

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

**Kennzahl:** \_\_\_\_\_

**Kennwort:** \_\_\_\_\_

**Arbeitsplatz-Nr.:** \_\_\_\_\_

**Herbst  
2012**

**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 Auswahlregel zu bearbeiten sind!**

**Anzahl der Druckseiten dieser Vorlage: 16**

---

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. A2, B1)!

**Bitte wenden!**

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

**Teilaufgabe 1:****Aufgabe 1: Datenmodellierung: ER-Modell, SQL**

In einer Flugdatenbank sollen folgende Informationen modelliert werden:

- Eine Fluggesellschaft hat einen Namen und mehrere Standorte.
- Ein Flughafen hat einen eindeutigen Identifikator (ID), und es werden die Stadt und das Land angegeben, in denen der Flughafen liegt.
- Ein Flug verbindet zwei Flughäfen; er wird durch eine Flugnummer und ein Startdatum identifiziert. Zusätzlich werden die Startzeit, das Landedatum, die Landezeit, die Flugstrecke und die Anzahlen der insgesamt verfügbaren sowie der bereits reservierten Plätze gespeichert.
- Zur Unterstützung von Datenbankabfragen über die Zeit soll es eine Tabelle geben, die jedem Datum den Tag, den Monat, das Jahr und die Jahreszeit (Frühjahr, Sommer, Herbst, Winter) als separate Attribute zuordnet. Eine weitere Tabelle soll jeder Flugnummer eine Fluggesellschaft zuordnen.
  - a) Modellieren Sie das oben dargestellte Szenario möglichst vollständig in einem ER-Modell! Verwenden Sie wann immer möglich (binäre oder auch höherstellige) Relationships!
  - b) Übertragen Sie Ihr ER-Modell ins relationale Datenmodell. Erstellen Sie dazu Tabellen mit Hilfe von CREATE TABLE-Statements in SQL. Berücksichtigen Sie die Fremdschlüsselbeziehungen!
  - c) Es soll die Integritätsbedingung eingehalten werden, dass die Anzahl der bereits reservierten Plätze eines Fluges höchstens so groß ist wie die Anzahl der insgesamt verfügbaren Plätze. Schreiben Sie ein SELECT-Statement, das diese Integritätsbedingung überprüft, indem es die verletzenden Flüge ausgibt.
  - d) Geben Sie geeignete INSERT-Statements an, die in alle beteiligten Tabellen jeweils mindestens ein Tupel einfügen, so dass alle Integritätsbedingungen erfüllt sind, nachdem alle Einfügungen ausgeführt wurden!

**Fortsetzung nächste Seite!**

**Aufgabe 2: Datenbankanfragen und -änderungen in SQL**

Formulieren Sie in SQL die folgenden Anfragen, Views bzw. Update-Statements an die Flugdatenbank aus Aufgabe 1:

- a) Welche Flüge der Fluggesellschaft Lufthansa (LH) starten in Frankfurt am Main, Deutschland. Ausgegeben werden sollen Flugnummer und Startdatum!
- b) Wie viele Flüge gibt es, gruppiert nach Startdatum, von Deutschland nach Frankreich?
- c) Bestimmen Sie alle Verbindungen von München, Deutschland, nach Australien mit einmaligem Umsteigen. Der Anschlussflug soll an demselben Tag erfolgen, an dem der erste Flug beendet wird. Für Flug und Anschlussflug sollen jeweils Flugnummer und Startdatum ausgegeben werden. Bestimmen Sie dabei auch die Gesamtflugstrecke!
- d) Bestimmen Sie die Anzahl der Reservierungen für die Flüge von Frankfurt am Main, Deutschland, nach London, Großbritannien, am 17.01.2012!
- e) Erstellen Sie eine View, die die Anzahl der ausgebuchten Flüge gruppiert nach Jahr und Jahreszeit angibt!
- f) Buchen Sie zwei Plätze des Fluges mit der Flugnummer DL007 am 18.01.2012 auf den Flug mit der Flugnummer LH007 am 19.01.2012 um - nehmen Sie an, dass es noch genügend viele freie Plätze gibt.

**Aufgabe 3: Funktionale Abhängigkeiten und Zerlegungen**

Gegeben sei das Relationenschema  $R = (U, F)$  mit der Attributmenge

$$U = \{A, B, C, D, E\}$$

und der folgenden Menge  $F$  von funktionalen Abhängigkeiten:

$$F = \{B \rightarrow C, C \rightarrow BD, ACD \rightarrow BE\}$$

- a) Geben Sie alle Schlüssel für das Relationenschema  $R$  (jeweils mit Begründung) sowie die Nichtschlüsselattribute an!
- b) Ist  $R$  in 3NF bzw. in BCNF? Geben Sie jeweils eine Begründung an.
- c) Geben Sie eine Basis  $G$  von  $F$  an! Zerlegen Sie  $R$  mittels des Synthesalgorithmus in ein 3NF-Datenbankschema.
- d) Zerlegen Sie  $R$  bezüglich der funktionalen Abhängigkeit  $C \rightarrow BD$  verlustfrei in zwei Relationenschemata  $R_i = (U_i, F_i)$ ,  $i = 1, 2$ ! Ist die Zerlegung unabhängig (mit Begründung)?

**Teilaufgabe 2:****Aufgabe 1: Allgemeinwissen**

1.1 Bewerten Sie die folgenden Aussagen.

Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind! Begründen Sie ihre Antworten kurz!

- a) B-Bäume verlieren ihre Vorteile gegenüber binären Bäumen in Hauptspeicherdatenbanken.
- b) Eine Relation kann mehrere Primärschlüsselattribute besitzen.
- c) Heutzutage werden zumeist objektorientierte Datenbanksysteme eingesetzt.
- d) Fremdschlüsselwerte dürfen nicht NULL sein.
- e) Alle Metadaten werden in Tabellen gespeichert.

1.2 Nennen Sie kurz die vier wesentlichen Transaktionen!

**Aufgabe 2: Normalformen**

2.1 Die Normalisierung von Relationenschemata dient der Vermeidung von Redundanzen und dadurch bedingter Anomalien. Diskutieren Sie anhand des folgenden Beispiels die Einfüge-, Änderungs- und Löschanomalien.

AbtNr	AbtName	AbtOrt	MitarbeiterNr	MitarbeiterName
1	Entwicklung	München	1	Schmidt, H.
2	Forschung	München	2	Müller, G.
1	Entwicklung	München	3	Kohl, H.
3	Fertigung	Nürnberg	4	Weizenbaum, L.
4	Logistik	Passau	5	Nienhof, R.
6	Vertrieb	Würzburg	6	Baier, V.

2.2 In welcher Normalform ist das Beispiel aus Aufgabenteil 2.1? Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind! Welche Bedingung der nächsthöheren Normalform ist verletzt? Berücksichtigen Sie, dass eine Relation nur dann in n-ter Normalform ist, wenn sie die Bedingungen aller m-ten Normalformen mit  $m \leq n$  erfüllt!

**Fortsetzung nächste Seite!**

## Aufgabe 3: ER-Modellierung und Relationenmodell

3.1 Nennen und erläutern Sie kurz vier Beispiele für Integritätsbedingungen!

3.2 Erstellen Sie ein ER-Diagramm bestehend aus Entitäts- und Beziehungstypen sowie Attributen! Geben Sie auch die Kardinalitäten mit an!

Verwenden Sie bei Entitäten und Attributen ausschließlich die beschriebenen, soweit dies möglich ist. Verwenden Sie keine künstlichen Schlüssel, es sei denn, diese sind der Aufgabenstellung eindeutig zu entnehmen. Die Kardinalitätseinschränkungen können Sie entweder in der (min,max)-Notation oder der einfachen Notation nach Chen (1:1, 1:N, M:N) angeben.

Als Szenario dient ein **Auktionshaus**, das folgendermaßen beschrieben ist:

### **Hinweis:**

In der Formulierung der Beziehungen sind die Kardinalitätseinschränkungen genau zu beachten. Im Allgemeinen gilt: alles was nicht explizit eingeschränkt ist, muss im Modell auch uneingeschränkt bleiben (keine Einschränkungen in eine Beschreibung hineininterpretieren). Alles was explizit eingeschränkt ist, darf im Modell nicht uneingeschränkt bleiben (alle formulierten Einschränkungen sind zu erkennen).

### **Szenario:**

1. Im Zentrum des Systems stehen die Benutzer. Sie werden durch eine ID identifiziert und haben einen Benutzernamen, sowie eine Adresse, die sich aus Straße, PLZ und Ort zusammensetzt.
2. Benutzer initiieren Auktionen. Jede Auktion ist genau einem Benutzer zugeordnet und wird ebenfalls über eine ID identifiziert. Daneben hat sie einen Namen, sowie Beginn- und Endezeitpunkt.
3. Benutzer geben Gebote für Auktionen ab. Zu jedem Gebot werden Betrag und Zeitpunkt des Gebotes gespeichert. Natürlich soll ein Benutzer auch mehrfach auf eine Auktion bieten können.
4. Zu guter Letzt existiert ein Bewertungssystem. Eine Bewertung wird von einem Benutzer zu einer bestimmten Auktion abgegeben. Zu ihr sollen die Bewertung selbst (ein numerischer Wert) und der Zeitpunkt der Bewertung gespeichert werden. Pro Benutzer und Auktion kann nur eine Bewertung abgegeben werden.

3.3 Erstellen Sie zu dem ER-Diagramm aus Aufgabenteil 3.2 ein Relationenschema!

Seien Sie dabei so vollständig wie möglich – bspw. Berücksichtigung totaler Partizipationen. Vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen!

### **Beispiel für die Notation:**

```
Relationenname (Primärschlüssel, Attribut1, Attribut2, ...,  
    Fremdschlüssel[AndereRelation],  
    (FKAttrA, FKAttrB)[DritteRelation])  
  
Attribut2 NOT NULL
```

**Fortsetzung nächste Seite!**

## Aufgabe 4: Schlüsselzugriffe

- 4.1 Gegeben sei ein anfangs leerer B-Baum (maximale Anzahl der Einträge pro Knoten ist  $2k$ ,  $k=1$ ). Die Zahlen 21, 25, 29, 19, 13, 23 und 20 sind in dieser vorgegebenen Reihenfolge in den B-Baum einzufügen. Löschen Sie dann den Schlüssel 23.

Zeichnen Sie die Zustände des B-Baums nach jeder Strukturänderung! Falls sich durch eine Operation die Struktur des Baumes nicht ändert, muss er nicht neu gezeichnet werden, sondern der einzufügende Wert kann einfach in den Baum eingetragen werden.

Wenn eine Operation mehrere Strukturänderungen erfordert, so ist der Baum nach jeder Änderung neu zu zeichnen.

- 4.2 Nennen Sie zwei Unterschiede zwischen B-Baum und B\*-Baum sowie je einen Vor- und Nachteil beider Varianten!

## Aufgabe 5: Anfrageverarbeitung

- 5.1 Zeichnen Sie für folgendes SQL-Statement einen nicht-optimierten Anfragegraphen.

```
SELECT a1.name AS aname, a1.price,
       artgroup.name AS gname,
       MAX (a2.price) AS max_price
FROM article a1, article a2, artgroup
WHERE a1.group_id = artgroup.id
      AND a2.group_id = a1.group_id
      AND artgroup.name LIKE 'Kabel %'
GROUP BY a1.name, a1.price, artgroup.name
HAVING MAX(a2.price) > 2*a1.price
```

- 5.2 Wie kann die relationale Algebra zur Optimierung von SQL-Anfragen eingesetzt werden? Geben Sie zwei Beispiele für heuristische Optimierungsregeln an.

## Themenschwerpunkt B (Betriebssysteme)

### Teilaufgabe 1:

### Aufgabe 1: Synchronisation mit Sempaphoren

1. Gegeben sei folgende informelle Definition der Operation P auf Semaphore s durch Pseudocode. Der Semaphore sei symbolisch durch int s dargestellt. Es wird als gegeben angenommen, dass die Operation atomar ist und wechselseitig ausgeschlossen ausgeführt wird.

```
public void P (int s) {  
    s = s - 1;  
    if (s < 0) { Prozess in die Menge der bezüglich s  
                wartenden Prozesse einreihen }  
}
```

2. Geben Sie eine analoge Definition der Operation V an.

Gegeben sei folgende Implementierung der Synchronisation des Erzeuger-Verbraucher-Problems in Pseudocode. Der Zugriff auf einen Puffer W erfolge durch einen Semaphore wa, der mit 1 initialisiert sei. Zusätzlich gibt es einen Semaphore voll (initialisiert mit 0), der die Elemente im Puffer zählt, sowie einen Semaphore leer (initialisiert mit der Anzahl der freien Pufferplätze), der die Anzahl der freien Pufferplätze zählt.

Erzeuger E	Verbraucher V
<pre>{   while (TRUE)   {     &lt;produziere&gt;;     P(leer);     P(wa);     &lt;schreibe nach W&gt;;     V(wa);     V(voll);   } }</pre>	<pre>{   while (TRUE)   {     P(wa);     P(voll);     &lt;entnimm aus W, falls Element da; sonst warte&gt;;     V(wa);     V(leer);     &lt;verarbeite&gt;;   } }</pre>

Was ist das Problem bei dieser Implementierung? Begründen Sie Ihre Antwort kurz.

**Fortsetzung nächste Seite!**

3. Ein Möbelhändler hat eine Lagerhalle für Tische und Stühle. Beliefert wird er von einem Fabrikanten F, der bei jeder Lieferung einen Tisch und zwei Stühle bringt. Ein Ausfahrer T bringt je Fahrt einen Tisch zum Kunden, ein Ausfahrer S je Fahrt einen Stuhl. Zum Be- und Entladen muss eine Laderampe benutzt werden. Für sich alleine betrachtet laufen die beteiligten Prozesse folgendermaßen ab:

Process T	Process S	Process F
{	{	{
while (TRUE)	while (TRUE)	while (TRUE)
{	{	{
<zur Rampe fahren>;	<zur Rampe fahren>;	<zur Rampe fahren>;
<1 Tisch aufladen>;	<1 Stuhl aufladen>;	<1 Tisch entladen>;
<Rampe verlassen>;	<Rampe verlassen>;	<2 Stühle entladen>;
}	}	<Rampe verlassen>;
}	}	}
		}

Synchronisieren Sie diese drei Prozesse, indem Sie in Pseudocode-Notation in geeigneter Weise Semaphore deklarieren und Semaphor-Operationen einfügen. Die Pseudocode-Operation zum Deklarieren eines Semaphors mit Namen *name* mit Startwert *n* laute:

name(*n*);

Die Pseudocode-Operationen zum Aufrufen von P und V auf der Semaphore mit Namen *name* lauten:

P(*name*)

V(*name*)

Sperrphasen sind möglichst kurz zu halten und folgende Bedingungen müssen erfüllt sein:

- Zur Laderampe kann jeweils nur ein Prozess fahren.
- Der Abholer T darf nur zur Rampe fahren, wenn noch ein Tisch im Lager ist.
- Der Abholer S darf nur zur Rampe fahren, wenn noch ein Stuhl im Lager ist.
- Die Lagerhalle hat beschränkte Kapazität, sie kann höchstens 12 Möbelstücke (Tische oder Stühle) aufnehmen.
- Der Fabrikant darf nur zur Rampe fahren, wenn er seine Lieferung vollständig abladen kann.
- Zu Beginn sei das Lager leer und die Rampe frei.

a) Listen Sie alle benötigten Semaphore auf und erklären Sie ihren Zweck bzw. Ihre Pragmatik!

b) Fügen Sie in dem oben aufgelisteten Pseudocode des Ablaufs der beteiligten Prozesse an geeigneter Stelle entsprechende Aufrufe zum Deklarieren der Semaphore und der Aufrufe von P und V ein! Übertragen Sie den Pseudocode dazu auf Ihr Arbeitsblatt.

**Fortsetzung nächste Seite!**



## Aufgabe 2: Scheduling

1. Was versteht man allgemein unter **preemptive scheduling** sowie **nonpreemptive scheduling**? Erläutern Sie kurz den Unterschied zwischen beiden Strategien.
2. Gegeben seien fünf Prozesse mit ihren jeweiligen Ankunftszeiten  $a$  und Bedienzeiten  $b$ :

Prozess	Ankunftszeit $a$	Bedienzeit $b$
P1	0	8
P2	6	7
P3	3	5
P4	8	3
P5	5	3

Übertragen Sie die nachfolgenden Gantt-Diagramme und tragen Sie für die angegebenen Scheduling-Verfahren ein, welcher Prozess zu welcher Zeit die CPU erhält. Markieren Sie rechenwillige Prozesse mit "-" und rechnende Prozesse mit "x" in der jeweiligen Spalte.

- (a) **First Come First Served (FCFS)**: FCFS bedeutet, dass die Prozesse in der Reihenfolge ihrer Ankunft ausgeführt werden. Prozessunterbrechung ist nicht möglich.

	0	5	10	15	20	25	Zeit
P1							
P2							
P3							
P4							
P5							

- (b) **Shortest Remaining Processing Time (SRPT)**: SRPT bedeutet, dass jeweils der Prozess mit der kürzesten Restbedienzeit ausgeführt wird. Prozessunterbrechung ist nur bei Ankunftszeitpunkten neuer Prozesse möglich. Haben zwei Prozesse gleiche Restbedienzeiten, soll in unserem System dem Prozess Vorrang gegeben werden, der *zuletzt* im System angekommen war.

	0	5	10	15	20	25	Zeit
P1							
P2							
P3							
P4							
P5							

Fortsetzung nächste Seite!

3. Berechnen Sie die **mittlere Verweilzeit**  $\bar{V}$  und die **mittlere Wartezeit**  $\bar{W}$  für die beiden Strategien der vorangehenden Teilaufgabe. Nutzen Sie zur Berechnung folgende Formeln:

- $c_i$  bezeichnet die **Abgangszeit** (der Zeitpunkt seiner fertigen Bearbeitung) des Prozesses  $P_i$ ,  $a_i$  seine Ankunftszeit und  $b_i$  seine Bedienzeit;
- $v_i = c_i - a_i$  bezeichnet die **Verweilzeit** des Prozesses  $P_i$ ;
- $w_i = v_i - b_i = c_i - a_i - b_i$  bezeichnet die **Wartezeit** des Prozesses  $P_i$ ;
- Gibt es  $n \in \mathbb{N}$  Prozesse, so gilt für die **mittlere Verweilzeit**:

$$\bar{V} = \frac{1}{n} \sum_{i=1}^n v_i$$

- Für die **mittlere Wartezeit** gilt:

$$\bar{W} = \frac{1}{n} \sum_{i=1}^n w_i$$

4. Warum ist bei interaktiven Systemen weder FCFS noch SRPT besonders gut als Strategie geeignet?

**Fortsetzung nächste Seite!**

### Aufgabe 3: Seitenersetzung

Bei der Ausführung eines Speicherzugriffs bei der virtuellen Speicherverwaltung kann es vorkommen, dass sich die referenzierte Seite nicht im Arbeitsspeicher befindet. Diese Situation wird Seitenfehler (engl. Page fault) genannt.

Die Behandlung eines Seitenfehlers erfordert Maßnahmen zur Ersetzung einer Seite im Arbeitsspeicher, um die gewünschte Seite in eine Kachel (frame) des Arbeitsspeichers einlagern zu können, muss zunächst eine andere Seite vom Arbeitsspeicher auf den Hintergrundspeicher ausgelagert werden oder eine unveränderte Seite kann verworfen werden. Bei der Auswahl der auszulagernden Seite können verschiedene Seitenersetzungs-Strategien angewendet werden.

1. Erklären Sie die Seitenersetzungs-Strategien **LIFO** und **LFU** jeweils knapp.
2. Sei eine Menge von Seiten  $N = \{0, 1, 2, 3, 4, 5\}$  und eine Menge von Kacheln (frames)  $F = \{f_1, f_2, f_3, f_4\}$  gegeben. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

$$w = 1 \ 3 \ 5 \ 4 \ 4 \ 2 \ 3 \ 2 \ 1 \ 0 \ 5 \ 3 \ 5 \ 0 \ 4 \ 3$$

Vollziehen Sie die Strategie **LRU (Least Recently Used)** schrittweise nach, indem Sie die folgende Tabelle auf Ihr Arbeitsblatt übertragen und vervollständigen! Notieren Sie in der betreffenden Spalte, ob ein Seitenfehler auftrat. Sie können optional die Spalte "Notizen" geeignet verwenden, wenn Sie möchten.

Anfrage	$f_1$	$f_2$	$f_3$	$f_4$	Seitenfehler	Notizen
1	1	-	-	-	ja	
3	1	3	-	-	ja	
5	1	3	5	-	ja	
4	1	3	5	4	ja	
4	1	3	5	4	nein	
2						
3						
2						
1						
0						
5						
3						
5						
0						
4						
3						

3. Wenn man die Menge der Kacheln vergrößert, sinkt dann zwangsläufig bei allen Seitenersetzungs-Strategien die Zahl der Seitenfehler? (Ohne Begründung)

**Teilaufgabe 2:****Aufgabe 1:**

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

Job	Ankunftszeit	Bearbeitungsdauer
T1	0	20
T2	4	10
T3	10	6

- (a) 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:

- (1) First-Come-First-Served (FCFS)
- (2) Shortest-Job-First (SJF)
- (3) Preemptive-Shortest-Job-First (Preemptive SJF)
- (4) Round-Robin (RR) mit Zeitquantum  $q = 8$

- (b) Zusätzlich zu den obigen Jobs T1, T2, T3 (der Kategorie Batch) sollen zu den Zeitpunkten 6 und 12 die Jobs T4 und T5 (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.

**Fortsetzung nächste Seite!**

**Aufgabe 2:**

Ein Rechnersystem mit vier Betriebsmitteltypen  $R_1, R_2, R_3, R_4$  und fünf Prozessen  $P_1, P_2, P_3, P_4, P_5$  befinde sich im unten gezeigten Zustand.

		$R_1$	$R_2$	$R_3$	$R_4$
Verfügbarkeitsvektor		0	2	2	5
Belegte Betriebsmittel	$P_1$	2	3	0	6
	$P_2$	0	0	2	0
	$P_3$	4	5	2	3
	$P_4$	2	1	0	0
	$P_5$	4	1	0	0
Maximale Anforderungen inkl. bereits belegter Betriebsmittel	$P_1$	2	5	0	6
	$P_2$	0	5	2	7
	$P_3$	6	5	4	3
	$P_4$	2	1	0	0
	$P_5$	6	5	0	6

- (a) Erklären Sie informell die Vorgehensweise des Banker's-Algorithmus'!
- (b) Prüfen Sie zunächst, ob sich das obige System in einem sicheren Zustand befindet!
- (c) Prüfen Sie mit Hilfe des Banker's-Algorithmus', ob eine neu hinzukommende Ressourcen-Anforderung (0 2 0 4) des Prozesses  $P_4$  erfüllt werden sollte!

**Fortsetzung nächste Seite!**

**Aufgabe 3:**

Bei einer Auktion mit einem Auktionator und 50 Interessenten werde der erzielbare Verkaufspreis des zu versteigernden Kunstwerks vom Auktionator folgendermaßen ermittelt:

Der Einstiegspreis von 1000,-- EUR wurde vom Auktionator festgesetzt. Zu diesem Preis sind auch garantiert alle Interessenten zahlungswillig.

In jeder Runde antwortet nun jeder der 50 Interessenten genau einmal, ob er bereit ist, den aktuellen Preis zu bezahlen. Die Antwortreihenfolge der Interessenten ist dabei beliebig.

Erst wenn sich alle 50 Interessenten erklärt haben, wird vom Auktionator geprüft, ob es mehr als einen zahlungswilligen Interessenten gibt.

- Ist dies der Fall, so setzt der Auktionator einen um 100,-- EUR höheren Preis für die nächste Runde fest.
- Findet sich für einen aktuellen Preis kein Zahlungswilliger mehr, so ist die Auktion beendet und der Auktionator kassiert den vorherigen, also den noch um 100,-- EUR niedrigeren Preis von einem beliebigen der zuletzt zahlungswilligen Interessenten.

Übertragen und ergänzen Sie die nachfolgenden Pseudocode-Fragmente für den Auktionator und die Interessenten-Prozesse zu einer korrekten Lösung bezüglich der oben formulierten Vorgaben.

Als Semaphor-Operationen sind `wait(.)` und `signal(.)` zu verwenden.

Kommentieren Sie dabei Ihre jeweiligen Verwendungen der `wait(.)`- und `signal(.)`-Operationen.

**Fortsetzung nächste Seite!**

*Globale Initialisierungen:*

```
/* wechselseitiger Ausschluss von Auktionator und der Gruppe aller Interessenten */  
Semaphore mutexPreisRunde = new Semaphore(1);
```

```
/* zur Synchronisation der Interessenten */  
Semaphore mutexInteressenten = new Semaphore(0);
```

```
int preis = 1000;  
int antworten = 0;  
int zahlungswillige = 50;
```

*Auktionator:*

```
while( Auktion nicht beendet ) {  
    .....  
    .....  
  
    if (zahlungswillige > 1) {  
        .....  
        .....  
        .....  
        .....  
        .....  
        .....  
    }  
    else {  
        Auktion beenden;  
        Preis kassieren;  
        Kunstwerk übergeben;  
    }  
}
```

*Interessent:*

```
while( Auktion nicht beendet ) {  
    .....  
    .....  
    .....  
    .....  
  
    if (neuer Preis noch ok) {  
        .....  
        .....  
        .....  
    }  
  
    .....  
    .....  
    .....  
    .....  
    .....  
    .....  
}
```

**Fortsetzung nächste Seite!**

**Aufgabe 4:**

- (a) Beschreiben Sie knapp die drei Speicherverwaltungsstrategien **First-Fit**, **Best-Fit** und **Worst-Fit**. Nennen Sie zu jeder Strategie mindestens einen Vorteil und einen Nachteil!

- (b) Gegeben seien die folgenden freien Speicherblöcke in der Freispeicher-Liste in genau dieser Reihenfolge, wobei 1MB = 1024KB. Die Blöcke liegen jeweils nicht direkt nebeneinander im RAM.

1000KB	1MB	200KB	110KB	4MB
--------	-----	-------	-------	-----

Es werden nun an das Betriebssystem in folgender Reihenfolge Speicheranforderungen gestellt:

1000KB	900KB	200KB	100KB	110KB	2MB
--------	-------	-------	-------	-------	-----

Wie teilt jede der Strategien **First-Fit**, **Best-Fit** und **Worst-Fit** den Speicher zu? Zur vollständigen Lösung gehört die Liste aller Frei-Blöcke nach jeder erfüllten Anforderung.

- (c) Welches Verfahren schneidet für das obige Beispiel am schlechtesten ab bzgl. Erfüllung der Anfragen? Begründen Sie Ihre Antwort!
- (d) Welches Verfahren arbeitet in obigem Beispiel am effizientesten bzgl. unerwünschter Fragmentierung des freien Speichers? Begründen Sie Ihre Antwort!