("%f", &operand2);

    if (operator == '+')
        printf("\n %f + %f = %f", operand1, operand2, operand1 + operand2);
    else if (operator == '-')
        printf("\n %f - %f = %f", operand1, operand2, operand1 - operand2);
    else if (operator == '*')
        printf("\n %f * %f = %f", operand1, operand2, operand1 * operand2);
    else printf("\n Operation nicht zugelassen!");
}
```

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.

```
/* --- switch-case-Anweisung anstelle von if-else-Kette --- */
```

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.

e) Der Getränkeautomat einer Mensa in Hagen soll neu programmiert werden. Bisherige Versuche, den Automaten neu zu konfigurieren, sind fehlgeschlagen. Entsprechend sollen Sie ein C Programm entwickeln, welches folgende Aspekte berücksichtigt: Es ist eine Funktion zu entwickeln, welche das eingeworfene Geldstück überprüft. Gültige Geldstücke sind 0.05, 0.10, 0.20, 0.50, 1.00 und 2.00 Euro. Ist das Geldstück nicht gültig, wird eine Meldung ausgegeben und der Betrag wird *nicht* vom zu zahlenden Betrag abgezogen. Die Funktion gibt nur den Wert *wahr* oder *falsch* zurück. Dies kann mit den Rückgabewerten 1 und 0 realisiert werden. Zusätzlich soll beim Bezahlvorgang berücksichtigt werden, dass die Benutzerin den Vorgang jederzeit mit 0 abbrechen kann. Der Benutzerin wird dann ihr bisher eingeworfenes Geld zurückgegeben (dies kann durch eine Bildschirmausgabe mit dem jeweiligen Betrag erreicht werden). Folgende Hauptroutine und Funktionen stehen Ihnen zur Verfügung: **(12P)**

```
#include<stdio.h>

// Funktionen
int auswahlGetraenkeSorte();
int eingabeMenge();
float ermittlePreis(int sorte);
float bezahlen(float zuZahlen);
int pruefeGeldstueck(float muenze);
void ausgabeGetraenke(int sorte, int menge);

int main(){
    printf(„Mensa Getraenke Automat v2.0\n\n“);

    int sorte, menge;
    float preis, zuZahlen, gezahlt;

    sorte = auswahlGetraenkeSorte();
    preis = ermittlePreis(sorte);
    menge = eingabeMenge();
    zuZahlen = preis * menge;
    gezahlt = bezahlen(zuZahlen);

    if(gezahlt >= zuZahlen){
        ausgabeGetraenke(sorte, menge);
        printf(„\n\nBitte entnehmen Sie Ihr(e) Getraenke.\n“);
    }else {
        printf(„\n\nBezahlvorgang abgebrochen.\n“);
        printf(„Ausgabe von %.2f Euro.\n“, gezahlt);
    }
    return 0;
}
```

1) Entwerfen Sie die Funktion `pruefeGeldstueck(float muenze)`. Die Funktion soll prüfen, ob ein „korrektes“ Geldstück eingeworfen wurde. Verwenden Sie für die Implementierung die `switch`-Anweisung. Hinweis: Die `case`-Anweisung kann nicht mit `float`-Werten arbeiten.

2) Entwerfen Sie die Funktion `auswahlGetraenkeSorte()`. Für die Benutzerin stehen folgende Getränke zur Auswahl: Wasser (1), Limonade (2) und Bier (3). Zusätzlich sollen dem Nutzer die Preise der Getränke angezeigt werden: Wasser (0.50€), Limonade (1.20€) und Bier (2.00€). Hinweis: Die Funktion soll die Variable `sorte` zurückgeben.

3) Entwerfen Sie die Funktion `ermittlePreis(int sorte)`, die mit der `switch`-Anweisung realisiert werden soll. Je nach gewählter Sorte soll der Preis zurückgegeben werden.

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.

4) Entwickeln Sie die Funktion `bezahlen(float zuZahlen)`. Innerhalb dieser Funktion wird der Bezahlvorgang eingeleitet. Dies kann durch eine Bildschirmausgabe simuliert werden. Der Benutzerin soll ausgegeben werden, wie sie den Vorgang abrechnen kann und wie viele Münzen noch fehlen. Weiterhin ist darauf zu achten, dass die Münzen korrekt eingeworfen werden, sodass bei zu viel eingeworfenen Münzen der Restbetrag erstattet wird. Dies können Sie ebenfalls über eine Bildschirmausgabe simulieren.

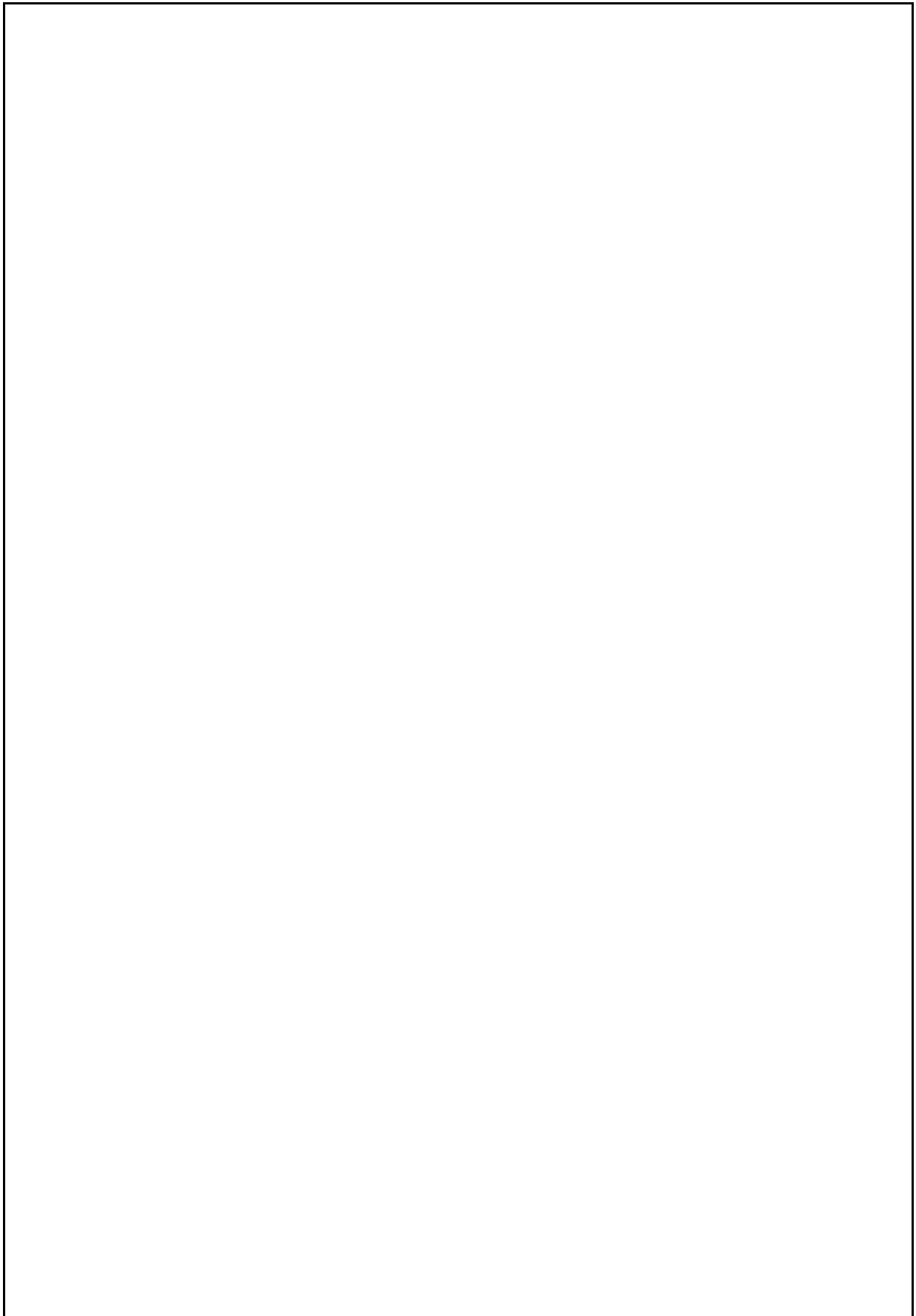
```
// --- Prototypisch implementierte Funktionen

int eingabeMenge();
    int menge = 1;
    printf("\nGeben Sie die gewuenschte Menge ein: ");
    scanf("%d", &menge);
    return menge;
}

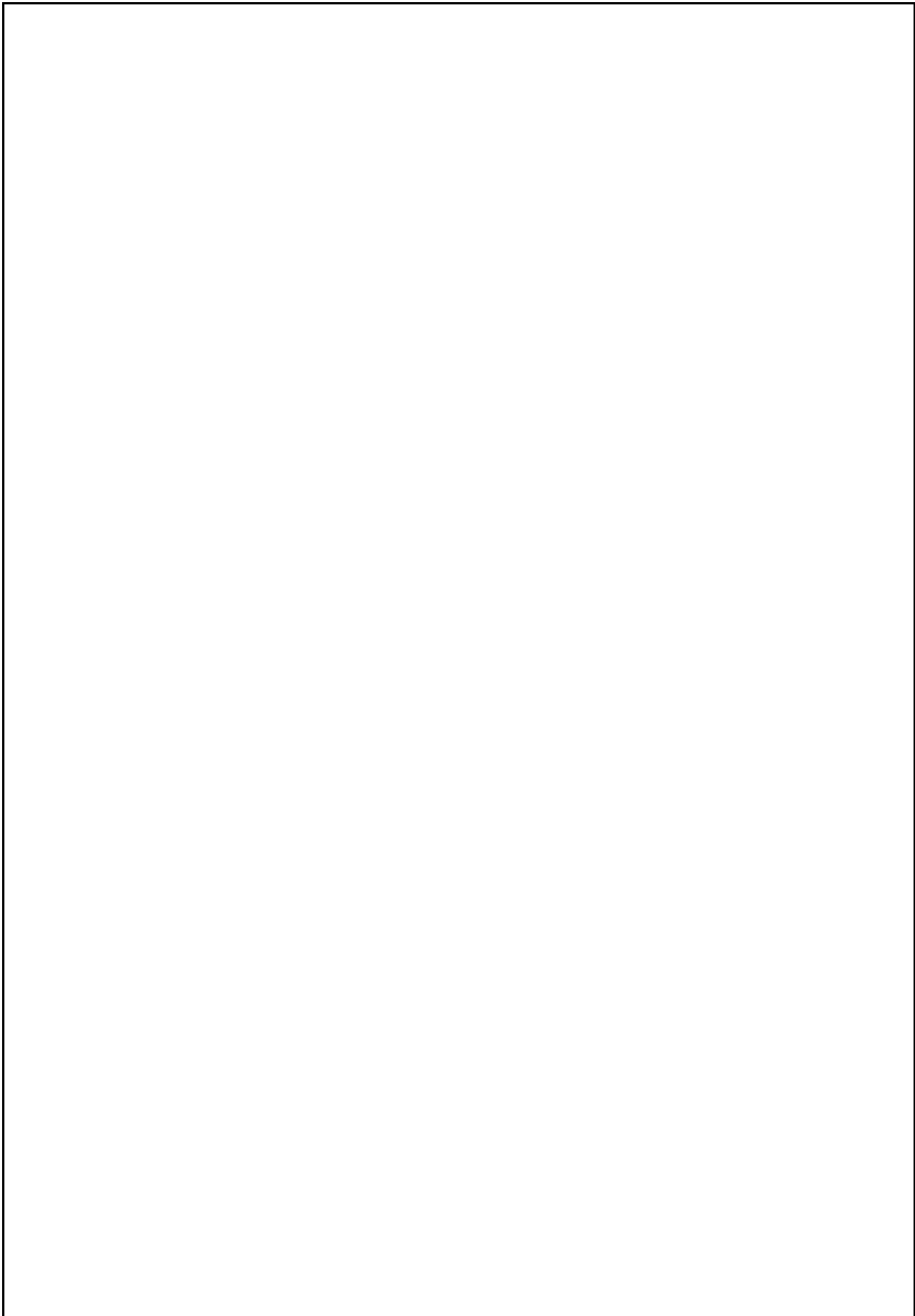
void ausgabeGetraenke(int sorte, int menge){
    printf("\n --- Getraenkeausgabe ---\n");
    int i;
    for(i=0;i < menge;i++){
        printf("\nFlasche %d von %d der Sorte %d wurde ausgegeben.", i+1,
            menge, sorte);
    }
}

// --- Aufgaben (1)-(4) ---
```

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.



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.



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.