

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

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2011**

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: **23**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 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:

1. ER-Modellierung

Erstellen Sie ein ER-Modell für die Datenbank einer Firmenkantine:

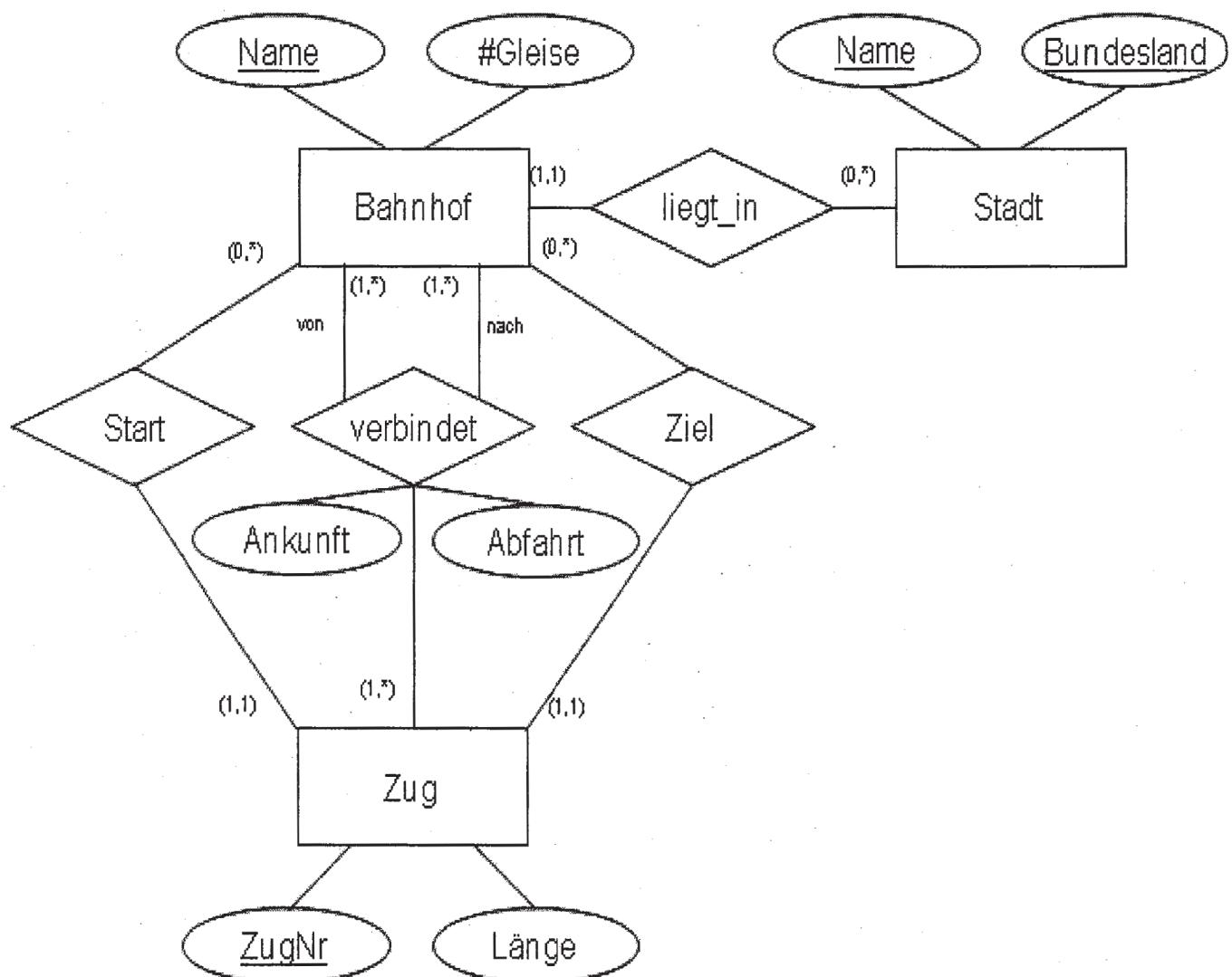
- Die Kantine hat eine reichhaltige Auswahl an Gerichten. Jedes Gericht hat eine Nummer, einen Namen und einen Preis. Gerichte, die sich in ihrer Zusammenstellung nur leicht (z. B. aufgrund einer Beilage) unterscheiden, können den gleichen Namen haben.
- Die Kantine bietet täglich aus ihrer Auswahl mehrere Gerichte an. Sie möchte Informationen darüber haben, welches Gericht an welchem Tag wie oft verkauft wurde.
- Jedes Gericht besteht aus mehreren Bestandteilen: genau einem Hauptbestandteil (z.B. Fleisch) und mehreren Beilagen (z.B. Suppe, Salat, Nachspeise). Es ist wichtig zu erkennen, ob es sich bei einem Bestandteil um einen Hauptbestandteil oder um eine Beilage handelt. Die Beilagen sind fest und können nicht ausgewählt werden. Die Zusammensetzung eines Gerichts ist nur für die Vorbereitung von Bedeutung, da jeder Bestandteil einzeln zubereitet wird. Jeder Bestandteil kann in verschiedenen Gerichten verwendet werden, wobei Redundanzen vermieden werden sollen.
- Für jeden Bestandteil eines Gerichts ist ein Rezept vorhanden.
- Ein Bestandteil besteht aus mehreren Zutaten (z.B. Zucker, Mehl, ...). Jede Zutat hat einen Namen und eine Kalorienanzahl (pro 100g).
- Zu guter Letzt soll die Menge einer Zutat, die in einem Bestandteil eines Gerichts verwendet wird, in der Datenbank erfasst werden.

Schlüsselattribute werden durch Unterstrichen gekennzeichnet. Kardinalitäten von Beziehungen sollen in (min, max)-Notation in das Diagramm mit aufgenommen werden.

Fortsetzung nächste Seite!

2. Relationenmodell

Gegeben sei im Folgenden eine ER-Modellierung von Zugverbindungen. Übertragen Sie das ER-Modell in ein relationales Schema und kennzeichnen Sie durch Unterstreichen die gewählten Primärschlüssel. Verfeinern Sie schließlich das relationale Schema soweit wie möglich durch Eliminierung von Relationen.



Fortsetzung nächste Seite!

3. Relationale Anfragen

Gegeben sei die folgende relationale Datenbank des Online-Blumenhändlers „Flowers4you“:

Das Warenangebot an Blumenbouquets findet sich in der Tabelle „Angebot“. Jedes Produkt hat eine Produktnummer (PNr), einen Produktnamen (PName), eine Produktinformation (PInfo) und einen Preis (Preis) in Euro:

Angebot			
<i>PNr</i>	<i>PName</i>	<i>PInfo</i>	<i>Preis</i>
1	Wolkenlos	12 Mini-Sonnenblumen	19,99
2	Sonnenstrahl	20 gelbe Rosen	25,99
3	Romantico	12 rote Rosen	19,99
4	Leuchtfieber	5 weiße Rosen, 5 rote Rosen, 5 gelbe Rosen	22,99
...

Die Kundendaten Kundennummer (KNr), Name (Name), Adresse (Adr), Stadt (Stadt), Postleitzahl (PLZ), Land (Land), Kreditkarte (KK), Kreditkartennummer (KKNr) und das Ablaufdatum der Kreditkarte (AD) werden in der Tabelle „Kunde“ gespeichert:

Kunde								
<i>KNr</i>	<i>Name</i>	<i>Adr</i>	<i>Stadt</i>	<i>PLZ</i>	<i>Land</i>	<i>KK</i>	<i>KKNr</i>	<i>AD</i>
1001	Robert Rosenthal	Blumenstr. 12	München	81929	DE	Visa	78574	09/11
2035	Willibald Wiese	Nelkenstr. 23	Wiesenthal	09484	DE	Visa	38521	05/10
3023	Florence Fleur	Rue des fleurs 2	Lyon	76849	FR	American Express	80765	08/10
...

Kundenaufträge werden in der Tabelle „Bestellung“ verwaltet. Jeder Kundenauftrag besteht aus einer Bestellnummer (BstNr), einer Produktnummer (PNr), einer Kundennummer (KNr), einem Bestelldatum (Datum) und den Adressinformationen des Empfängers der Bestellung (EName, EAdr, EStadt, EPLZ, ELand):

Bestellung								
<i>BstNr</i>	<i>PNr</i>	<i>KNr</i>	<i>Datum</i>	<i>EName</i>	<i>EAdr</i>	<i>EStadt</i>	<i>EPLZ</i>	<i>ELand</i>
00001	3	3023	09.05.2010	Florence Fleur	Rue des fleurs 2	Lyon	76849	FR
00002	3	1001	02.03.2010	Nina Wiese	Blumenfeld 3	München	81929	DE
00003	1	1001	23.10.2010	Otto Lilienthal	Grasweg 7	Nürnberg	90001	DE
...

a) Formulieren Sie die folgenden Anfragen in voller Allgemeinheit (d.h. nicht in Bezug auf die obigen Beispieldaten) in **relationaler Algebra**:

- i. Alle Produkte, die über 30 Euro kosten.
- ii. Die Empfängernamen und Empfängeradressen aller Bestellungen ab dem 01.01.2010, die nach München geliefert werden.
- iii. Den Namen und die Stadt aller Kunden, die das Produkt mit der Produktnummer 12 bestellt haben.
- iv. Die Nummern aller Bestellungen, bei denen der Besteller und der Empfänger in derselben Stadt wohnen.

Was ist der Fehler in folgender relationaler Anfrage? Erklären Sie diesen kurz!

$$\pi_{PInfo}(Kunde \mid x \mid Bestellung)$$

b) Formulieren Sie die folgenden Anfragen in voller Allgemeinheit (d.h. nicht in Bezug auf die obigen Beispieldaten) in **SQL**:

- i. Die Empfängernamen und Empfängeradressen aller Bestellungen vom 01.01.2010, die nach Nürnberg geliefert werden.
- ii. Die Bestellnummer und die zugehörigen Informationen (EName, EStadt, ELand) aller Bestellungen, die über 40 Euro kosten.
- iii. Die Produktnummern der Produkte, die unter dem Durchschnittspreis aller angebotenen Produkte liegen.
- iv. Die Anzahl an registrierten Kunden pro Stadt.
- v. Die Anzahl an Bestellungen pro Stammkunde (Kunden mit mehr als 5 Bestellungen) mit dem jeweils im **Durchschnitt** ausgegebenen Geldbetrag.

Fortsetzung nächste Seite!

4. Normalformen

Gegeben sei der folgende Ausschnitt aus einer relationalen Flugbuchungs-Datenbank:

Buchung								
<i>KNr</i>	<i>KName</i>	<i>KOrt</i>	<i>Abflug</i>	<i>Flugnr</i>	<i>Abflugort</i>	<i>Zielort</i>	<i>Gesellschaft</i>	<i>Gezahlt</i>
18	Hans Meier	Berlin	10.10.2010/ 12:25	LH723	Frankfurt	New York	Lufthansa	VISA
42	Markus Baum	München	23.5.2010/ 9:30	BA534	München	Paris	British-Airways	Bar
48	Johann Hartl	Hamburg	26.9.2010/ 17:45	BA744	Hamburg	London	British-Airways	Konto
...

Wobei die folgenden funktionalen Abhängigkeiten gelten sollen:

KNr → *KName*, *KOrt*

Flugnr → *Abflugort*, *Zielort*, *Gesellschaft*

- Geben Sie sämtliche Schlüsselkandidaten der Relation Buchung an!
- Welchen Normalformen genügt die Relation Buchung? Begründung!
- Zeigen Sie anhand von Beispielen die drei möglichen Anomalien auf, die bei der Verwendung der Relation Buchung auftreten können!
- Erklären Sie, wie man die Relation „Buchung“ verändern müsste, damit sie in dritter Normalform vorliegt! Geben Sie auch eventuell neue Relationen oder Attribute an, die dafür nötig sein könnten!

5. Transaktionsverwaltung und Mehrbenutzersynchronisation

- Nennen und beschreiben Sie die vier grundlegenden Eigenschaften einer Datenbanktransaktion!
- Geben Sie zwei Fehler an, die im unkontrollierten (und nicht synchronisierten) Mehrbenutzerbetrieb auftreten können und beschreiben Sie diese jeweils anhand eines Beispiels!

Teilaufgabe 2:**1. E/R Modellierung, SQL-DDL**

Für den Buchhandel Dubenhugel soll ein System zur Verwaltung entworfen werden:

- Das Unternehmen besitzt mehrere Filialen, welche eindeutig durch die Filialen-ID gekennzeichnet sind. Jede Filiale hat eine Adresse (Strasse, Hausnummer, Ort und Postleitzahl) und einen Filialleiter. Daneben ist jede Filiale in mehrere Abteilungen thematisch untergliedert.
- Jede Abteilung hat eine eindeutige Abteilungs-ID und einen Abteilungsleiter.
- Jeder Mitarbeiter von Dubenhugel ist mindestens einer Abteilung zugeordnet. Für jeden Mitarbeiter sollen neben dem Namen auch die Adresse (gleiche Attribute wie Filialadresse), die Telefonnummer und das Geburtsdatum abgespeichert werden.
- Außerdem sollen in dem System die momentan vorhandenen Bücher gespeichert werden. Ein Buch hat eine ISBN-Nummer und einen Titel. Jedes Buch wurde von mindestens einem Autor verfasst. Zu jedem Buch soll bekannt sein, in welcher Filiale und Abteilung es sich befindet.
- Zu jeder Abteilung soll gespeichert werden, welche Bücher in welcher Stückzahl vorhanden sind.
- Zu den Autoren in der Datenbank soll der Name, das Geburtsdatum und der Geburtsort abgelegt werden.
- Zusätzlich sind zu jedem Buch ähnliche Bücher bekannt. Auch diese Information soll in das System einfließen.

- a) Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank.
- b) Setzen Sie das gegebene E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Identifizieren Sie dazu zunächst weitere Attribute, die in obigem Diagramm noch nicht enthalten sind und die gegebenen Entities in sinnvoller Weise beschreiben. Es sollten mindestens zwei Attribute pro Entity angegeben werden. Geben Sie die resultierenden Relationschemata in folgender Schreibweise an:

Relation (Attribut1, Attribut2, ..., AttributN)

Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen. Achten Sie auf eine geeignete Modellierung der Beziehungen (Relationships).

2. Relationale Algebra

Gegeben sei das folgende relationale Datenbank-Schema zur Verwaltung von Zehnkampf-Wettbewerben. Die Primärschlüssel-Attribute sind jeweils unterstrichen.

<u>ANr</u>	Name	Land
A1	Roman Sebrle	Tschechien
A2	Bryan Clay	USA
A3	Tomas Dvorak	Tschechien
A4	Dan O'Brien	USA
A5	Dimitri Karpow	Kasachstan

“Athlet”:

<u>DNr</u>	Bezeichnung	Art
D1	100m	Lauf
D2	Weitsprung	Sprung
D3	Kugelstossen	Wurf
D4	Hochsprung	Sprung
D5	400m	Lauf
D6	110m Hürden	Lauf
D7	Diskuswerfen	Wurf
D8	Stabhochsprung	Sprung
D9	Speerwerfen	Wurf
D10	1500m	Lauf

“Disziplin”:

<u>WNr</u>	Bezeichnung	Jahr
W1	Olympische Spiele	2004
W2	Olympische Spiele	2008
W3	Weltmeisterschaft	2004
W4	Weltmeisterschaft	2006
W5	Weltmeisterschaft	2008

“Wettkampf”:

<u>ANr</u>	<u>DNr</u>	<u>WNr</u>	Platzierung
A1	D2	W1	1
A1	D10	W1	4
A1	D7	W4	2
A2	D2	W1	3
A2	D2	W2	1
A3	D10	W1	2
A4	D7	W1	5
A4	D2	W3	2
A5	D3	W1	4
A5	D10	W4	3

“Teilnahme”:

Formulieren Sie die folgenden Anfragen in relationaler Algebra:

- Finden Sie die Namen aller Athleten aus Tschechien.
- Geben Sie Bezeichnungen und Jahre aller Wettkämpfe aus, an denen Athleten aus den USA teilgenommen haben.
- Gesucht sind die Bezeichnungen aller Disziplinen, für die Ergebnisse für die Weltmeisterschaft aus dem Jahr 2004 vorliegen.

Geben Sie das Ergebnis der folgenden Ausdrücke der relationalen Algebra als Tabellen an:

- $\pi_{DNr}(Teilnahme) \bowtie Disziplin$
- $\pi_{ANr, DNr, Platzierung}(\sigma_{Platzierung < 4}(Teilnahme)) \bowtie \pi_{DNr, Art}(Disziplin)$

3. SQL

Gegeben sei das relationale Datenbank-Schema aus Aufgabe 2.

“Athlet”:

<u>ANr</u>	Name	Land
A1	Roman Sebrle	Tschechien
A2	Bryan Clay	USA
A3	Tomas Dvorak	Tschechien
A4	Dan O'Brien	USA
A5	Dimitri Karpow	Kasachstan

“Disziplin”:

<u>DNr</u>	Bezeichnung	Art
D1	100m	Lauf
D2	Weitsprung	Sprung
D3	Kugelstossen	Wurf
D4	Hochsprung	Sprung
D5	400m	Lauf
D6	110m Hürden	Lauf
D7	Diskuswerfen	Wurf
D8	Stabhochsprung	Sprung
D9	Speerwerfen	Wurf
D10	1500m	Lauf

“Wettkampf”:

<u>WNr</u>	Bezeichnung	Jahr
W1	Olympische Spiele	2004
W2	Olympische Spiele	2008
W3	Weltmeisterschaft	2004
W4	Weltmeisterschaft	2006
W5	Weltmeisterschaft	2008

“Teilnahme”:

<u>ANr</u>	<u>DNr</u>	<u>WNr</u>	Platzierung
A1	D2	W1	1
A1	D10	W1	4
A1	D7	W4	2
A2	D2	W1	3
A2	D2	W2	1
A3	D10	W1	2
A4	D7	W1	5
A4	D2	W3	2
A5	D3	W1	4
A5	D10	W4	3

Formulieren Sie die folgenden Anfragen in SQL:

- Finden Sie die Namen und Herkunftsländer aller Athleten, die bei der Teilnahme an einem Wettkampf den ersten Platz erreicht haben.
- Geben Sie alle Attribute der Wettkämpfe aus, an denen kein Athlet teilgenommen hat.
- Geben Sie die Nummern und Namen aller Athleten zusammen mit der Anzahl der verschiedenen Disziplinen, in der jeder Athlet an einem Wettkampf teilgenommen hat. Sortieren Sie das Ergebnis absteigend nach dieser Anzahl.

Wie sieht die Ergebnisrelation zu folgenden Anfragen aus?

- select a.ANr, b.ANr from Athlet a, Athlet b where a.Land = b.Land and a.ANr <> b.ANr;
- select Bezeichnung from Disziplin where DNr in (select DNr from Teilnahme where ANr not in (select ANr from Athlet where Land <> 'Tschechien' and Land <> 'USA'));

Fortsetzung nächste Seite!

4. Entwurfstheorie

Gegeben sei die nachfolgende relationale Datenbank mit unterstrichenen Schlüsselattributten. Sie enthält die Daten von Bestellungen bei einem Versandhaus.

Relation "Bestellung":

<u>Waren</u> <u>ID</u>	<u>Kunden</u> <u>ID</u>	<u>Zustelldienst</u> <u>ID</u>	Kunden Name	Kunden Wohnort	Waren Bezeichnung	Waren Preis	Zustelldienst Name
1	1	3	Müller	München	Blumentopf	10	DHL
2	1	2	Müller	München	Decke	12	UPS
3	2	1	Huber	Nürnberg	Rasierer	26	FedEx
2	2	3	Huber	Nürnberg	Decke	12	DHL
5	3	2	Meier	Hamburg	Gläser	26	UPS
6	4	1	Meier	München	Block	8	FedEx

- Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.
- Geben Sie für das obige Datenbankschema alle funktionalen Abhängigkeiten (inkl. der transitiven) an.
- Überführen Sie das obige Relationenschema in die dritte Normalform.
- Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung hat.

Themenschwerpunkt B
(Betriebssysteme)

Teilaufgabe 1:

1. Verklemmung

- a) Nennen Sie alle Bedingungen für das Auftreten von Verklemmungen (Deadlocks).
- b) Fünf Philosophen haben in ihrer wöchentlichen Diskussionsrunde ein regelmäßiges Problem. Sie sitzen an einem runden Tisch, auf dem eine Schüssel mit Spaghetti steht. Zwischen jeweils zwei Philosophen liegt eine Gabel. Somit liegen insgesamt fünf Gabeln auf dem Tisch. Wenn ein Philosoph hungrig ist, greift er zuerst die Gabel links von seinem Teller und dann die Gabel rechts. Danach beginnt er zu essen. Ein Philosoph braucht immer zwei Gabeln zum Essen. Wenn er satt ist, legt er seine Gabeln wieder zurück. Danach diskutiert er mit den anderen Philosophen weiter. Liegt eine Gabel nicht an ihrem Platz, wartet ein Philosoph solange, bis die Gabel wieder verfügbar ist.
 - i. Welches Problem tritt regelmäßig auf? Begründen Sie ihre Antwort in 3-4 Sätzen.
 - ii. An manchen Abenden kann ein einzelner Philosoph nicht kommen. Somit sitzen die anderen nur zu viert an dem Tisch, der für fünf Philosophen gedeckt ist. Tritt das Problem aus a) an diesen Abenden auch auf? Begründen Sie Ihre Antwort in 2-3 Sätzen.
 - iii. Die fünf Philosophen haben sich nach längerer Diskussion auf folgendes Verfahren geeinigt. Sie installieren eine Tischglocke, welche alle fünf Minuten klingelt. Beim Klingeln muss der Philosoph, welcher am längsten seine Gabel(n) hält, diese auf den Tisch zurücklegen. Beweisen Sie in Form eines Widerspruchsbeweises, dass dieses Verfahren das Problem der Philosophen löst.

2. Synchronisation

Für die Führungen in einem Museum soll ein neues Besucher-Warte-System eingeführt werden. Das Museum hat genau einen Museumsführer und einen Warteraum für maximal 10 Besucher. Wenn ein Besucher zum Museum gelangt, kann er nur dann den Warteraum betreten, wenn ein Platz frei ist. Der Zugang zum Warteraum ist ein langer schmaler Gang. Durch diesen kann immer nur in einer Richtung gegangen werden. Der Museumsführer nimmt immer genau einen Besucher bei einer Führung mit. Wenn gerade kein Besucher im Warteraum ist, legt sich der Museumsführer schlafen.

```
public interface Semaphore {
    public void wait();
    public void signal();
}

public class BinarySemaphore implements Semaphore {
    public BinarySemaphore() {
        [...]
    }
    [...]
}

public class CountingSemaphore implements Semaphore {
    public CountingSemaphore(int initialCount) {
        [...]
    }
    [...]
}
```

Vervollständigen Sie die Codeschablonen in den Teilaufgaben a), b) und c), indem Sie die Kommentare der Form // Ergänzung X durch Code für die Initialisierung der verwendeten Semaphore bzw. für die Synchronisation der Threads der Klassen Besucher und Museumsfuehrer ersetzen. Sie können an den mit // Ergänzung X gekennzeichneten Stellen beliebig viele Codezeilen oder auch keine einfügen. Schreiben Sie Ihre Lösung in folgender Form auf:

Ergänzung	Code
1	<code>new BinarySemaphore()</code>
2	<code>...</code>
3	<code>...</code>
...	<code>...</code>

Ihre Lösung soll einen möglichst hohen Grad an Parallelität ermöglichen.

- a) Vervollständigen Sie die folgenden Initialisierungen der Semaphore.

Codeschablone für die Initialisierung:	
// Zugriff auf den Warteraum des Museums	
Semaphore mutex = // Ergänzung 1 ;	
// Zählt die Besucher im Warteraum des Museums	
Semaphore vistor = // Ergänzung 2 ;	
// Zählt die freien Plätze im Warteraum des Museums	
Semaphore empty = // Ergänzung 3 ;	

- b) Ergänzen Sie die folgende Vorlage für die Klasse Besucher.

```
public Besucher extends Thread {  
    public void run() {  
        while (active) {  
            museumsbesuch();  
        }  
    }  
    private void museumsbesuch() {  
  
        // Ergänzung 4  
  
        // Besucher nimmt im Warteraum Platz  
  
        // Ergänzung 5  
  
    }  
    [...]  
}
```

- c) Ergänzen Sie die folgende Vorlage für die Klasse Museumsfuehrer.

```
public Museumsfuehrer extends Thread {  
    public void run() {  
        while (active) {  
            fuehren();  
        }  
    }  
    private void fuehren() {  
  
        // Ergänzung 6  
  
        // Besucher aus dem Warteraum holen.  
  
        // Ergänzung 7  
  
        // Besucher durch das Museum führen.  
  
        // Ergänzung 8  
  
    }  
    [...]  
}
```

3. Prozess-Scheduling

- a) Nennen Sie vier Optimierungsziele eines CPU-Schedulers.

Gegeben sind die folgenden Prozesse:

Prozess	P0	P1	P2	P3	P4
Ankunftszeit	0	1	2	3	4
Benötigte Rechenzeit	7	5	10	2	6

- b) Führen Sie das Scheduling für die gegebenen Prozesse mit den Algorithmen *First Come First Served (FCFS)*, *Round Robin (RR)* (Zeitscheibe der Größe 2) und *präemptivem Shortest Remaining Time Next (SRTN, preemptive)* durch. Stellen Sie den Ablauf inklusive der Zustände der Prozesse (Wartend, Rechnend) als Gantt-Diagramm (Balkendiagramm) dar.

Hinweis: Bei Round Robin kann es vorkommen, dass zum gleichen Zeitpunkt ein Prozess eintrifft und einem anderen Prozess die CPU entzogen wird. Gehen Sie in diesem Fall davon aus, dass der ankommende Prozess zuerst in die Queue eingeordnet wird. Gehen Sie desweiteren bei SRTN davon aus, dass für Prozesse mit gleichen verbleibenden Rechenzeiten FCFS verwendet wird.

- c) Geben Sie für *Round Robin* die durchschnittliche Wartezeit und den Durchsatz an.

4. Speicherverwaltung

In einem Betriebssystem ist eine Seitenrahmentabelle vorhanden, welche für maximal vier Seiten ausgelegt ist.

Gegeben ist eine Folge von Seitenzugriffen:

Seitenzugriffe P0 P1 P2 P3 P4 P2 P0 P1

Führen Sie die Seitenzugriffe unter Verwendung des Seitenersetzungsalgorithmus *Least Recently Used (LRU)* durch. Stellen Sie die Einträge in der Seitenrahmentabelle für jeden Zugriff dar. Geben Sie außerdem an, ob ein Seitenfehler aufgetreten ist.

Sie können Ihre Lösung zum Beispiel in Tabellenform angeben.

Seite	(Seite 1)	(Seite 2)	(Seite 3)	(Seite 4)	Seitenfehler
P0					
P1					
P2					
P3					
P4					
P2					
P0					
P1					

Teilaufgabe 2:**1. Seitenersetzung**

In dieser Aufgabe sollen verschiedene Seitenersetzungssstrategien (Paging-Strategien) am konkreten Beispiel verglichen werden. Dabei sei die Menge der Seiten gegeben durch $N = \{A, B, C, D, E\}$. Die Menge der Seitenrahmen, die für die Speicherung der Seiten im Arbeitsspeicher zur Verfügung stehen, sei gegeben durch $F = \{f_0, f_1, f_2\}$. Auf die fünf Seiten der Menge N werde in folgender Reihenfolge zugegriffen:

$$\{w = A \text{ E } D \text{ E } D \text{ B } D \text{ E } A \text{ D } C \text{ B } A \text{ E } \}$$

Der Arbeitsspeicher ist zu Beginn leer.

- a) Ermitteln Sie die Anzahl der Seitenfehler für die Paging-Strategie LFU (Least Frequently Used), indem Sie alle Veränderungen im Speicher in der folgenden Tabelle dokumentieren. Tragen Sie außerdem alle Informationen in die Tabelle ein, die aufgrund der Paging-Strategie zusätzlich benötigt werden.

Referenzierte Seiten	f_0	f_1	f_2	Summe Seitenfehler
A				
E				
D				
E				
D				
B				
D				
E				
A				
D				
C				
B				
A				
E				

- b) Nehmen Sie an, dass Sie nun die Anzahl der Seitenrahmen im Arbeitsspeicher erhöhen. Wie verhält sich dann die Anzahl der Seitenfehler? Begründen Sie ohne eine neue Tabelle anzufertigen.

Fortsetzung nächste Seite!

2. Prozesse

- a) Welche drei Arten von Informationen enthält der Prozesskontrollblock (PCB)?
- b) Was ist der Unterschied zwischen PCB und Prozess-Image?
- c) Welche Vorteile ergeben sich, wenn ein Prozessmodell zwischen vielen verschiedenen Prozesszuständen differenziert?
- d) Welche Nachteile ergeben sich, wenn ein Prozessmodell zwischen vielen verschiedenen Prozesszuständen differenziert?
- e) In den folgenden Fragen soll das 5-Zustands-Prozessmodell betrachtet werden.
 - i) Skizzieren Sie das 5-Zustands-Prozessmodell. Kennzeichnen Sie alle Übergänge durch Pfeile und benennen Sie die Übergänge.
 - ii) Nummerieren Sie die Übergänge in Ihrer Skizze. Geben Sie für jeden Übergang ein Beispiel an.

Fortsetzung nächste Seite!

3. Multilevel Feedback Queueing (MLFQ)

In dieser Aufgabe soll die Scheduling-Strategie Multilevel Feedback Queueing genauer untersucht werden.

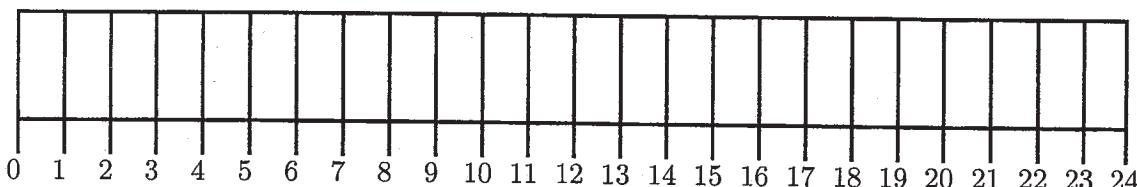
Gegeben seien die Prozesse A bis E mit den folgenden Ankunfts- und Rechenzeiten:

Prozess	Ankunftszeitpunkt	Rechenzeit
A	0	9
B	1	5
C	2	1
D	5	5
E	10	2

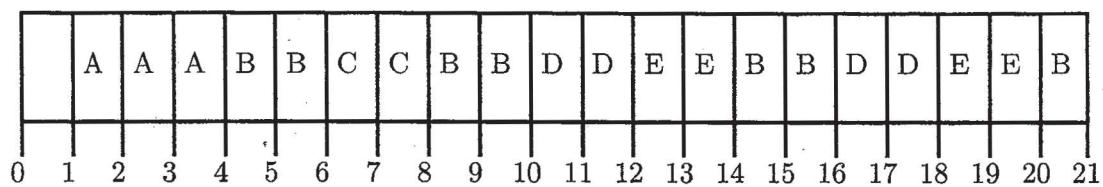
Es gilt: Der Ankunftszeitpunkt t bezeichnet auch den Zeitpunkt, zu dem der Prozess in die Warteschlange eingereiht wird (ohne jede Verzögerung). Wird ein Prozess vor seinem Terminieren zum Zeitpunkt t' unterbrochen, so reiht er sich in die entsprechende Warteschlange mit Ankunftszeit t' wieder ein. Sollten zwei Prozesse zum gleichen Zeitpunkt in die Warteschlange eingeordnet werden, so wird die lexikalische Ordnung auf den Prozessnamen herangezogen (Prozess A vor Prozess B, usw.).

- a) Geben Sie für die **Strategie Multilevel Feedback Queueing (MLFQ)** an, wann welchem Prozess Rechenzeit zugeteilt wird und wann die Prozesse jeweils terminieren, indem Sie das nachfolgende Gantt-Chart vervollständigen. Gehen Sie von drei Prioritätstypen (0 ist die höchste Priorität) mit folgenden Werten aus:

Priorität	Verfahren
0	Round-Robin mit Quantum 1
1	Round-Robin mit Quantum 3
2	First Come First Served (FCFS)



b) Gegeben sei folgendes Gantt-Chart:



Die Ankunftszeiten und Rechenzeiten der Prozesse sind dabei wie folgt gegeben:

Prozess	Ankunftszeitpunkt	Rechenzeit
A	0	3
B	2	7
C	2	2
D	5	4
E	7	4

Hinweis: es handelt sich dabei **nicht** um ein Gantt-Chart, das sich aus den gegebenen Prozessen nach Anwendung der vorhin beschriebenen MLFQ-Strategie ergibt.

Berechnen Sie für alle hier angegebenen Prozesse jeweils die Verweilzeit und die Wartezeit sowie (als Dezimalzahl mit einer Nachkommastelle) die mittlere Verweilzeit und die mittlere Wartezeit.

4. Synchronisation nebenläufiger Prozesse

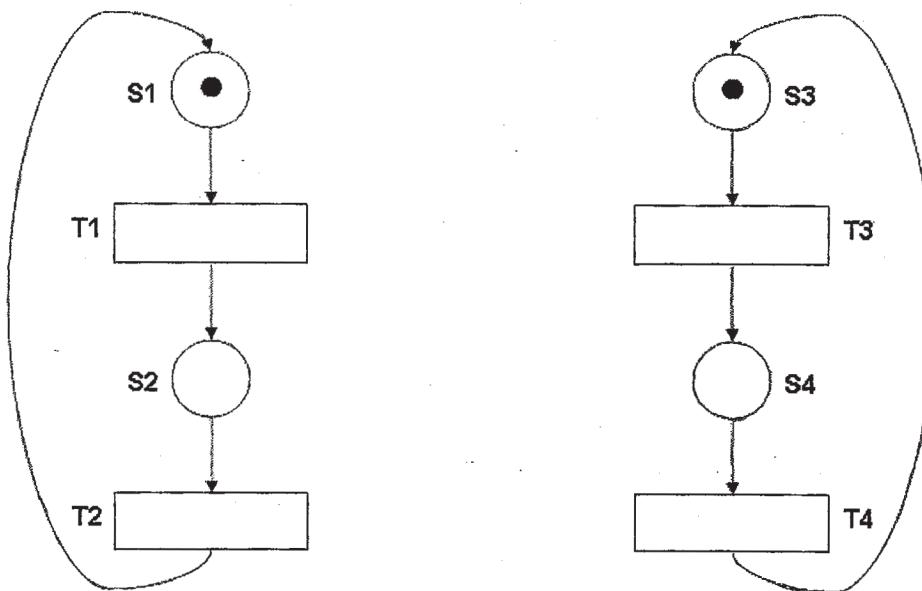
Wenn Prozesse nebenläufig ausgeführt werden, macht das eine geeignete Synchronisation dieser Prozesse erforderlich. Um den wechselseitigen Ausschluss der Zugriffe auf kritische Bereiche zu realisieren, können Binärsemaphore wie folgt eingesetzt werden:

```
BinarySemaphore mutex;
init(mutex, 1);
```

P:

```
REPEAT
    <rechne im unkritischen Bereich>
    wait(mutex);
    <rechne im kritischen Bereich>
    signal(mutex);
    <rechne im unkritischen Bereich>
UNTIL FALSE;
```

Beispielhaft für **zwei** Prozesse P_1 und P_2 soll dieses Synchronisationsproblem durch ein Petrinetz modelliert werden. Die Stellen können als Zustände interpretiert werden. Dabei entspricht S_1 (S_3) dem Zustand " P_1 (P_2) rechnet im unkritischen Bereich", und S_2 (S_4) dem Zustand " P_1 (P_2) rechnet im kritischen Bereich".



- a) Übertragen Sie dieses Petrinetz auf Ihr Klausurpapier und ergänzen Sie dieses um weitere Stellen, Transitionen und Kanten (sofern jeweils erforderlich), sodass zu jedem Zeitpunkt höchstens ein Prozess in seinem kritischen Bereich rechnen kann. Verwenden Sie hierfür **keine** Inhibitorkanten. Zeichnen Sie direkt in das obige Petrinetz.
- b) Geben Sie den Erreichbarkeitsgraphen für Ihr Petrinetz an, und beschriften Sie darin alle Übergänge mit den Namen der jeweiligen Transitionen. Begründen Sie, ob das Petrinetz verklemmt ist oder nicht.
- c) Das Beispiel soll nun zu einem Erzeuger/Verbraucher-Szenario erweitert werden. Der Prozess P_1 erzeugt Daten und schreibt diese in einen gemeinsamen Speicher mit 5 Plätzen. Der Prozess P_2 liest diese Daten. Dazu werden die beiden Zählsemaphoren `platz` und `bestand` eingeführt und wie folgt initialisiert:

```
Semaphore platz, bestand;  
init(platz, 5);  
init(bestand, 0);
```

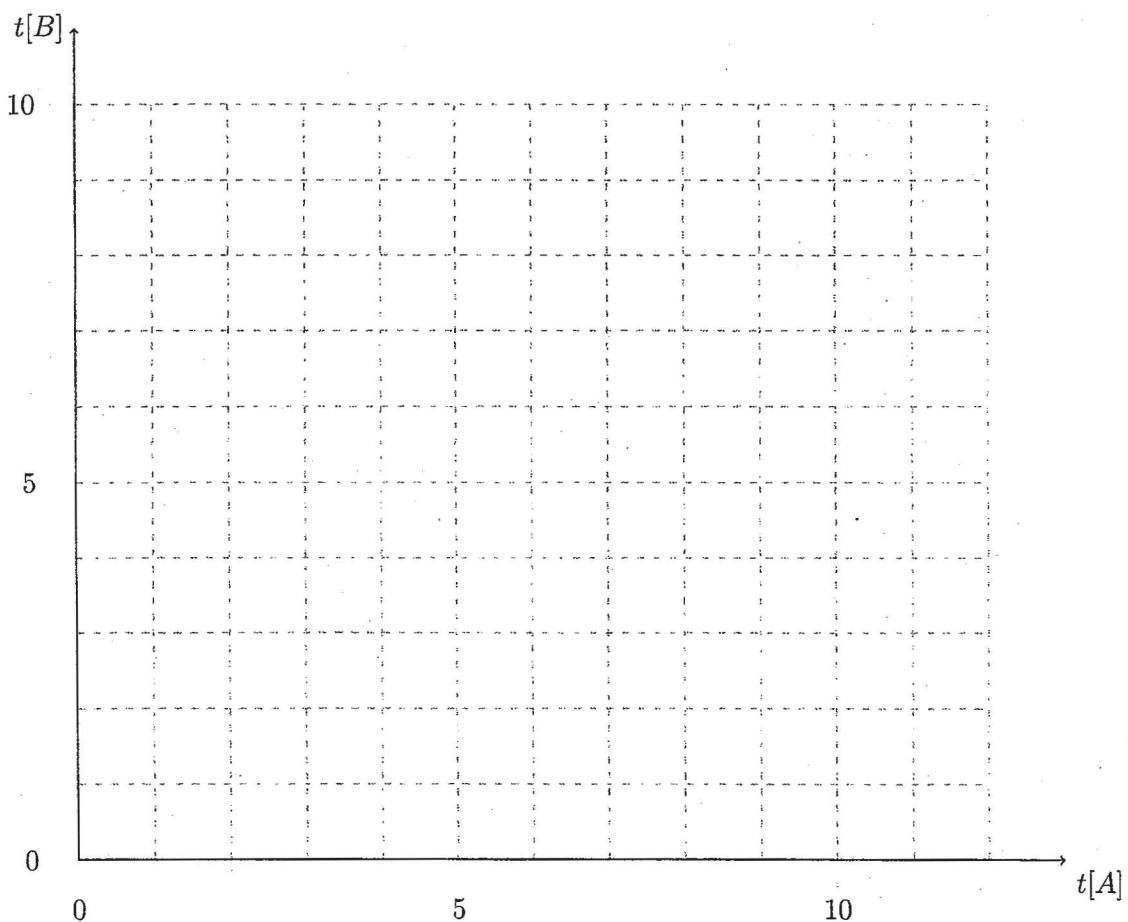
Ergänzen Sie die folgenden Prozessdefinitionen unter Verwendung dieser Semaphoren so, dass P_1 nicht in den vollen Speicher schreiben und P_2 nicht aus einem leeren Speicher lesen kann.

P1: <rechne im unkritischen Bereich>	P2: <rechne im unkritischen Bereich>
wait(mutex);	wait(mutex);
<erzeuge Element>	<liest Element>
signal(mutex);	signal(mutex);
<rechne im unkritischen Bereich>	<rechne im unkritischen Bereich>

5. Prozessfortschrittsdiagramm

Gegeben seien zwei Prozesse A und B . A benötigt zu seiner Ausführung zwölf Zeiteinheiten, B zehn Zeiteinheiten. Es stehen 3 Betriebsmittel (BM) zur Verfügung, die von den Prozessen während ihrer Ausführung benötigt werden. A benötigt BM1 im Zeitraum 2 – 6, BM2 in 4 – 7, BM3 in 8 – 10 und Prozess B benötigt BM1 in 5 – 8, BM2 in 4 – 7 und BM3 in 1 – 3.

- a) Skizzieren Sie im unten angegebene Diagramm das Prozessfortschrittsdiagramm! Zeichnen Sie alle unmöglichen und unsicheren Bereiche ein und beschriften Sie diese entsprechend! Wo genau tritt ggf. der Deadlock ein?



- b) Kann es bei nicht-preemptivem Scheduling zu einem Deadlock kommen? Begründen Sie Ihre Antwort und zeichnen Sie für nicht-preemptives Scheduling alle *prinzipiell* (!) verschiedenen Möglichkeiten der Abarbeitung von Prozess A und B in ihrer Abbildung aus Aufgabe a) ein!
- c) Wie viele verschiedenen Möglichkeiten gibt es *insgesamt* (!) bei preemptivem Scheduling die Prozesse erfolgreich terminieren zu lassen?

6. Deadlock-Vermeidung

Wenn man eine der Bedingungen für die Entstehung eines Deadlocks bei der Konzeption eines Betriebssystems von Anfang an ausschliesst, können Deadlocks gänzlich vermieden werden. Benennen und erklären Sie die vier Bedingungen, die zur Entstehung eines Deadlocks führen.