

**Kennzahl:** \_\_\_\_\_**Kennwort:** \_\_\_\_\_**Arbeitsplatz-Nr.:** \_\_\_\_\_**Herbst  
2007****66114**

---

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

---

**Fach:** **Informatik (vertieft studiert)****Einzelprüfung:** **Datenbank- und Betriebssysteme****Anzahl der gestellten Themen (Aufgaben):** 4 Aufgaben, von denen zwei gemäß untenstehender Auswahlregeln zu bearbeiten sind!**Anzahl der Druckseiten dieser Vorlage:** 10

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr.“ die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

## **Themenschwerpunkt A** **(Datenbanksysteme)**

### **Aufgabe 1:**

#### **1. Allgemeine Fragen**

Geben Sie für jede der folgenden Aussagen an, ob diese richtig oder falsch ist! Begründen Sie Ihre Aussage in jedem Fall!

- a) Jede Relation muss mindestens ein Attribut besitzen.
- b) Ein Primärschlüssel muss immer aus mindestens zwei Attributen zusammengesetzt sein.
- c) Alle Relationen eines Datenbankschemas müssen in der ersten Normalform vorliegen.
- d) Datenbanksysteme können gleichzeitig von maximal 32 Anwendern genutzt werden.
- e) Ein Fremdschlüssel verweist auf genau ein Tupel einer anderen Relation.
- f) Eine Datenbanktransaktion entspricht genau einem SQL-Statement.
- g) Der Name eines Fremdschlüsselattributes darf nicht gleich dem Namen des referenzierten Primärschlüsselattributes sein.
- h) Die logische Datenstruktur einer relationalen Datenbank beschreibt, wo welche Daten auf der Festplatte gespeichert sind.
- i) Alle Attribute einer Relation müssen unterschiedliche Namen haben.
- j) Kein Attribut einer Relation darf den gleichen Namen wie die Relation tragen.

Definieren Sie folgende Begriffe bzw. Abkürzungen im Kontext von Datenbanksystemen!

- k) DML
- l) Schlüsselkandidat
- m) Index
- n) Stored Procedure

#### **2. Relationenmodell und relationale Anfragesprachen**

- a) Relationen werden oft mit Tabellen verglichen. Dieser Vergleich ist jedoch nicht ganz korrekt. Stellen Sie kurz dar, welche Unterschiede zwischen den Konzepten „Tabelle“ und „Relation“ existieren!
- b) Stellen Sie kurz den Unterschied zwischen „Relationenschema einer Relation“ und der „Extension einer Relation“ dar! Illustrieren Sie Ihre Erklärung an einem kleinen Beispiel!
- c) Nennen und beschreiben Sie in jeweils 1-2 Sätzen drei Grundoperationen der relationalen Algebra!
- d) Benennen und beschreiben Sie kurz die prinzipiellen Möglichkeiten der Anfrage-Optimierung!

#### **3. Normalformenlehre**

- a) Erläutern Sie kurz Sinn und Zweck der Normalformenlehre!
- b) Charakterisieren und erläutern Sie an einem kurzen Beispiel die erste, zweite und dritte Normalform!
- c) Stellen Sie kurz dar, welche Korrektheitskriterien man bei der Zerlegung eines Relationenschemas beachten muss!

**Fortsetzung nächste Seite!**

#### 4. Transaktionen und Transaktionsverarbeitung

- a) Nennen und definieren Sie die vier wesentlichen Merkmale einer Datenbanktransaktion!
- b) Was versteht man unter dem sog. Lost-Update Problem? Skizzieren Sie den Sachverhalt an einem kurzen Beispiel!
- c) Um die quasi parallele Ausführung von Transaktionen auf einem Datenbanksystem fehlerfrei zu ermöglichen, verwendet man sog. Sperren. Stellen Sie kurz dar, welche Arten von Sperren es gibt und wie sie zueinander in Beziehung stehen!
- d) Welches Problem kann sich bei der Verwendung von Sperren ergeben? Wie kann man dieses Problem in der Praxis umgehen?

#### 5. ER-Modellierung und Relationenmodell

##### a) ER-Modellierung

Erstellen Sie das Modell zur Organisation einer fiktiven Bundestagswahl in E/R-Notation! Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und – falls nötig – Rollennamen sollen ins Diagramm aufgenommen werden. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist!

##### „Bundestagswahl“

Bei der Bundestagswahl werden **Wahlkreise** durch eine eindeutige Wahlkreisnummer identifiziert und durch einen Namen genauer beschrieben.

In einem Wahlkreis tritt mindestens ein **Kandidat** zur Wahl an, jedoch kann ein Kandidat maximal in einem Wahlkreis antreten. Jeder Kandidat hat einen Namen, der ihn eindeutig identifiziert.

Jeder Kandidat ist Mitglied maximal einer **Partei**, die jedoch mindestens 10 Kandidaten stellen muss. Eine Partei kann vor der Wahl eine Koalitionsaussage zu beliebig vielen anderen Parteien machen. Jede Partei wird durch ihren Namen eindeutig identifiziert und durch ein Regierungsprogramm charakterisiert.

Jede Partei stellt für die Wahl genau eine **Liste** mit Kandidaten auf. Jede Liste muss zu genau einer Partei gehören. Auf einer Liste sind mindestens fünf Kandidaten positioniert. Jeder Kandidat darf maximal auf einer Liste positioniert sein. Eine Liste wird durch eine Nummer gekennzeichnet und verfügt über eine bestimmte Anzahl von Plätzen.

##### b) Relationenmodell

Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. Wie gewohnt, werden dabei Primärschlüssel durch Unterstreichen, Fremdschlüssel durch Überstreichen kenntlich gemacht.

## 6. SQL

### Szenario 1: Tischreservierung

Für die Verwaltung der Reservierungen in einer Datenbank verwendet ein kleines Gourmetrestaurant die folgenden Relationen:

**GAST** (GNR, Name, Vorname, Adresse)

**RESERVIERUNG** (GNR, TNR, Datum, Anzahl)

**TISCH** (TNR, Position, Plätze, Kinderplätze)

Die Primärschlüssel der Relationen sind unterstrichen, **GNR** in **RESERVIERUNG** ist Fremdschlüssel zu **GNR** in **GAST**, **TNR** in **RESERVIERUNG** ist Fremdschlüssel zu **TNR** in **TISCH**. Formulieren Sie die folgenden Datenbankoperationen in SQL!

- Geben Sie den Namen und Vornamen aller Gäste aus, die einen Tisch für den 31.12.2005 reserviert haben!
- Geben Sie eine Liste (**TNR** und **Position**) aller Tische aus, die für mindestens drei Personen am 31.12.2005 reserviert sind!
- Fügen Sie eine neue Reservierung mit folgenden Daten ein:
  - Gastnummer (**GNR**): 17
  - Tischnummer (**TNR**): 3
  - Datum: 31.12.2005
  - Anzahl: 4 Personen
- Ändern Sie die Anzahl der reservierten Plätze in folgender Reservierung von 4 auf 3:
  - Gastnummer (**GNR**): 13
  - Tischnummer (**TNR**): 9
  - Datum: 31.12.2005
- Löschen Sie den Gast mit der Nummer (**GNR**) 27 und *alle* zugehörigen Reservierungen!

### Szenario 2: Beim Paketdienst

Der internationale Paketdienst IPS (International Parcel Service) muss täglich eine Vielzahl von Postsendungen ausliefern und verwendet dazu folgende Relationen in einer relationalen Datenbank:

**POSTKUNDE** (KNR, Name, Vorname, Straße, PLZ)

**SENDUNG** (SID, AbsenderNR, EmpfängerNR, Porto, Einschreiben)

**TOUR** (TNR, Fahrer, Startzeit, Dauer)

**LADELISTE** (TNR, SendungsID, Position)

Es existieren folgende Fremdschlüsselbeziehungen: Das Attribut **SendungsID** der Relation **LADELISTE** ist ein Fremdschlüssel auf das Attribut **SID** der Relation **SENDUNG**. Die Attribute **AbsenderNR** und **EmpfängerNR** der Relation **SENDUNG** sind Fremdschlüssel auf das Attribut **KNR** der Relation **POSTKUNDE**. Die Attribute **KNR** und **PLZ** der Relation **POSTKUNDE**, das Attribut **SID** der Relation **SENDUNG**, sowie das Attribut **TNR** der Relation **TOUR** sind Ganzzahlen. Bei dem Attribut **Startzeit** der Relation **TOUR** handelt es sich um ein Datum, das Attribut **Dauer** wird als Ganzzahl in Minuten angegeben. Bei dem Attribut **Einschreiben** der Relation **SENDUNG** handelt es sich um einen booleschen Wert. Die Datentypen der weiteren Attribute sind dem Kontext zu entnehmen.

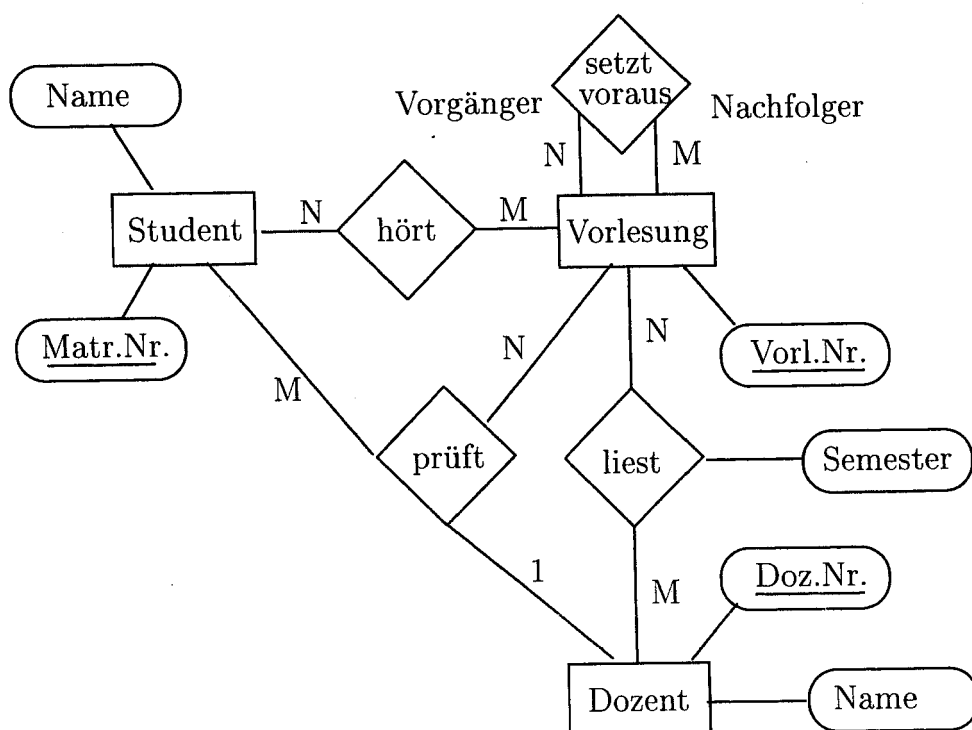
Geben Sie SQL-Anweisungen für folgende Problemstellungen an:

- Erzeugung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints.
- Geben Sie die Namen, Vornamen und Straßen aller Postkunden im Postleitzahlenbereich (**PLZ**) 30999 bis 31999 aus!
- Geben Sie eine Liste aller Touren, bestehend aus Fahrer, Startzeit und Dauer, aufsteigend geordnet nach Tournummer (**TNR**) aus!
- Geben Sie für die Tour, mit der Tournummer (**TNR**) 4227 eine Liste aller SendungsIDs (**SID**), die als Einschreiben auszuliefern sind, aufsteigend sortiert nach ihrer Position in der Ladeliste aus!
- Berechnen Sie das durchschnittliche Porto aller Sendungen der Postkundin mit dem Vornamen „Pippi“ und dem Nachnamen „Langstrumpf“!
- Geben Sie eine Liste aller Touren (**TNR**, **Fahrer**) mit der Anzahl ihrer Positionen/Sendungen, geordnet nach der Dauer der Tour aus!
- Fehlsendungen: Geben Sie eine Liste aller Sendungen, jeweils mit dem Vor- und Nachnamen des zugehörigen Postkunden aus, deren Sender und Empfänger gleich sind!

## Aufgabe 2:

### 1. Datenmodellierung: ER-Modell, SQL

Gegeben sei folgender Ausschnitt aus dem ER-Modell einer Universitätsdatenbank:



- Übersetzen Sie das ER-Modell in das Relationenmodell!
- Geben Sie für die Tabellen prüft, liest und hört CREATE TABLE-Statements an, welche auch die nötigen Schlüssel- sowie Fremdschlüsselbedingungen enthalten!
- Ergänzen Sie das ER-Modell um einen schwachen Entity-Typen Übungsgruppe mit den Attributen Gruppennummer und Raum, der in Verbindung zur Tabelle Vorlesung stehen soll!

## 2. Datenbankanfragen in SQL

Gegeben seien die folgenden Tabellen:

- Student(Mat.Nr., Name, Alter, Fachbereich, Fachsemester).
- hört(Mat.Nr., Vorlesung, Semester, Note).

Erstellen Sie in SQL folgende Anfragen:

- Bestimmen Sie alle Studierenden, die älter sind als 23 Jahre!
- Bestimmen Sie das durchschnittliche Fachsemester der Studierenden des Fachbereichs "Informatik"!
- Bestimmen Sie für jeden Studierenden - gegeben durch die Matrikelnummer - und jedes Semester die Durchschnittsnote über alle gehörten Vorlesungen!
- Bestimmen Sie für jeden Studierenden - gegeben durch den Namen - und jedes Semester die Anzahl der Vorlesungen, die er in diesem Semester hört, vorausgesetzt, dass der Studierende im betrachteten Semester mindestens fünf Vorlesungen hört!
- Bestimmen Sie alle Studierenden - gegeben durch den Namen - die dieselbe Vorlesung (mindestens) in zwei verschiedenen Semestern gehört haben!

## 3. Funktionale Abhängigkeiten und Zerlegungen

Gegeben sei das Relationenschema  $R = (U, F)$  mit der Attributmenge  $U = \{A, B, C, D, E\}$  und folgender Menge  $F$  von funktionalen Abhängigkeiten:

$$F = \{A \rightarrow B, AC \rightarrow BD, BC \rightarrow A\}$$

- Bestimmen Sie die Attributhüllen  $\{A\}_F^+$  und  $\{B, C\}_F^+$ .
- Geben Sie alle Schlüssel für  $R$  an!
- Zerlegen Sie  $R$  mittels des Synthesalgorithmus in ein 3NF-Datenbankschema!
- Ist die Zerlegung von  $R$  in die beiden Relationenschemata  $R_i = (U_i, F_i), i = 1, 2$ , mit  $U_1 = \{A, C, D\}$  und  $U_2 = \{A, B, C\}$  und entsprechenden Mengen  $F_i = \Pi_{U_i}(F^+)$  von funktionalen Abhängigkeiten verlustfrei? Ist sie in BCNF?

## Themenschwerpunkt B (Betriebssysteme)

### Aufgabe 1:

#### 1. Speicherverwaltung

Gegeben sei folgendes C-Programm:

```
char feld[5000];  
int main() {  
    int i;  
    for (i=0; i<10000; i++) { feld[i] = 'x'; }  
}
```

Dieses Programm wird für einen 16-Bit-Prozessor mit MMU für Seitenadressierung übersetzt und gebunden. Die Seitengröße sei 4096 Byte. Der logische Adressraum sei so organisiert, dass die erste Seite immer ungenutzt bleibt (um Fehlzugriffe auf NULL-Zeiger zu erkennen). Für den Programmcode wird eine Seite benutzt, die ggf. auch von mehreren Prozessen gemeinsam genutzt werden kann. Die Daten liegen dahinter, der Stack wie üblich am Ende des Adressraums. Für den Stack wird die minimal mögliche Speichermenge angelegt. Die MMU unterstützt Zugriffsschutz (lesen, schreiben, ausführen) für die einzelnen Seiten, Auslagerung von Seiten, die Bildung von Freiseitenpuffern und einen Clock (Second-Chance)-Algorithmus.

- a) Beschreiben Sie den Aufbau eines Seitendeskriptors einer solchen MMU (Skizze mit Erläuterung der verschiedenen Daten und ihrem Verwendungszweck)!
- b) In einer Liste freier Kacheln (Seitenrahmen) des Hauptspeichers sind aufgeführt (jeweils die Anfangsadresse der Kachel): 0x4000, 0x7000, 0x2000, 0x9000, 0x6000, 0xa000, 0x3000.  
Obiges Programm wird nun geladen. Skizzieren Sie den Aufbau der kompletten Seiten-Kachel-Tabelle des Prozesses! (Lassen Sie in Ihrer Zeichnung um die Tabelle herum Platz für die Erweiterungen der Skizze in den Teilaufgaben c) und d)!)
- c) Wenn der Prozess in den Zustand laufend versetzt wird, muss die MMU in der Lage sein, die entsprechenden Adressumsetzungen vorzunehmen. Was muss das Betriebssystem beim Dispatching tun, damit dies funktioniert? Erweitern Sie Ihre Skizze um die hierfür erforderlichen Elemente und markieren Sie diese mit c)!
- d) Die Adresse &feld[0] sei 0x1008. Beschreiben Sie, wie die Adressumsetzung bei der Anweisung feld[0] = 'x'; erfolgt, so dass der Wert 'x' an eine Stelle im Hauptspeicher des Rechners geschrieben wird! Ergänzen Sie auch Ihre Skizze entsprechend und markieren Sie die Elemente mit d)!
- e) In welcher Seite liegt die Variable i?
- f) Die for-Schleife überschreitet das Ende des Feldes feld. Was hat dies für Folgen? Bei welchem Schleifendurchlauf treten diese Folgen auf?

## 2. Dateisystem

- a) Hinter der hierarchischen Sicht auf ein Dateisystem, wie es typischerweise unter UNIX oder WindowsNT eingesetzt wird, verbirgt sich eigentlich ein einfach strukturierter flacher Namensraum (die "eentlichen" Namen sind die Inode-Nummern unter UNIX bzw. die Indizes in die Master-File-Table bei WindowsNT).  
Beschreiben Sie für ein solches Dateisystem Ihrer Wahl, wie das Betriebssystem es organisiert, dass der Inhalt der Datei /home/user1/datei1 bzw. \home\user1\datei1 (Dateigröße sei ca. 1 MByte) aufgefunden werden kann!  
Skizzieren Sie hierzu alle Datenstrukturen, die hierfür auf dem Hintergrundspeicher angelegt sind, und beschreiben Sie für jeden Typ einer solchen Datenstruktur, wofür sie konzeptionell vorgesehen ist!
- b) Zu jeder Datei verwaltet das Betriebssystem eine Menge von Attributen. Wo werden diese Attribute verwaltet?
- c) Beispiele für solche Dateiattribute sind:  
Dateityp,  
Zugriffsrechte,  
Dateigröße,  
Zeit der letzten Änderung,  
Zeit des letzten Zugriffs,  
Link-Zähler und  
Eigentümer.  
Beschreiben Sie für jedes dieser Attribute jeweils möglichst ausführlich, wofür das Betriebssystem das Attribut verwaltet (was sagt das Attribut im Detail aus, in welchen Situationen ändert es sich ggf., auf welche Operationen hat es Auswirkungen etc.)!
- d) UNIX- und auch neuere Versionen von WindowsNT-Dateisystemen unterscheiden zwischen "Hard-Links" und "Symbolic-Links". Was ist der Unterschied und warum benötigt man diese unterschiedlichen Arten von Links?

## 3. Prozesse

- a) Nach welchen Kriterien kann man Auswahlstrategien zur Prozessorvergabe (Scheduling) klassifizieren?
- b) Welche Strategie würden Sie für das Scheduling in einem Betriebssystem einsetzen, das für die Überwachung und Steuerung von Abläufen in einem Kernkraftwerk eingesetzt ist? Welche Parameter sind hierbei für die Schedulingstrategie primär entscheidend und wie werden sie in der von Ihnen gewählten Strategie umgesetzt?
- c) Welche Strategie wird heutzutage üblicherweise in Mehrbenutzer-/Mehrprogramm-Betriebssystemen wie z. B. UNIX eingesetzt?  
Welche Ziele werden mit solch einer Strategie verfolgt?  
Bei solch einer Strategie spielen mehrere Parameter eine Rolle - teilweise haben diese Parameter sehr gegensätzliche Wirkungen. Beschreiben Sie die Parameter und ihre Wirkung auf das Systemverhalten!



#### 4. Threads und Koordinierung

Zwei Threads implementieren ein klassisches Erzeuger-Verbraucher-System:

Thread T1 liest von einem Eingabekanal (blockierendes Lesen, falls dort gerade keine Daten anstehen) und schreibt die Daten in einen Ringpuffer.

Thread T2 entnimmt die Daten aus dem Ringpuffer und gibt sie auf einem Ausgabekanal aus.

- Wählen Sie zur Synchronisation der beiden Threads ein geeignetes Verfahren und begründen Sie diese Wahl kurz!
- Skizzieren Sie in einer Programmiersprache Ihrer Wahl die Funktion `producer()`, die die Aufgabe von Thread T1 realisiert, sowie die Funktion `consumer()`, die die Aufgabe von Thread T2 realisiert!
- Warum funktioniert die beschriebene Aufgabe mit User-Level-Threads nicht einwandfrei, sondern nur mit Kernel-Level-Threads? Beschreiben Sie hierzu kurz den grundsätzlichen Unterschied zwischen beiden Kategorien und arbeiten Sie dabei heraus, welcher Umstand das korrekte Verhalten des beschriebenen Erzeuger-Verbraucher-Szenarios verhindert!

#### Aufgabe 2:

##### Teilaufgabe 1

Die folgenden drei Jobs stehen zu den angegebenen Zeiten (in Minuten) zur Ausführung an:

Job	Ankunftszeit	Bearbeitungsdauer
1	0	20
2	4	10
3	10	6

- Stellen Sie die zeitlichen Abläufe in Gantt-Diagrammen (Balkendiagramme über der Zeitachse) dar und berechnen Sie dazu den Turnaround eines jeden Jobs, sowie den Average Turnaround bei den folgenden Scheduling-Strategien:
  - First-Come-First-Served (FCFS)
  - Shortest-Job-First (SJF)
  - Preemptive-Shortest-Job-First (Preemptive SJF)
  - Round-Robin (RR) mit Zeitquantum  $q = 8$
- Zusätzlich zu den obigen Jobs 1 bis 3 (von der Kategorie Batch) sollen zu den Zeitpunkten 6 und 12 die Jobs Nummer 4 und 5 (von der Kategorie Interactive) mit jeweils einer Laufzeit von 10 Zeiteinheiten zur Bearbeitung anstehen. Nun soll eine Multi-Level-Scheduling-Strategie verfolgt werden derart, dass jeweils eine Warteschlange für die Kategorien Interactive und Batch existiert, wobei das Scheduling bei interaktiven Jobs mit dem Round-Robin-Verfahren mit Zeitquantum  $q = 4$  und bei den Batch-Jobs nach FCFS abläuft. Die Warteschlange der interaktiven Jobs hat Priorität über die Warteschlange der Batch-Jobs, d. h. falls gerade ein Batch-Job bearbeitet wird und es kommt ein neuer interaktiver Job an, so wird der Batch-Job solange angehalten, bis keine interaktiven Jobs mehr im System sind. Geben Sie die Wartezeit eines jeden Jobs an!

**Teilaufgabe 2**

Sie sind gerade an einen neuen Wohnort gezogen und wollen dies nun dort bei der zuständigen Meldebehörde anzeigen. Bei dieser Meldebehörde gibt es einen Warteraum, der maximal 20 Personen fasst. Sobald Sie in den Warteraum eintreten können, ziehen Sie aus einem Automaten eine fortlaufend vergebene Nummer, mit der Sie später von einem der drei Schalter aus aufgerufen werden. Ein Schalterbeamter wählt jeweils aus den (maximal 20) wartenden Personen diejenige mit der niedrigsten Nummer aus und bearbeitet diese Anmeldung vollständig.

Implementieren Sie Ihr obiges Vorgehen zur Anmeldung und die Arbeit eines Schalterbeamten in Pseudo-Code! Die Synchronisation der beteiligten Prozesse soll ausschließlich über allgemeine Semaphore (Counting Semaphore) geschehen. Hinweis: Die obige Aufgabenstellung beinhaltet im Wesentlichen ein Erzeuger-Verbraucher-Schema mit beschränktem Puffer.

**Teilaufgabe 3**

In der Freispeicherliste zu einem Arbeitsspeicher existieren die freien Blöcke in der angegebenen Reihenfolge mit den folgenden Größen:

10KB, 4KB, 20KB, 9KB, 12KB

Geben Sie für jeden Vergabeschritt die verbleibenden freien Speicherblöcke für die folgenden Anfragen an (ebenfalls in dieser Reihenfolge) nach Blöcken der Größe:

12KB, 10KB, 9KB

Verwenden Sie dabei die folgenden Speichervergabe-Verfahren:

- a) First-Fit
- b) Best-Fit
- c) Worst-Fit