

**Sammlung aller Staatsexamensaufgaben der
Prüfungsnummer**

46111

**Programmentwicklung /
Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Herbst 1987

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____ Kennwort: _____ Arbeitsplatz-Nr.: _____	HERBST 1987	46111

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: Informatik (nicht vertieft studiert)

Einzelprüfung: Programmentw./Systempr./Datenbanksys.

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 5

bitte wenden!

Sämtliche Teilaufgaben sind zu beantworten!

Teilaufgabe 1: Programmentwicklung

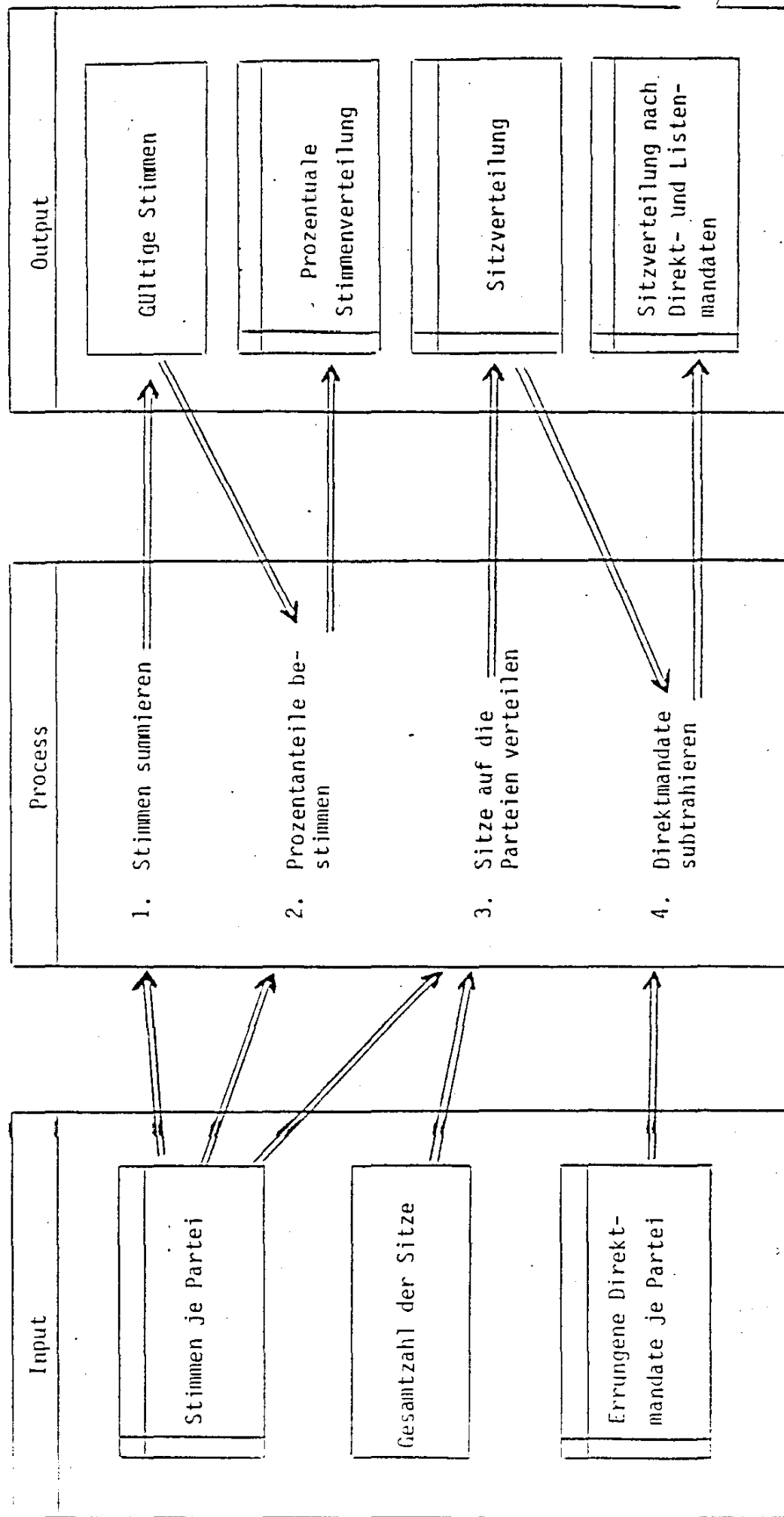
Ein bekanntes Verfahren für die Verteilung der Sitze nach einer Parlamentswahl ist das D'Hondtsche Höchstzahlverfahren. Dabei werden die erzielten Stimmenzahlen der Reihe nach durch die natürlichen Zahlen dividiert:

Divisor	Stimmenzahlen		
1	147173 x	173147 x	18299 x
2	73586,5 x	86573,5 x	9149,5
3	49057,7 x	57715,7 x	
4	36793,3 x	43286,8 x	
5	29434,6 x	34629,4 x	
6	24528,8 x	28857,7 x	
7	21024,7 x	24735,3 x	
8	18396,6 x	21643,4 x	
9	16352,6 x	19238,6 x	
10	14717,3	17314,7 x	
		15740,6	

Sind N Parlamentssitze zu vergeben, so werden diese den N höchsten Zahlen dieser Matrix zugeordnet. Sind beispielsweise 20 Sitze zu vergeben, so sind also die 20 größten Zahlen herauszufinden. Im Beispiel sind diese mit x markiert. Dort entfallen auf die Partei der ersten Spalte 9, auf die der zweiten 10 und auf die der dritten 1 Sitz(e).

Die Abbildung zeigt ein Diagramm zur Gliederung des Programms. Schreiben Sie das Programm in einer beliebigen Programmiersprache! (Geben Sie an, welche!) Schreiben Sie dabei für jede der vier Teilaufgaben eine selbständige Prozedur, wobei die im Diagramm eingetragenen Daten als Parameter auftreten und außer diesen keine Größen gemeinsam benutzt werden sollen!

Auswertung eines Wahlergebnisses



Hinweis:

- a) Zur Vereinfachung sei angenommen, daß in der Matrix keine Zahl doppelt auftritt und keine sog. Überhangmandate entstehen (mehr Direktmandate als die Sitzverteilung ergibt).
- b) Eine elegante Lösung umgeht die Speicherung der Matrix.

Teilaufgabe 2: Systemprogrammierung

Seit ALGOL 60 verlangen die meisten Programmiersprachen vom Betriebssystem einer Rechenanlage eine dynamische Speicherverwaltung. Dabei sind verschiedene Gesichtspunkte zu unterscheiden:

- Die Blockstruktur führt zu einer pulsierenden Speicherverwaltung.
- Zeiger-Konzepte (Beispiel: PASCAL) führen zu einer sog. Halde.

Während sich dies auf die Verwaltung des Speicherbereichs eines Programmes bezieht, entstehen für das Betriebssystem weitere Aufgaben aus der Tatsache, daß sich im Rechner mehrere Programme (Prozesse) gleichzeitig befinden. Um bei der Programmierung der Einzelprogramme darauf keine Rücksicht nehmen zu müssen, verwendet man einen virtuellen Speicherraum, dessen Adressen vom Betriebssystem (mit Hardware-Unterstützung) in reale Adressen umgesetzt werden.

Beschreiben Sie

- a) die auf ein Programm bezogene Speicherverwaltung, die auftretenden Probleme und möglichen Lösungen! Vergessen Sie insbesondere nicht den Fall von Sprunganweisungen, die aus Blöcken hinausführen,
- b) die Speicherverwaltung auf der Basis einer Seiten-Kachel-Tabelle!

Fortsetzung nächste Seite!

Abstr. record
strukt.
inh.
verb.

Teilaufgabe 3: Datenbanksysteme

X

Das Verwalten eines dynamischen Datenbestandes, d.h. einer Ansammlung von Informationen, die durch Einfügen neuer Daten wachsen und durch Löschen schrumpfen kann, und in der bestimmte Informationen gesucht werden, steht und fällt mit einer möglichst optimalen Struktur des Datenbestandes.

Definieren und erläutern Sie Vor- und Nachteile bzgl. der oben unterstrichenen Operationen bei Verwendung folgender Strukturen:

- binärer Baum,
- balancierter binärer Baum,
- Bayer-Baum.

Disk. Daten

Die zuletzt genannte Struktur hat eine besondere Bedeutung, wenn der Datenbestand so groß ist, daß er nicht mehr im Arbeitsspeicher gehalten werden kann. Erläutern Sie die Organisation eines Datenbestandes als Bayer-Baum bei Verwendung heute üblicher peripherer Speicher!

a) Nachteil: kann zu Link werden (78 von Ersetzen v. Tabellen)
Nachteil: zufolge Operationen (Suchen, Löschen, Einfügen)

b) Vorteil: kein Warten bei Suchen oder Einfügen oder Löschen
Vorteil: kein Warten bei Suchen oder Einfügen oder Löschen

Nachteil: kein Warten bei Suchen oder Einfügen oder Löschen
Nachteil: kein Warten bei Suchen oder Einfügen oder Löschen

Vorteil: kein Warten bei Suchen oder Einfügen oder Löschen
Vorteil: kein Warten bei Suchen oder Einfügen oder Löschen

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Frühjahr 1990

Kennzahl: _____

FRÜHJAHR**46111**

Kennwort: _____

1990

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**- Prüfungsaufgaben -****Fach:** Informatik (nicht vertieft studiert)**Einzelprüfung:** Programmentw./Systempr./Datenbanksys.**Anzahl der gestellten Themen (Aufgaben):** 1**Anzahl der Druckseiten dieser Vorlage:** 2Aufgabe

Es soll eine geschlossene Rechenstruktur für geordnete Binärbäume über den ganzen Zahlen Z entwickelt werden. Dabei soll die Ordnung in einem Binärbaum B folgendermaßen festgelegt sein: Sei k ein Knoten von B , $L(k)$ bzw. $R(k)$ der linke bzw. rechte Unterbaum von k in B und $I(k)$ die zu k gehörige Zahl aus Z ; dann gilt:

$$(\forall k \in B)((\forall k_l \in L(k))(I(k_l) < I(k)) \wedge (\forall k_r \in R(k))(I(k) < I(k_r))). \Rightarrow \text{Satz 2.1}$$

Aus dieser Festlegung folgt zugleich, daß die $I(k)$ in B paarweise verschieden sind. Schließlich ordnen wir jedem Binärbaum B in der üblichen Weise seine Höhe $h(B)$ durch folgende rekursive Definition zu: Ein leerer Binärbaum hat die Höhe 0, und die Höhe $h(B)$ eines nichtleeren Binärbaums B mit der Wurzel w ist

$$h(B) := \text{Max}\{h(L(w)), h(R(w))\} + 1.$$

Als Höhe $h(k)$ eines Knotens k von B bezeichnen wir (wieder wie üblich) die Höhe des Unterbaums von B , dessen Wurzel k ist.

- a) Geben Sie in PASCAL-Notation eine geeignete Datenstruktur (durch entsprechende Deklarationen) an, mit der geordnete Binärbäume mit den oben angegebenen Festlegungen dargestellt werden können! Die Knoten sollen dabei sowohl ihre zugeordnete Zahl aus Z als auch ihre Höhe enthalten. Als Anker eines nichtleeren Baums soll ein Verweis (pointer) auf dessen Wurzel dienen; ein leerer Baum soll durch einen mit nil besetzten Verweis als Anker dargestellt werden.

Bevor Sie mit Teilaufgabe b) beginnen, überprüfen Sie, ob die angegebene Datenstruktur für eine möglichst effiziente Durchführung aller in b) geforderten Operationen geeignet ist, und begründen Sie Ihre schließliche Wahl der Datenstruktur im Hinblick auf die in b) vorgesehenen Operationen i) - vi).

INFO 12 - Datenstr.

b) Entwickeln Sie systematisch PASCAL-Prozeduren (bzw. -Funktionen) für die folgenden 6 Operationen an geordneten Binärbäumen! Dabei darf für die Operationen unter ii) - vi) vorausgesetzt werden, daß der jeweilige Binärbaum durch kreiere eingerichtet wurde, und daß er bei Aufruf der jeweiligen Operation die oben angegebenen Integritätsbedingungen (Festlegungen) erfüllt, d.h. daß er in der festgelegten Weise geordnet ist, und daß die Höhen in den Knoten entsprechend der Definition korrekt eingetragen sind.

i) kreiere(t)

Diese Operation richtet einen leeren Binärbaum ein und weist dessen Anker der Variablen t zu.

ii) leer(t)

Diese Operation ist als eine Funktion darzustellen; sie liefert den Wahrheitswert *true*, wenn der Binärbaum mit dem Anker t leer ist; andernfalls liefert die Funktion den Wert *false*.

iii) enthalten(z,t)

Diese Operation ist als eine Funktion darzustellen; sie liefert den Wahrheitswert *true*, wenn der Binärbaum mit dem Anker t einen Knoten mit der Zahl $z \in \mathbb{Z}$ enthält; andernfalls liefert die Funktion den Wert *false*.

iv) speichere(z,t)

Diese Operation fügt in den Binärbaum mit dem Anker t einen Knoten mit der Zahl $z \in \mathbb{Z}$ ein, falls der Binärbaum noch keinen Knoten mit z enthält; andernfalls bleibt der Binärbaum unverändert. Der resultierende Binärbaum muß wieder die Integritätsbedingungen erfüllen; sein Anker muß an t zugewiesen sein.

v) loesche(z,t)

Diese Operation entfernt den Knoten mit der Zahl $z \in \mathbb{Z}$ aus dem Binärbaum mit dem Anker t , falls der Binärbaum einen solchen Knoten enthält; andernfalls bleibt der Binärbaum unverändert. Der resultierende Binärbaum muß wieder die Integritätsbedingungen erfüllen; sein Anker muß an t zugewiesen sein.

vi) avl(t)

Diese Operation ist als eine Funktion darzustellen; sie liefert den Wahrheitswert *true*, wenn der Binärbaum mit dem Anker t ein AVL-Baum ist; andernfalls liefert die Funktion den Wert *false*.

so für AVL-Baum

so für AVL-Baum

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Frühjahr 1995

Prüfungsteilnehm

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Frühjahr

46111

Kennwort: _____

1995

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: Informatik (nicht vertieft studiert)

Einzelprüfung: Programmentw./Systempr./Datenbanksys.

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Geben Sie eine formale Definition an für die Begriffe

- 1.1 Chomsky-Grammatik
- 1.2 Direkte Ableitbarkeit und (allgemeine) Ableitbarkeit
- 1.3 Kontextfreie Grammatik

Aufgabe 2:

- 2.1 Spezifizieren Sie eine Objektart "Rational", die rationale Zahlen darstellen und bearbeiten kann! Mögliche Funktionen und Operationen sollen sein:
Addition, Subtraktion, Multiplikation, Division und Vergleich.
(Die Ergebnisse sollten, wenn möglich, gekürzt werden. Dazu steht Ihnen eine Funktion $ggT(a,b)$ zur Ermittlung des größten gemeinsamen Teilers zur Verfügung)
- 2.2 Schreiben Sie in einer Ihnen geläufigen Programmiersprache einen Modul, der die in 2.1 spezifizierte Objektart implementiert!

Aufgabe 3:

- 3.1 Beschreiben Sie einen der Ihnen bekannten Synchronisationsmechanismen!
- 3.2 Was versteht man unter einer Verklemmung?
- 3.3 Welche 4 Bedingungen sind notwendig, damit eine Verklemmung auftreten kann?
- 3.4 Geben Sie eine Methode an, wie man Verklemmungen vermeiden kann!

Aufgabe 4:**4.1 Lösen Sie das folgende Synchronisationsproblem mit zählenden Semaphoren!**

Die Mensa besitze je einen Behälter mit 100 Löffeln, 100 Gabeln und 100 Messern.

Es gibt nun (neben anderen Gerichten) die drei folgenden Hauptgerichte.

- a) Spaghetti, die von den Studenten mit Gabel und Löffel gegessen werden.
- b) Eintopf mit Würstchen, man braucht dazu einen Löffel und ein Messer (um das Würstchen zu zerschneiden).
- c) Schnitzel, das mit Messer und Gabel gegessen wird.

Skizzieren Sie für die 3 verschiedenen Gerichte je eine Routine, die den Zugriff und die Rückgabe der notwendigen Betriebsmittel Messer, Gabel, Löffel löst (mit zählenden Semaphoren)! (Sie können davon ausgehen, daß die Gegenstände bei der Rückgabe wieder in den entsprechenden Behälter kommen und dabei gereinigt werden)

Beachten Sie bei Ihrer Lösung, daß keine Verklemmungen entstehen und möglichst viele Personen gleichzeitig essen können!

4.2 Beschreiben Sie, wie Sie bei Ihrer Lösung eine Verklemmung vermieden haben!**Aufgabe 5:**

In einem Computersystem mit Mehrprogrammbetrieb gibt es verschiedene Verfahren, die Prozesse den Prozessoren zuzuteilen.

Beschreiben Sie kurz Methode, sowie Vor-, Nachteile und Haupteinsatzgebiet für die Prozessorvergabe-strategien!

- 5.1 RR (Round Robin)
- 5.2 MLFB (Multilevel Feedback)
- 5.3 FCFS (First Come First Serve)

Aufgabe 6:

- 6.1 Erklären Sie die Begriffe "Seite" und "Kachel"!
- 6.2 Wie ist eine virtuelle Adresse aufgebaut (nur reines Paging)?
- 6.3 Skizzieren Sie den Abbildungsmechanismus beim Paging von virtuellen auf physikalische Adressen!
- 6.4 Welcher Hardwarevorgang und welche Softwareroutinen laufen ab, wenn eine Seite angesprochen wird, die nicht im Arbeitsspeicher ist (und der Rechner Paging unterstützt)?
- 6.5 Erklären Sie den Seitenaustauschalgorithmus LRU!

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Herbst 1996

Kennzahl: _____

Herbst**46111**

Kennwort: _____

1996

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: Informatik (nicht vertieft studiert)

Einzelprüfung: Programmentw./Systempr./Datenbanksys.

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Entwerfen Sie ein Programm-Modul zur Verwaltung einer Studentenkartei! Die Kartei soll für jeden Studenten die Einträge Name, Vorname, Geburtsdatum, Adresse, Matrikelnummer, Studienfach und Semesterzahl enthalten!

Das Modul soll folgende Operationen bzw. Funktionen zur Verfügung stellen:

- Eintragen eines Studenten
- Löschen eines Studenten (durch Matrikelnummer identifiziert)
- Ermitteln der Anzahl von Studenten für ein bestimmtes Studienfach
- Berechnung des Mittelwertes der Semesterzahl aller Studenten

- 1.1 Spezifizieren Sie zunächst die einzelnen Operationen und Funktionen! (Im Fehlerfall soll ein Fehlercode geliefert werden.)
- 1.2 Überlegen Sie sich mindestens drei sinnvolle Datenstrukturen zur Speicherung der Studentenkartei, und geben Sie Vor- und Nachteile an!

Aufgabe 2:

- 2.1 Welche Einträge enthält der Dateikatalog (directory) eines Dateisystems?
- 2.2 Skizzieren Sie den Aufbau eines hierarchischen Dateisystems in einem Mehrprogrammsystem (Multiusersystem)!
- 2.3 Beschreiben Sie Möglichkeiten, wie in einem Mehrprogramm-System die Dateien der einzelnen Benutzer vor unerwünschten Zugriffen durch andere geschützt und anderen Benutzern erwünschte Zugriffe ermöglicht werden können!
- 2.4 Geben Sie für ein Ihnen bekanntes Mehrprogramm-Betriebssystem die Realisierung des Zugriffsschutzes an!

Aufgabe 3:

- 3.1 Was versteht man unter einer Verklemmung?
- 3.2 Welche 4 Bedingungen sind notwendig, damit eine Verklemmung auftreten kann?
- 3.3 Geben Sie eine Methode an, wie man Verklemmungen vermeiden kann!

Fortsetzung nächste Seite!

Aufgabe 4:

- 4.1 Formulieren Sie eine Implementierung für zählende Semaphore mit Hilfe des Monitorkonzeptes!
- 4.2 In einem Rechensystem stehen Ihnen als Synchronisationsmechanismus zählende Semaphore zur Verfügung. Sie können beliebig viele Prozesse generieren, die über gemeinsame Variablen Daten austauschen können.

Geben Sie eine Methode an, wie mit diesen Mitteln ein "Oder-Warten" realisiert werden kann, d.h., ein Prozeß möchte auf verschiedene Ereignisse warten und beim Eintreffen eines der Ereignisse entsprechend reagieren!

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Frühjahr 1997

Kennzahl: _____

Frühjahr**46111**

Kennwort: _____

1997

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**- Prüfungsaufgaben -**

Fach: Informatik (nicht vertieft studiert)

Einzelprüfung: Programmentw./Systempr./Datenbanksys.

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

- 1.1 Beschreiben Sie die verschiedenen Parameterübergabemechanismen beim Aufruf von Prozeduren!
- 1.2 Geben Sie für jeden der in 1.1 genannten Mechanismen ein Beispiel aus einer Programmiersprache an!

Aufgabe 2:

Zwei typische Vertreter zusammengesetzter Objektarten sind die Reihung (Feld) und der Verbund. (Record, Struktur)

- 2.1 Beschreiben Sie die beiden Objektarten !
- 2.2 Wie werden die beiden Objektarten in einer Ihnen geläufigen Programmiersprache dargestellt und wie erfolgt der Zugriff auf die einzelnen Komponenten? (Beispiele)

Fortsetzung nächste Seite!

Aufgabe 3:

Entwerfen Sie in einer Ihnen bekannten Programmiersprache eine internationale Adreß-Kartei, die die üblichen Einträge enthält, wie Vorname, Name, Straße, Hausnummer, Stadt, Postleitzahl und Land!

- 3.1 Formulieren Sie die notwendigen Datenstrukturen für eine einzelne Adresse und für eine Liste von Adressen! Achten Sie bitte genau auf die notwendigen Typen der einzelnen Einträge insbesondere im Hinblick auf die internationale Auslegung (z.B. hat eine englische "Postleitzahl" die Form: EH9 2NG)!
- 3.2 Entwerfen Sie Operationen zum Eintragen von neuen Adressen, zum Löschen von Adressen und zum Zählen der Einträge für ein bestimmtes Land!
- 3.3 Entwerfen Sie eine Prozedur, die eine Adresse länderspezifisch druckt für Deutschland und für England, wobei eine englische Adresse folgende Form hat:

Vorname Name

Hausnummer Straße

Stadt Postleitzahl

Aufgabe 4:

- 4.1 Implementieren Sie das Leser-Schreiber-Problem als Prozeßsystem P, mit folgenden Eigenschaften:
Es gibt eine beschränkte Zahl von k Lesern, die alle gleichzeitig lesen dürfen. Es gibt eine beliebige Zahl von Schreibern. Ein Schreiber braucht exklusiven Zugriff zu den Daten. Sie können frei wählen, welches Synchronisationssystem Sie verwenden wollen, ebenso, ob die Schreiber Vorrang haben sollen oder nicht (bitte aber angeben).

Ihre Lösung darf sich nicht verklemmen und darf kein aktives Warten (busy wait) aufweisen.

- 4.2 Beschreiben Sie das von Ihnen verwendete Synchronisationssystem!

Aufgabe 5:

In einem Computersystem mit Mehrprogrammbetrieb gibt es verschiedene Verfahren, die Prozesse den Prozessoren zuzuteilen.

Beschreiben Sie kurz Methode, sowie Vor-, Nachteile und Haupteinsatzgebiet für die Prozessorvergabe-Strategien:

- 5.1 RR (Round Robin),
- 5.2 MLFB (Multilevel Feedback),
- 5.3 FCFS (First Come First Served)!

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Herbst 1997

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
--------------------	----------------	----------------------

Kennzahl: _____

Herbst

46111

Kennwort: _____

1997

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: Informatik (nicht vertieft studiert)

Einzelprüfung: Programmentw./Systempr./Datenbanksys.

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

1. Programmentwicklung

- (1a) Mit folgender Datenstruktur können in C binäre Bäume dargestellt werden:

```
struct Node {  
    struct Node *left, *right; /* linker und rechter Kindknoten */  
    struct Node *parent;      /* Elternknoten */  
    info data;                /* Knoteninhalt */  
};
```

oder in PASCAL:

```
TYPE nodepointer = ^node;  
    node = RECORD left, right: nodepointer;  
                parent:    nodepointer;  
                data:      info  
            END;
```

(Der Datentyp `info` sei geeignet definiert.) Schreiben Sie in der Programmiersprache Ihrer Wahl eine rekursive Prozedur, die als Parameter den Zeiger auf die Wurzel des Baumes erhält und die Elemente in In-Order-Reihenfolge ausgibt! Zur Vereinfachung nehmen Sie an, daß es eine Druckprozedur `print` gibt, die Daten vom Typ `info` ausdrückt!

- (1b) Nehmen Sie nunmehr an, daß in der Datenstruktur nur die Komponenten `left`, `right` und `info` besetzt sind! Schreiben Sie eine rekursive Prozedur, die zu jedem Knoten die `parent`-Komponente bestimmt!
- (1c) Lösen Sie jetzt noch einmal die Teilaufgabe (a), jedoch mit einer iterativen Prozedur, ersetzen Sie also die Rekursion durch eine Schleife!

2. Programmentwicklung

- (2a) Im Zusammenhang mit der axiomatischen Spezifikation von Datenmoduln spielen die von Parnas stammenden Begriffe V-Funktion und O-Funktion eine Rolle. Was versteht man darunter, und wie kann man diese Funktionen einsetzen, um eine Spezifikation als hinreichend vollständig zu erkennen?
- (2b) Seien die Datentypen `Bool` (mit der üblichen Semantik) und `info` als bekannt vorausgesetzt. Auf dem Datentyp `stack` seien die folgenden Operationen definiert:

`EmptyStack`, `IsEmpty`, `GetTop`, `Push`, `Pop`.

Welche dieser Funktionen sind V- und welche O-Funktionen? Geben Sie eine hinreichend vollständige Spezifikation für den Datentyp `stack` in axiomatischer Form an, wobei die formale Semantik der Operationen der durch die Operationsbezeichnungen gegebenen informellen entspricht!

Fortsetzung nächste Seite!

3. Systemprogrammierung

- (3a) Wir betrachten einen Mehrprogrammbetrieb. Welche drei Zustände kann ein aus der Sicht des Benutzers aktiver Prozeß aus der Sicht des Betriebssystems annehmen, und welche Übergänge zwischen diesen Zuständen gibt es?
- (3b) Was versteht man in diesem Zusammenhang unter verdrängungsfreien Prozessorvergabestrategien?
- (3c) Wie arbeitet die Vergabestrategie „Round Robin“, und in welcher Betriebsart spielt sie eine wichtige Rolle?
- (3d) Beschreiben Sie die Multilevel-Feedback-Strategie! Welche Vorteile hat diese Strategie?

4. Datenbanksysteme

In den letzten Jahren ist die objektorientierte Systementwicklung auch auf dem Gebiet der Datenbanken sehr aktuell geworden.

- (4a) Bei der objektorientierten Systementwicklung orientiert sich die Modularisierung an Objekten. Was versteht man in diesem Zusammenhang unter Attributen, Methoden, Kapselung, Klassen, Vererbung und Polymorphismus?
- (4b) Als Motivation für die Einführung der objektorientierten Systementwicklung werden Schlagwörter wie Erweiterbarkeit, Wiederverwendbarkeit und Wartbarkeit genannt. Was versteht man unter diesen Begriffen, wie unterscheiden sie sich, und welche der unter (a) genannten Konzepte tragen in welcher Weise zu diesen Argumenten bei?
- (4c) Im relationalen Datenbankmodell können Objekte nur durch Schlüssel identifiziert werden, also durch eines ihrer Attribute. Im objektorientierten Datenbankmodell läßt sich die Objektidentität eher konventionell durch Zeiger bzw. Namen, aber auch durch Surrogate (identifizierende Attribute) oder abstrakte Objekte erfassen. Erläutern Sie die Unterschiede dieser Techniken am Beispiel einer Schema-Definition für Bücher, wie man sie in einer Bibliotheksdatenbank benötigt!
- (4d) Neben der bereits in (a) angesprochenen Beziehung zwischen Klasse und Unterklassen ist auch die Beziehung zwischen Klasse und Komponentenklassen von Bedeutung. Erläutern Sie Begriffe wie gemeinsames bzw. privates Komponentenobjekt, abhängiges bzw. unabhängiges Komponentenobjekt am Beispiel einer Bibliotheksdatenbank! (Hinweis: Naheliegende Klassen sind z.B. Bücher, Studierende, Verlage, Ausleihungen)

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Herbst 1998

Prüfungsteilnehm.	Prüfungstermin	Einzelprüfungsnummer
-------------------	----------------	----------------------

Kennzahl: _____

Herbst

Kennwort: _____

1998

46111

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Programmentw./Systempr./Datenbanksys.**

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 4

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1 (Programmentwicklung)

Der folgende Datentyp CharList wird zur Darstellung von Worten (Strings) mit maximal 1000 Buchstaben verwendet. Für jedes Wort w des Typs CharList gibt $w.laenge$ die Länge des Wortes an. Die einzelnen Buchstaben von w sind in den Feldkomponenten $w.elemente[1]$, ..., $w.elemente[w.laenge]$ gespeichert. (In den Feldkomponenten $w.elemente[i]$ mit $i > w.laenge$ können beliebige Zeichen gespeichert sein.) Für das leere Wort gilt $w.laenge = 0$.

```
TYPE  Laenge = [0..1000];
      Index = [1..1000];
      CharList = RECORD
          laenge: Laenge;
          eiemente: ARRAY Index OF CHAR
      END;
```

In der Teilaufgabe a) sollen einige Standardprozeduren für Worte erstellt werden. In der Teilaufgabe b) sollen diese Prozeduren zur Lösung eines speziellen Wort-Problems verwendet werden. In Teilaufgabe c) ist die (Ordnung der) Zeitkomplexität aller Prozeduren zu bestimmen. Die Prozeduren sind in der Syntax von Modula oder von Pascal zu schreiben. Die Prozedurköpfe sind im folgenden jeweils in der Syntax von Modula angegeben.

a) Zu erstellen sind folgende Prozeduren:

1. Eine Funktionsprozedur mit Kopf `PROCEDURE length (w: CharList) : Laenge;` die die Länge eines Wortes w liefert.
2. Eine Funktionsprozedur mit Kopf `PROCEDURE get (w: CharList; i: Index) : CHAR;` die das i -te Element eines Wortes w liefert. (Es kann davon ausgegangen werden, daß w nicht leer ist und daß $i \leq \text{length}(w)$ gilt.)
3. Eine reine Prozedur mit Kopf `PROCEDURE insert (VAR w: CharList; c: CHAR; i: Index);` sodaß nach Ablauf der Prozedur das Zeichen c an der i -ten Stelle in das Wort w eingefügt ist (bzw. angehängt ist, falls $i = \text{length}(w)+1$). (Es kann davon ausgegangen werden, daß $i \leq \text{length}(w)+1 \leq 1000$ gilt.)
4. Eine reine Prozedur mit Kopf `PROCEDURE delete (VAR w: CharList; i: Index);` ~~sodaß~~ **sodaß nach Ablauf der Prozedur das i -te Zeichen aus dem Wort w entfernt ist.** (Es kann davon ausgegangen werden, daß w nicht leer ist und daß $i \leq \text{length}(w)$ gilt.)

b) Ein Wort w ist ein "Spiegelwort" genau dann, wenn es von vorne und von hinten gelesen dieselbe Zeichenkette bildet (z. B. OTTO).

1. Schreiben Sie eine nicht-rekursive Funktionsprozedur mit Kopf
`PROCEDURE ist_spiegelwort (w: CharList) : BOOLEAN;`
die feststellt, ob w ein Spiegelwort ist!
2. Schreiben Sie eine rekursive Funktionsprozedur mit Kopf
`PROCEDURE ist_spiegelwort_rek (w: CharList) : BOOLEAN;`
die feststellt, ob w ein Spiegelwort ist!

Fortsetzung nächste Seite!

- c) Bestimmen Sie für jede der in Teil a) und b) erstellten Prozeduren die Größenordnung der Zeitkomplexität im schlechtesten Fall ("worst case") in Abhängigkeit von der Länge n des Wortes w .

Aufgabe 2 (Systemprogrammierung)

Im folgenden sei ein Rechner mit einem Prozessor und einem Mehrprogramm-Betriebssystem (zur Verwaltung mehrerer parallel ablaufender Prozesse) gegeben.

- a) Geben Sie die verschiedenen Zustände an, in denen sich ein Prozeß befinden kann! Erläutern Sie, welche Übergänge zwischen den einzelnen Zuständen eines Prozesses möglich sind, und nennen Sie jeweils eine Ursache für einen Zustandswechsel!
- b) Beschreiben Sie ein Zuteilungsverfahren, mit dem der Prozessor den einzelnen Prozessen zugeteilt werden kann!
- c) Gegeben seien die folgenden Prozeßbeschreibungen zweier parallel ablaufender Prozesse P_1 und P_2 . Jeder Prozeß P_i ($i = 1, 2$) führt in einer nichtterminierenden WHILE-Schleife zunächst eine Anweisungsfolge A_i aus und möchte danach auf ein gemeinsames Betriebsmittel B zugreifen.

```
P1:  WHILE TRUE DO
      A1 ;
      Zugriff auf B
    END;
```

```
P2:  WHILE TRUE DO
      A2 ;
      Zugriff auf B
    END;
```

Die beiden folgenden Teilaufgaben c1) und c2) sind unabhängig voneinander zu bearbeiten!

- c1) Beschreiben Sie, wie man unter Verwendung eines Semaphors die beiden Prozesse so synchronisieren kann, daß sie nicht gleichzeitig auf das Betriebsmittel B zugreifen können, jedoch ihre Anweisungsfolgen A_1 und A_2 parallel ausführen können!
- c2) Die beiden Prozesse sollen so synchronisiert werden, daß sie abwechselnd auf das Betriebsmittel B zugreifen. Geben Sie eine Lösung dieses Problems mit Hilfe von zwei Semaphoren an, bei der Prozeß P_1 als erster auf das Betriebsmittel B zugreift! (Vergessen Sie dabei nicht, die Semaphore geeignet zu initialisieren!)

Aufgabe 3 (Datenbanksysteme)

Eine Bibliothek möchte eine Datenbank anlegen. Ausgangspunkt sind eine Relation "Buch" (für die Bücher der Bibliothek) und eine Relation "Ausleihe" (mit Angaben zu den aktuell ausgeliehenen Büchern). Die Relation "Buch" hat die Attribute

Buch-Nr Titel Autor Verlag

und die Relation "Ausleihe" hat die Attribute

Buch-Nr Entleiher-Nr Ausleihdatum Entleiher-Name Entleiher-Telefon

wobei "Buch-Nr" der Schlüssel jeder Relation ist.

- a) Die Relation "Ausleihe" ist in zweiter Normalform (2NF) aber nicht in dritter Normalform (3NF). Geben Sie jeweils eine Begründung dafür an!
- b) Geben Sie drei Nachteile an, die sich bei der Verwendung der Relation "Ausleihe" ergeben können!
- c) Überführen Sie die Relation "Ausleihe" in zwei Relationen in dritter Normalform, und geben Sie den beiden neuen Relationen jeweils einen Namen!

Die Bibliotheksdatenbank soll nun die Relation "Buch" und die beiden in Teil c) konstruierten Relationen enthalten. Formulieren Sie folgende Anfragen an die Datenbank in SQL:

- d) Finde die Titel aller Bücher, die vom Autor "Wirth" verfaßt wurden.
- e) Finde für jedes ausgeliehene Buch die Buch-Nr, den Titel, den Autor und die Entleiher-Nr.
- f) Finde den Titel und den Autor aller nicht ausgeliehenen Bücher.
- g) Finde den Namen und die Telefonnummer jedes Entleihers, der mehr als 10 Bücher ausgeliehen hat.

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Frühjahr 1999

Prüfungsteilnehmer	Prüfungstermin	Einz	Prüfungsnummer
--------------------	----------------	------	----------------

Kennzahl: _____

Frühjahr

Kennwort: _____

1999

46111

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Programm entw./Systempr./Datenbanksys.**

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1: Programmentwicklung

Mit folgender Datenstruktur können in C doppelt verkettete Listen dargestellt werden:

```
struct Node {  
    struct Node *left, *right; /* linker und rechter Nachbarknoten */  
    int    prior;              /* Prioritaet des Knotens */  
    info   data;              /* Knoteninhalt */  
};
```

oder in PASCAL:

```
TYPE nodepointer = ^node;  
    node = RECORD left, right: nodepointer;  
                prior:      integer;  
                data:       info  
            END;
```

(Der Datentyp info sei geeignet definiert.)

- a) Der Zeiger `currptr` zeige auf ein Element dieser Liste, der Zeiger `actitem` auf ein neues Element. Erläutern Sie mit Hilfe einer Zeichnung, wie die Zeiger umzusetzen sind, damit das neue Element hinter bzw. vor dem durch `currptr` definierten eingefügt wird!
- b) Schreiben Sie in der Programmiersprache Ihrer Wahl die entsprechenden Prozeduren `insertbefore` und `insertafter`!
- c) Sie wollen mit der angegebenen Datenstruktur eine prioritätengesteuerte Warteschlange implementieren. Bei einer einfachen Implementierung fügt `insert` das neue Element hinten (oder vorne) an, und `getnext` durchsucht die gesamte Liste nach dem Element mit der höchsten Priorität und liefert dieses zurück. Führen Sie die Implementierung unter Verwendung der zuvor definierten Prozeduren aus!
- d) Welche besseren Lösungsmöglichkeiten fallen Ihnen ein? Skizzieren Sie eine solche (ohne Implementierung!) und begründen Sie durch grobe Abschätzung der durchschnittlichen Laufzeit, warum diese Lösung besser ist als die in der vorhergehenden Teilaufgabe beschriebene!

Fortsetzung nächste Seite!

Teilaufgabe 2: Datenbanksysteme

Das derzeit am weitesten verbreitete Datenmodell ist das Relationenmodell. Ein Miniaturbeispiel könnte folgendes Aussehen haben:

Name	Geb.dat.	Matr.nr.	Fachrichtung	Prüf.fach	Prüfer	Note
Maier	12.03.74	234567	Informatik	Datenbanken	Wedekind	1,7
Huber	11.02.75	123456	Informatik	Prog.sprachen	Eickel	2,3
Müller	14.04.76	345678	Elektrotechn.	Prozessrechner	Färber	1,7
Huber	11.02.75	123456	Informatik	Theoret. Inf.	Noltemeier	1,0
Maier	12.03.74	234567	Informatik	Prozessrechner	Färber	2,7

- Erläutern Sie die Grundstruktur! (Was versteht man unter Tupeln, Attributwerten, Schlüsseln?)
- In eine Datenbank müssen häufig neue Tupel eingefügt bzw. aus ihr Tupel entfernt werden. Außerdem können sich Attributwerte in Tupeln ändern. Dabei sind Konsistenzbedingungen zu beachten. Was versteht man unter Domänenbeschränkungen, Schlüsselbeschränkungen und referenzieller Integrität?
- Was versteht man unter Normalformen? Durch welche Bedingungen unterscheiden sich die erste, zweite, dritte Normalform vom allgemeinen Fall?
- Transformieren Sie das Miniaturbeispiel so, dass es mindestens die Kriterien der zweiten Normalform erfüllt!

46111

**Programmentwicklung / Systemprogrammierung /
Datenbanksysteme (nicht vertieft)**

Herbst 1999

Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Herbst

Kennwort: _____

1999

46111

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Programmentw./Systempr./Datenbanksys.**

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1 (B-Bäume):

Wir betrachten B-Bäume vom Typ m . In einem solchen Baum speichert also jeder Knoten zwischen m und $2 \cdot m$ Schlüsselwerten.

- a) Geben Sie eine *Typvereinbarung* für einen einzelnen Knoten eines B-Baumes vom Typ $m = 2$ in einer Programmiersprache an, wobei die Schlüsselwerte vom Typ Integer sein sollen.
- b) Geben Sie ein *Programm* - in der Programmiersprache aus a) - zum *LWR-Durchlauf* eines B-Baumes an. Dabei soll für einen Knoten v mit k Schlüsselwerten s_1, \dots, s_k und $k + 1$ Teilbäumen T_1, \dots, T_{k+1} folgende Durchlaufreihenfolge vorliegen:

$$T_1, s_1, T_2, s_2, \dots, T_i, s_i, T_{i+1}, \dots, T_k, s_k, T_{k+1}.$$

Beim Durchlaufen eines Schlüsselwerts s_i soll dieser ausgegeben werden.

Benutzen Sie beispielsweise eine der folgenden Programmiersprachen: Pascal, C, C++, Java.

Teilaufgabe 2 (Lineare Listen):

- a) Implementieren Sie das bekannte Sortiervorgehen *Quicksort* auf einfach verketteten linearen Listen.
- b) Implementieren Sie ein Verfahren zur Invertierung einfach verketteter linearer Listen. Aus einer linearen Liste $L = (a_1, a_2, \dots, a_n)$ soll die inverse lineare Liste $L^i = (a_n, \dots, a_2, a_1)$ erzeugt werden. Dabei darf die alte Listenstruktur von L zerstört werden.

Benutzen Sie beispielsweise eine der folgenden Programmiersprachen: Pascal, C, C++, Java.

Teilaufgabe 3 (SQL-Anfragen):

Gegeben sei eine relationale Datenbank mit den Relationen *Buch*, *Entleiher* und *Ausleihe* mit den Attributmengen U_b , U_e , und U_a :

$$\begin{aligned} U_b &= \{ \text{Buch_Nr}, \text{Buch_Titel}, \text{Standort} \}, \\ U_e &= \{ \text{Entleiher_Nr}, \text{Name}, \text{Vorname}, \\ &\quad \text{Anschrift}, \text{Alter}, \text{Geschlecht}, \text{Hauptfach} \}, \\ U_a &= \{ \text{Entleiher_Nr}, \text{Buch_Nr}, \text{Entleih_Datum}, \text{Rückgabe_Datum} \}. \end{aligned}$$

Fortsetzung nächste Seite!

Geben Sie SQL-Statements an für folgende Anfragen:

- a) Entleiher mit Name, Vorname, Alter und Anschrift
- b) Hauptfächer der Entleiher des Buches „Objektorientierte Datenbanken“
- c) Entleiher mit Name und Vorname sowie den Buchtiteln und den Rückgabedaten ihrer entliehenen Bücher
- d) Anzahl der weiblichen Entleiher zwischen 25 und 30 Jahren
- e) Durchschnittliches Alter der Entleiher der gespeicherten Bücher gruppiert nach Buchtiteln