Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr
2009

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 2 gemäß untenstehender Auswahlregel zu bearbeiten sind

Anzahl der Druckseiten dieser Vorlage: 20

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

AUFGABE 1

1. Modellierung

Ein Pizzadienst möchte sein Bestellsystem mit einer relationalen Datenbank abwickeln:

Die Datenbank enthält allgemein Personen, die sich immer in Kunden und Personal unterteilen lassen. Zu jeder Person gibt es eine Nummer (PNr), den Namen und den Vornamen. Zu Kunden wird auch die Adresse (enthält Ort, PLZ, Straße und Hausnummer) gespeichert, zum Personal der Lohn. Die Mitglieder des Personals können aber auch eigene Bestellungen aufgeben. In so einem Fall werden zusätzlich auch die Kundeninformationen gespeichert.
Eine Bestellung wird immer einem Kunden zugeordnet. Die Bestellung besitzt eine eindeutige Bestellungsnummer (BNr) und sowohl Datum als auch Uhrzeit der Bestellung. Verknüpft mit der Bestellung werden beliebig viele, aber mindestens eine Bestellzeile gespeichert, welche die bestellten Gerichte aufführen. Die Zeilen einer Bestellung werden angefangen bei "1" durchnummeriert. Zu jeder Bestellzeile wird exakt 1 Gericht verknüpft. Der korrespondierende Preis wird in der Bestellzeile gespeichert.
Gerichte besitzen eine eindeutige Nummer (GNr), einen Namen und eine optionale Beschreibung. Gerichte unterteilen sich in die Typen Nudeln, Pizzen und Muscheln. Die Unterteilung geschieht natürlich überschneidungsfrei. Den Pizzen sollen insbesondere die enthaltenen Zutaten zugeordnet werden, welche eigens gespeichert werden. Jede Pizza muss aber mindestens eine Zutat enthalten. Die Zutaten können anhand ihres Namens unterschieden werden.
Es kommt immer wieder vor, dass Sonderwünsche an Speisen bestellt werden. Diese sind dann mit eigener Nummer als Gericht zu speichern. Weiter ist es möglich für Pizzen beliebig viele zusätzliche Zutaten zu ordern. Die Aufzählung dieser Zutaten erfolgt durch Erweiterung der Verknüpfung zwischen Bestellzeile und Gericht.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute;
- ein passendes ER-Diagramm;
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen;
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

Fortsetzung nächste Seite!

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an.

c) Laut Aufgabenstellung ist es nur bei Pizzen erlaubt, optionale Zusatzzutaten zu bestellen. Das bisherige Relationenmodell gestattet es aber, Zusatzzutaten für jedes Gericht anzugeben. Dies soll in der Datenbank anhand von Integritätsbedingungen verhindert werden.

   i) Welche Arten von Integritätsbedingungen lassen sich allgemein unterscheiden?

   ii) Welche Ansätze bieten relationale Datenbanken zur Berücksichtigung der unterschiedlichen Arten von Integritätsbedingungen? Nennen Sie pro Art aus Teilaufgabe (i) einen möglichen Ansatz mit kurzer Begründung.

Fortsetzung nächste Seite!

2. Normalformen

Gegeben sei folgende Datenbasis:

ANr	Datum	Zeile	ProduktID	Hersteller	KNr	Kunde	SB	Kommentar	PreisEuro
1	2.3.2004	1	PC160X	Semins	356	Meierhof 1976	SL3	Interesse an RAID1	900
1	2.3.2004	2	LCD17A	Sitech	356	Meierhof 1976	HH5	-	120
1	2.3.2004	3	THZ99DB	Heifi	356	Meierhof 1976	HH5	Anbindung zu LCD17A	250
2	1.7.2005	1	HIFI2000	Datavolt	235	Hemebach 1982	HH5	Erweiterung für DDSS 8.9	1200
3	5.7.2005	1	SDU23X8	Sitech	56	Craiton 1964	SL3	-	1500
...	...	...	...						

Für jedes Angebot wird eine eindeutige Angebotsnummer (ANr) vergeben und das Datum des Angebots gespeichert. Ein Angebot wird immer für genau einen Kunden angelegt. Zu jedem Kunden werden Name und Geburtsjahr gespeichert. Ein Angebot umfasst mehrere Angebotszeilen (Zeile). Jede Zeile repräsentiert ein angebotenes Produkt, zu dem jeweils auch der Hersteller gespeichert wird. Für jede Angebotszeile wird ein(e) SachbearbeiterIn angegeben, der/die für den jeweiligen Teil des Angebotes zuständig ist. Jede(r) SachbearbeiterIn hat die Möglichkeit einen eigenen Kommentar zur Angebotszeile abzugeben, welcher für die interne Verwendung im Unternehmen gedacht ist. Zusätzlich kann der Preis jeder Angebotszeile einzeln festgelegt werden. Der Preis einer Angebotszeile muss nicht mit dem Preis des angebotenen Produktes übereinstimmen, sondern kann davon abweichen. Der/die SachbearbeiterIn kann den Preis stattdessen nach eigenem Ermessen anpassen.

a) Wo verletzt die Datenbasis bereits die 1. Normalform?

b) Welche Probleme (Anomalien) treten bei dieser Datenbasis auf, die sich durch eine Überführung in die Boyce-Codd-Normalform beheben lassen?

c) Beschreiben Sie obige Anforderungen mit einer Menge von nichttrivialen funktionalen Abhängigkeiten für die Datenbasis.

d) Formen Sie das gegebene Schema in die 3. Normalform um. Geben Sie hierfür zunächst ein Schema in 1. Normalform an, welches Sie anschließend in die 3. Normalform überführen.

Fortsetzung nächste Seite!

3. Relationale Algebra und SQL

Gegeben sei folgendes Schema der relationalen Datenbank eines Transportunternehmens:

$Ladung\{\underline{LNr}, GewichtKG, VolumenM, StreckeKM\}$

$Fahrzeug\{\underline{FzNr}, Hersteller, MaxLastKG, MaxVolumenM, Stillgelegt\}$

$Fahrer\{\underline{FrNr}, Vorname, Nachname, Ort\}$

$Transport\{\underline{LNr}, \underline{TNr}, FzNr, FrNr, VonOrt, VonZeit, NachOrt, NachZeit, LaengeKM\}$

Jede beauftragte Ladung wird mit einer Nummer (LNr) versehen. Neben Gewicht (in Kilogramm) und Volumen (in Kubikmeter) wird auch die Gesamtlänge der Transportstrecke (in Kilometer) einer Ladung eingetragen.
Für den Transport stehen verschiedene Fahrzeuge und Fahrer zur Verfügung. Fahrzeuge besitzen eine Maximalbeschränkung bezüglich Last und Volumen ihrer Ladung. Es ist prinzipiell möglich verschiedene Ladungen in einem Fahrzeug gleichzeitig zu transportieren, so lange die Gesamtkapazität des Fahrzeuges nicht überschritten wird.
Der eigentliche Transport einer Ladung kann in einem Stück bis zum Zielort erfolgen oder in mehrere Teilstrecken zerlegt werden. Wird eine Zerlegung vorgenommen, so können die einzelnen Teilstrecken von unterschiedlichen Fahrzeugen und/oder Fahrern gefahren werden. Die Nummern für die einzelnen Teilstrecken (TNr) werden dabei fortlaufend ab 1 vergeben.

a) Formulieren Sie die Anfragen in den folgenden Teilaufgaben (i - ii) in der Anfragesprache SQL. Zur Vereinfachung dürfen Sie sich beliebige Sichten (Views) definieren.

   i) Geben Sie Vorname und Nachname der fünf Fahrer mit der größten Anzahl gefahrener Kilometer aus.

   ii) Geben Sie gruppiert nach Hersteller die durchschnittlich je Fahrzeug transportierte Last aus.

b) Formulieren Sie in relationaler Algebra die Anfragen in den folgenden Teilaufgaben (iii - iv).

   iii) An welchen Ort wurde die Ladung mit der Nummer "341" als letztes geliefert (Zielort)? Sie können dazu annehmen, dass Start- und Zielort der Ladung verschiedene Namen besitzen.

   iv) Welche Fahrer hatten mindestens eine Fahrt mit Stopp im Ort "Rom"? Geben Sie Vorname, Nachname und Hersteller des verwendeten Fahrzeuges aus.

Fortsetzung nächste Seite!

4. Transaktionen

Gegeben sei folgender, aus elementaren Operationen der Transaktionen T_1, T_2 und T_3 bestehender Schedule:

$$s_{in} = r_2(A)w_3(A)w_2(C)w_1(C)w_3(A)r_1(A)r_3(B)w_2(B)c_2c_3c_1$$

a) Geben Sie den Konfliktgraphen für s_{in} an. Wie kommen die Knoten und Kanten zu Stande?

b) Welche Aussage lässt sich aus dem Konfliktgraphen von Teilaufgabe a) über den zugrunde liegenden Schedule treffen? Begründen Sie Ihre Antwort.

c) Bestimmen Sie einen möglichen Output-Schedule s_{out}, den ein Scheduler erzeugt, der dem strikten 2-Phasen-Sperr-Protokoll folgt. Geben Sie im Output-Schedule insbesondere auch die erzeugten *lock*− und *unlock*−Befehle an.

Aufgabe 2

1. Normalformenlehre

a) Wann ist ein Superschlüssel Schlüsselkandidat?

b) Welche Aussagen sind für funktionale Abhängigkeiten richtig? Begründen Sie Ihre Antwort kurz.

i) $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$

ii) $X \rightarrow Y$ und $Y \rightarrow Z \Rightarrow X \rightarrow Z$

iii) $X \rightarrow Y$ und $Z \rightarrow Y \Rightarrow X \rightarrow Z$ oder $Z \rightarrow X$

iv) $W \rightarrow X$ und $Y \rightarrow Z \Rightarrow WY \rightarrow XZ$

c) Warum ist folgende Relation nicht in dritter Normalform?
`Adresse(Straße, Hausnummer, Postleitzahl, Wohnort)`
Warum werden Adressen dennoch häufig so gespeichert?

d) Wie muss Relation
`R(A, B, C, D, E)`
mit weiteren funktionalen Abhängigkeiten $B \rightarrow D$ und $C \rightarrow E$ zerlegt werden, damit sie in zweiter Normalform vorliegt? Wie muss sie weiter zerlegt werden, so dass sie in dritter Normalform vorliegt?

Fortsetzung nächste Seite!

2. ER-Modellierung und Relationenmodell

a) **ER-Modellierung**

Erstellen Sie das Modell eines fiktiven Krankenhauses in ER-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. Verwenden Sie zur Angabe der Kardinalität die (Min, Max)-Notation. Führen Sie Surrogatschlüssel nur ein, falls es nötig ist.

„Im Krankenhaus"

Zimmer werden durch eine eindeutige Zimmernummer identifiziert. Weiterhin ist noch die Anzahl der Betten im Zimmer angegeben.
Stationäre **Patienten** liegen in maximal einem Zimmer, ambulante Patienten dagegen liegen nicht in einem Zimmer. Zimmer können leer sein oder bis zu vier Patienten beherbergen. Patienten haben einen Vornamen, Nachnamen und ein Geburtsdatum.
Behandlungen werden durch ein Datum und eine Uhrzeit beschrieben. An einer Behandlung nimmt genau ein Patient teil. Jeder Patient wird mind. einer Behandlung unterzogen.
Ärzte werden durch Vorname, Nachname und eine eindeutige Piepsernummer beschrieben.
Fachgebiete werden durch einen Namen beschrieben und durch eine Abkürzung eindeutig identifiziert.
Behandlung, Fachgebiet und Arzt stehen in einer engen Beziehung: Im Rahmen einer Behandlung beherrscht jeder Arzt genau ein Fachgebiet und es gibt für jedes Fachgebiet genau einen Arzt. Natürlich kann ein Arzt ein Fachgebiet nicht nur im Rahmen einer einzigen, sondern bei mehreren Behandlungen vertreten.

b) **Relationenmodell**

Ausgehend von der ER-Darstellung ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. **Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht, z.B.:**

`Haus (Straße, OrtId[Ort])`
`Ort (OrtId, PLZ, Name)`

Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

Fortsetzung nächste Seite!

3. SQL

Bitte beachten Sie: Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributsnamen kenntlich gemacht.

Beispiel:
`Haus (Straße, OrtId[Ort])`
`Ort (OrtId, PLZ, Name)`

Interpretation: Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

a) **Szenario 1: „Musikverwaltung“**
Eine Musikliebhaberin verwendet folgendes einfache Datenbankschema für die Verwaltung ihrer CD-Sammlung:

```
Künstler (KId, Name, Musikrichtung, Webseite)
CD (CDId, Name, KId[Künstler], Spielzeit, VöD)
Track (TId, CDId[CD], Nummer, Name, Spielzeit)
```

Jeder Künstler veröffentlicht CDs, die wiederum aus mehreren Tracks bestehen können. `VöD` in der Tabelle `CD` bezeichnet das Veröffentlichungsdatum. Primärschlüssel sind unterstrichen. `KId` in `CD` ist Fremdschlüssel zu `KId` in `Künstler`, `CDId` in `Track` ist Fremdschlüssel zu `CDId` in `CD`.

Formulieren Sie bitte folgende Datenbankoperation in SQL:

i) Geben Sie die Namen und Spielzeiten aller Tracks der CD mit der CDId „42“ aus!

ii) Geben Sie die Namen der CDs mit der längsten Spielzeit aus!

iii) Geben Sie eine Liste der Namen aller CDs des Künstlers „Robbie Williams“, aufsteigend sortiert nach VöD, aus!

iv) Geben Sie die durchschnittliche Spielzeit aller Tracks der CDs des Künstlers „Sting“ aus!

Die Musikliebhaberin möchte in Zukunft neben CDs auch andere Medientypen, wie z.B. DVDs oder Kassetten verwalten.

v) Beschreiben Sie in jeweils max. 3 Sätzen, zwei grundlegend unterschiedliche Methoden zur Erweiterung des Datenbankschemas um die zusätzlichen Medientypen.

vi) Geben Sie für jede der beiden genannten Alternativen genau einen Vor- und einen Nachteil an!

vii) Geben Sie jeweils komplett die beiden alternativen Datenbankschemata nach der Berücksichtigung der Medientypen an!

Fortsetzung nächste Seite!

b) **Szenario 2: „Bei der Feuerwehr“**

Die Kreisfeuerwehr Eggolshausen wird unter Obereinsatzleiter Mayerle modernisiert. Die Rettungskräfte haben zur Einsatzverwaltung eine eigene kleine Datenbank entwickelt. Diese ist schon seit einigen Wochen in Betrieb und muss derzeit noch per Hand mit SQL-Befehlen gepflegt werden.

```
Anschrift (AID, Straße, Nr, PLZ, Stockwerk)
Feuerwehrmann (PID, Name, Vorname, EID[Einsatzfahrzeug])
Einsatzfahrzeug (EID, Kennzeichen, Plätze, Ausrüstung)
Einsatz (AID[Anschrift], EID[Einsatzfahrzeug], Datum, Ursache)
```

Es existieren folgende Fremdschlüsselbeziehungen: Das Attribut `AID` der Relation `Einsatz` bezieht sich auf das Attribut `AID` der Relation `Anschrift`. Das Attribut `EID` der Relation `Einsatz` bezieht sich auf das Attribut `EID` der Relation `Einsatzfahrzeug`. Die Attribute `AID` und `Nr` der Relation `Anschrift`, `PID` der Relation `Feuerwehrmann` und `EID` der Relation `Einsatzfahrzeug` sind Ganzzahlen. Bei dem Attribut `Datum` der Relation `Einsatz` handelt es sich um ein Datum. Das Attribut `PLZ` der Relation `Anschrift` ist eine fünfstellige Ganzzahl. Die Attribute `Kennzeichen` und `Ausrüstung` der Relation `Einsatzfahrzeug` werden als Text gespeichert. Die Datentypen der restlichen Attribute sind dem Kontext zu entnehmen.

Geben Sie SQL-Anweisungen für folgende Problemstellungen an:

i) Erzeugung des beschriebenen Relationenschemas mit Tabellen, Primary Key Constraints und Foreign Key Constraints!

ii) Einem Haus in der Heinestr. 4 in 99099 Eggolshausen wurde ein neues drittes Stockwerk aufgesetzt. Nun soll das neue Stockwerk eingetragen werden.

iii) Nachdem der Bungalow in der Musterstr. 3 in 99099 Eggolshausen bereits mehrfach gebrannt hat und auch Löscheinsätze gefahren wurden, wird er jetzt abgerissen. Löschen Sie den Bungalow aus der Tabelle `Anschrift` und berücksichtigen Sie auch ggf. davon abhängige Datensätze.

iv) Ein Einsatz wurde gefahren: Es brannte erstmalig in der Weidenstr. 8 im 4. Stock in 99099 Eggolshausen am 15.07.2006. Das Löschfahrzeug mit dem Kennzeichen EGG-UM 0815 und der `EID` 7 wurde zum Einsatz beordert. Tragen Sie den Einsatz und ggf. weitere notwendige Datensätze in die Datenbank ein.

v) Geben Sie eine Liste aller Einsatzursachen, sortiert nach Häufigkeit der Einsatzursache aus.

Themenschwerpunkt B

Betriebssysteme

AUFGABE 1

1. Prozess-Scheduling

a) Skizzieren Sie in einem Diagramm, welche Zustände Prozesse in einem System annehmen können und welche Zustandsübergänge existieren. Beschriften Sie die Knoten des Diagramms mit den Zustandsbezeichnungen und die Kanten des Diagramms mit den Ereignissen, die den entsprechenden Zustandsübergang auslösen.

b) Betrachten Sie die folgenden Prozesse:

Prozess	Ankunftszeit	Benötigte Rechenzeit
P0	0	8
P1	1	6
P2	2	3
P3	3	9
P4	4	4

Führen Sie das Scheduling für dieses Beispiel mit den Algorithmen *First Come First Served* und *Preemptive Shortest Job First* (auch bekannt als "*Shortest Remaining Time First*") durch.

Stellen Sie den Gesamtablauf je Algorithmus tabellarisch dar, indem Sie für jeden Zeitpunkt den jeweils aktiven Prozess, die getroffene Scheduling-Entscheidung sowie den Zustand der "Ready-Queue" *nach* der Scheduling-Entscheidung angeben. Sie dürfen Zeilen auslassen, in denen sich im Vergleich zur vorigen Zeile nichts ändert.

Gehen Sie dabei von folgender Annahme aus: Sollte das Scheduling-Kriterium keine eindeutige Scheduling-Entscheidung zwischen zwei oder mehreren Prozessen zulassen, so wird von diesen der Prozess mit der *frühesten Ankunfszeit* bevorzugt.

c) Berechnen Sie für obiges Beispiel und NUR für *Preemptive Shortest Job First* die durchschnittliche Wartezeit und den Durchsatz. Geben Sie die Durchlaufzeit von Prozess P1 an.

Fortsetzung nächste Seite!

2. Verklemmungen

a) Geben Sie (ohne weitere Erläuterung) die Bedingungen an, die erfüllt sein müssen, damit eine Verklemmung auftreten kann.

b) Eine einfache Möglichkeit zur Verhinderung von Verklemmungen oder *Deadlocks* ist der Einsatz des Zeitstempelverfahrens *wound-wait*, welches nach folgendem Prinzip arbeitet:

- Jedem Prozess P wird vom Betriebssystem ein eindeutiger Zeitstempel $TS(P)$ zugeordnet, so dass $TS(P) \neq TS(Q)$ für $P \neq Q$.
- Die Zeitstempel werden streng monoton wachsend vergeben, so dass jeder Prozess einen kleineren Zeitstempel besitzt als alle jüngeren Prozesse. Für zwei Prozesse P und Q gilt:
  P ist *älter* als Q genau dann, wenn $TS(P) < TS(Q)$.
  P ist *jünger* als Q genau dann, wenn $TS(P) > TS(Q)$.
- Wenn ein Prozess P ein Betriebsmittel anfordert, das erst von einem Prozess Q freigegeben werden müsste, wird nach folgender Strategie verfahren:
  - Ist P älter als Q, wird Q abgebrochen, so dass P weiterlaufen kann.
  - Ist P jünger als Q, wartet P auf die Freigabe des Betriebsmittels.

i) Wenden Sie das wound-wait Verfahren auf folgende Situation an:

Gegeben sind zwei Prozesse P_1 und P_2 mit Zeitstempel $TS(P_1) = 11$ und $TS(P_2) = 12$. P_1 hält Betriebsmittel A und B. P_2 hält Betriebsmittel C. Zum Zeitpunkt 20 fordert P_2 Betriebsmittel A an, das P_1 hält. Zum Zeitpunkt 21 fordert P_1 Betriebsmittel C an, das P_2 hält.

Beschreiben Sie kurz, wie sich das System zu den Zeitpunkten 20 und 21 verhält, wenn das wound-wait Verfahrens angewendet wird.

ii) Begründen Sie, warum beim wound-wait Verfahren kein Deadlock zwischen *zwei* verschiedenen Prozessen auftreten kann.

iii) Beweisen Sie für eine *beliebige* endliche Menge von n Prozessen $\{P_1, ..., P_n\}$, dass beim Einsatz des wound-wait Verfahrens kein Deadlock auftreten kann. Nehmen Sie dazu an, dass sich k Prozesse, $k \in \{2, ..., n\}$, in einem Wartezyklus befinden. Zeigen Sie im Anschluss, dass diese Situation unter Anwendung des wound-wait Verfahren nicht auftreten kann.

Fortsetzung nächste Seite!

3. Synchronisation

Synchronisieren Sie mit Hilfe von Semaphoren den Parkvorgang von Fahrzeugen, die nebenläufig in einem Parkhaus ankommen und abfahren. Das Parkhaus besitzt genau eine Einfahrt, genau eine Ausfahrt, zwei Kassenautomaten und 100 Stellplätze.

Bei der Einfahrt in das Parkhaus fordert der Fahrzeugführer ein Parkticket an. Sobald Stellplätze frei sind, öffnet sich die Einfahrtsschranke und das Fahrzeug parkt an einem freien Stellplatz.

Beim Verlassen des Parkhauses wird der Parkpreis an einem der beiden Kassenautomaten entrichtet und anschließend das Fahrzeug durch die Ausfahrt bewegt.

Der Parkvorgang soll durch die Prozesse `Ankunft` und `Abfahrt` simuliert werden, welche die kontinuierliche nebenläufige Ankunft bzw. Abfahrt von Fahrzeugen aus dem Parkhaus abbilden. Zur Synchronisation stehen Ihnen binäre (`BinarySemaphore`) und zählende Semaphore (`CountingSemaphore`) zur Verfügung.

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

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

public class CountingSemaphore implements Semaphore{
  public CountingSemaphore(int N){...}
  ...
}
```

Vervollständigen Sie die Codeschablonen auf den nächsten Seiten, indem Sie die Kommentare der Form `/* Ergänzung` $N.M$ `*/` durch Code für die Initialisierung der verwendeten Semaphore bzw. für die Synchronisation von Threads der Klassen `Ankunft` und `Abfahrt` ersetzen. Sie können an den mit `/* Ergänzung` $N.M$ `*/` gekennzeichneten Stellen beliebig viele Codezeilen oder auch keine einfügen. Schreiben Sie Ihre Lösung in folgender Form auf:

Ergänzung	Code
1.1	`new BinarySemaphore()`
1.2	...
1.3	...
...	...
3.3	...

Fortsetzung nächste Seite!

Ihre Lösung soll einen möglichst hohen Grad an Parallelität ermöglichen. Sorgen Sie dafür, dass nur dann Fahrzeuge das Parkhaus verlassen können, wenn sich welche im Parkhaus befinden. Zu Beginn der Simulation sollen 20 der 100 Stellplätze des Parkhauses bereits belegt sein.

Codeschablone für die Initialisierung:

```
// Einfahrt des Parkhauses
Semaphore einfahrt = /* Ergänzung 1.1 */;

// Ausfahrt des Parkhauses
Semaphore ausfahrt = /* Ergänzung 1.2 */;

// Kassenautomaten
Semaphore kassenautomat = /* Ergänzung 1.3 */;

// freie Stellplaetze des Parkhauses
Semaphore freierStellplatz = /* Ergänzung 1.4 */;

// belegte Stellplaetze des Parkhauses
Semaphore belegterStellplatz = /* Ergänzung 1.5 */;
```

Codeschablone für die Klasse `Ankunft`:

```
public class Ankunft extends Thread {
  public void run() {

    while(true) {

    /* Ergänzung 2.1 */

    // Parkticket an der Parkhauseinfahrt entnehmen

    /* Ergänzung 2.2 */

    // In das Parkhaus einfahren

    /* Ergänzung 2.3 */

    // Parken

    /* Ergänzung 2.4 */

    }
  }
}
```

Fortsetzung nächste Seite!

Codeschablone für die Klasse `Abfahrt` (10 Punkte):

```
public class Abfahrt extends Thread {
  public void run() {

    while(true) {

    /* Ergänzung 3.1 */

    // Parkpreis am Automaten entrichten

    /* Ergänzung 3.2 */

    // Aus dem Parkhaus ausfahren

    /* Ergänzung 3.3 */
    }
  }
}
```

AUFGABE 2

1. Prozesse

a) Welche vier Prozesszustände gibt es? Erklären Sie die Prozesszustände jeweils in einem Satz!

b) Die zwei folgenden Programme A und B laufen parallel auf einer Einprozessormaschine mit RoundRobin-Scheduling-Strategie. Gehen Sie davon aus, dass die Abarbeitung einer nummerierten Programmzeile immer genau eine ganze Zeitscheibe in Anspruch nimmt. Nehmen Sie die Eingabe `a.readData()` als blockierend an. Bei Deblockierung wird der Befehl wiederholt. Gehen Sie außerdem davon aus, dass zu Beginn die Variable `count=0` ist! Etwa in der Mitte von Zeitscheibe 6 kommt einmalig ein Datum für A.

```
Program A:
while (true) {                (A1)
  a.readData();               (A2)
  array[count]=a;             (A3)
  count=count+1;              (A4)
}
```

```
Program B:
while (true) {                    (B1)
  if (count>0) {                  (B2)
    count=count-1;                (B3)
    array[count].print();         (B4)
  }
}
```

Schreiben Sie die Abarbeitungsreihenfolge als Folge von Programmzeilennummern (A1, B1, ...) beginnend mit Programm A in Zeitscheibe 1 bis zur Zeitscheibe 20 nieder!

Muster:

Zeit	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20***
	A1	B1	...																	

Was sind die Prozesszustände der beiden Prozesse jeweils kurz vor Ende der Zeitscheiben 4, 7 und 10?

Fortsetzung nächste Seite!

2. Schedulingstrategien

a) Definieren Sie *Fairness* für eine Schedulingstrategie aus Sicht eines Benutzers! Was ist *nicht-preemptives Scheduling*? Inwieweit ist nicht-preemptives Scheduling fair?

b) Gegeben sind folgende Prozesse mit ihren Bereitzeitpunkten und ihrem Rechenzeitbedarf. Führen Sie für eine Einprozessormaschine das nicht-preemptive Scheduling gemäß Shortest Processing Time First (SPF) und das preemptive Scheduling gemäß Shortest Remaining Time First (SRTF) mit Zeitscheibenlänge 1 durch.

Prozess	***Bereitzeit***	***Rechenzeit***
A	0	5
B	0	4
C	1	2
D	4	1
E	6	2

Zeichnen Sie die entsprechenden Gantt-Diagramme und berechnen Sie jeweils die mittlere Verweilzeit!

Muster:

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***Mittlere Verweilzeit***
SPF	...													
SRTF	...													

Welche Gefahr bergen diese beiden Strategien SPF und SRTF generell in Bezug auf Prozesse mit hohem Rechenzeitbedarf?

Nennen Sie je eine preemptive und eine nicht-preemptive Schedulingstrategie, die diesen Nachteil grundsätzlich nicht haben! Begründen Sie Ihre Antworten!

c) Gegeben sind folgende periodische, unterbrechbare Prozesse A, B und C auf einer Einprozessormaschine. A benötigt alle 15 Zeiteinheiten eine Zeiteinheit Rechenzeit, B benötigt alle 5 Zeiteinheiten 3 Zeiteinheiten Rechenzeit und C benötigt alle 6 Zeiteinheiten 2 Zeiteinheiten Rechenzeit.

Berechnen Sie die Prozessorauslastung!

Interpretieren Sie die periodischen Prozesse jeweils als Folge von einzelnen Prozessen und führen Sie das Scheduling gemäß preemptiven Shortest Remaining Time First mit Zeitscheibenlänge 1 durch und zeichnen Sie das entsprechende Gantt-Diagramm. Bei gleichen Fristen wird A gegenüber B und C bevorzugt und B wird gegenüber C bevorzugt.

Muster:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A																														
B																														
C																														

Fortsetzung nächste Seite!

3. Synchronisation

a) Nennen Sie die vier notwendigen und hinreichenden Bedingungen für Verklemmungen und jeweils eine passende Maßnahme um Verklemmungen auszuschließen!

b) Betrachten Sie die folgenden Prozeduren `P(s)` und `V(s)`!

```
void P(int s)
{
   while (s<=0) noop;
   s=s-1;
}
```

```
void V(int s)
{
   s=s+1
}
```

Welche Grundannahme ist gefordert, damit diese beiden Prozeduren zu einer Implementierung der Semaphoroperationen `P(s)` und `V(s)` für ein Semaphor `s` werden und wie kann man diese Annahme erfüllen?

Welche zwei Nachteile hat die hier gezeigte Implementierungsvariante für Semaphore?

Geben Sie eine Implementierungsvariante für `P(s)` und `V(s)` ohne diese Nachteile an! Geben Sie explizit an, welche Datenstrukturen für `s` benötigt werden und wozu!

Fortsetzung nächste Seite!

4. Virtueller Speicher

a) Wie viel Speicher wird bei einer Seitengröße von p Bytes in einem System „verschwendet", in dem jeder Prozess durchschnittlich s Bytes belegt und ein Eintrag in der Seitentabelle e Bytes Platz verbraucht? Begründen Sie Ihre Antwort!

b) Gegeben sei ein Rechner mit 5 Bit physikalischem und virtuellem Adressraum und 5 Bit breiten Worten. Der Inhalt des physikalischen Speichers ist unten angegeben. Es gelten folgenden Annahmen:

- Die Seiten sind jeweils 4 Worte groß.
- Die Basisadresse der Seitentabelle ist 00000.
- Ein Seitentabelleneintrag besteht aus 5 Bit, die ersten (=höherwertigsten) 3 Bits geben die physikalische Seitenrahmennummer, das 4. Bit sei das present- bzw. mapped-Bit und das 5. Bit sei hier nicht von Bedeutung.
- Bei der zweistufigen Adressierung werden 2 Bit zur Indizierung der Seitentabellen der zweiten Stufe verwendet.

Adresse	*Inhalt*	*Adresse*	*Inhalt*	*Adresse*	*Inhalt*	*Adresse*	*Inhalt*
00000	10111	01000	01110	10000	11111	11000	00110
00001	10011	01001	00011	10001	11010	11001	01010
00010	11011	01010	10010	10010	01011	11010	00010
00011	11101	01011	00000	10011	01101	11011	10001
00100	01000	01100	11000	10100	00101	11100	10100
00101	00100	01101	10000	10101	11100	11101	01001
00110	01111	01110	11110	10110	00111	11110	10110
00111	11001	01111	01100	10111	10101	11111	00001

Welche Daten stehen an der Adresse `11001`, wenn Sie diese Adresse als physikalische Adressen, als 1-stufige und als 2-stufige virtuelle Adresse verwenden?

Was passiert wenn Sie auf die einstufige virtuelle Adresse `11101` zugreifen?

Fortsetzung nächste Seite!

5. Seitenverwaltung

a) Beschreiben Sie der Reihe nach, welche Schritte zur Behebung eines Seitenfehlers nötig sind!

b) Zu jedem Zeitpunkt der „Abarbeitung" einer Referenzkette durch eine hier nicht näher spezifizierte Verdrängungsstrategie seien die Seiten in einem als Keller organisierten Speicher gegeben. Bestimmen Sie die entsprechende Distanzkette!

1	A	A	B	C	B	D	E	F	G	D	H	H	A	H	B	F	H	D	A	F	F	E	B	E
2			A	B	C	B	D	E	F	G	D	D	H	A	H	B	F	H	E	A	A	F	E	B
3				A	A	C	B	D	E	F	G	G	D	D	A	H	B	F	H	E	E	A	F	F
4						A	C	B	D	E	F	F	G	G	D	A	A	B	F	H	H	H	A	A
5							A	C	B	B	E	E	F	F	G	D	D	A	B	B	B	B	H	H
6								A	C	C	B	B	E	E	F	G	G	G	G	G	G	G	G	G
7									A	A	C	C	B	B	E	E	E	E	E	E	E	E	E	E
8											A	A	C	C	C	C	C	C	C	C	C	C	C	C