
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2016**

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: 20

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 ausgewählten Teilaufgaben anzugeben (z. B. A2, B1)!

Bitte wenden!

**Themenschwerpunkt A
(Datenbanksysteme)**

Teilaufgabe 1:**1. Modellierung**

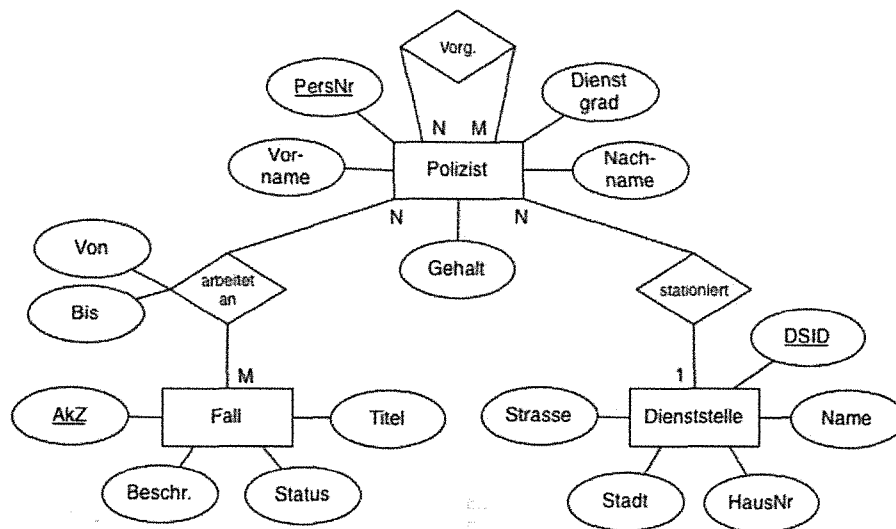
Für die bayerische Forstverwaltung wird eine Datenbank zur Erschließung einer Jagd-Statistik benötigt. Gehen Sie dabei von folgendem Szenario aus:

- Die Administration von Jagdgebieten obliegt den Landkreisen. Jeder Landkreis besitzt, neben seinem Namen und der Einwohnerzahl, ein eindeutiges KFZ-Kennzeichen.
 - Die Jagd findet in Jagdgebieten statt. Ein Jagdgebiet soll dem Landkreis zugeteilt werden, indem es liegt. Gehen Sie davon aus, dass Jagdgebiete nicht in mehreren Landkreisen liegen können. Zusätzlich ist für jedes Jagdgebiet der Name und die Gesamtfläche zu speichern. Dabei ist zu beachten, dass die Namen nur innerhalb eines einzelnen Landkreises eindeutig sind.
 - Die Erlaubnis zum Jagen wird durch einen Jagdschein erteilt. Dieser kann nur von einem Landkreis ausgestellt werden und beschränkt sich auf ein oder mehrere Jagdgebiete. Er wird durch eine Jagdschein-Nummer (JSNR) identifiziert und ist in einem bestimmtem Zeitintervall gültig. Dieses soll über zwei Zeitpunkte festgelegt werden (gültig von, gültig bis).
 - Ein Jäger besitzt genau einen Jagdschein. Zu einem Jäger sollen Name, Stadt, Straße und Hausnummer, gespeichert werden. Da die Jagdtradition innerhalb einer Familie häufig von einer zur nächsten Generation weitergegeben wird, kann es vorkommen, dass Name und Adresse von zwei unterschiedlichen Jägern gleich ist (z.B. Vater und Sohn). Aus diesem Grund ist eine eindeutige Identifikationsnummer notwendig.
 - Um Statistiken erheben zu können, muss berücksichtigt werden, welches Wild von welchen Jägern zu welchem Zeitpunkt in welchem Jagdgebiet erlegt worden ist. Gehen Sie davon aus, dass es mehrere Jäger geben kann, die gemeinsam ein Wild erlegen (z.B. in einer Jagdgesellschaft). Zu einem Wild gehört die Art (z.B. Reh), die Größe, das Gewicht, sowie eine eindeutige Identifikationsnummer. Zusätzlich unterscheidet man zwischen Haarwild und Federwild, wobei beim Haarwild der Typ des Gehörns (z.B. Hirschgeweih) und beim Federwild die Flügelspannweite betrachtet werden soll.
- a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
 - die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen, und
 - die Funktionalitäten der Relationship-Typen.
- b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an. Achten Sie dabei insbesondere darauf, dass die Relationenschemata keine redundanten Attribute enthalten.

Fortsetzung nächste Seite!

2. Relationale Algebra, SQL

Gegeben sei folgendes ER-Modell, welches Polizisten, deren Dienststelle und Fälle, an denen sie arbeiten, speichert:



Gehen Sie dabei von dem dazugehörigen relationalen Schema aus:

Polizist: {[PersNr, DSID, Vorname, Nachname, Dienstgrad, Gehalt]}

Dienststelle: {[DSID, Name, Strasse, HausNr, Stadt]}

Fall: {[AkZ, Titel, Beschreibung, Status]}

Arbeitet An: {[PersNr, AkZ, Von, Bis]}

Vorgesetzte: {[PersNr, PersNr_Vorgesetzter]}

- Formulieren Sie eine Anfrage in relationaler Algebra, welche den Vornamen und Nachnamen von Polizisten zurückgibt, deren Dienstgrad 'Polizeikommissar' ist und mehr als 1500 Euro verdienen.
- Formulieren Sie eine Anfrage in relationaler Algebra, welche die Titel der Fälle ausgibt, die von Polizisten mit dem Nachnamen 'Mayer' bearbeitet wurden.
- Formulieren Sie eine SQL-Anfrage, welche die Anzahl der Polizisten ausgibt, die in der Stadt 'München' arbeiten und mit Nachnamen 'Schmidt' heißen.
- Formulieren Sie eine SQL-Anfrage, welche die Namen der Dienststellen ausgibt, die am 14.02.2012 an dem Fall mit dem Aktenzeichen XZ1508 beteiligt waren. Ordnen Sie die Ergebnismenge alphabetisch (aufsteigend) und achten Sie darauf, dass keine Duplikate enthalten sind.
- Definieren Sie die View 'Erstrebenswerte_Dienstgrade', welche Dienstgrade enthalten soll, die in München mit durchschnittlich mehr als 2500 Euro besoldet werden.

Fortsetzung nächste Seite!

- f) Formulieren Sie eine SQL-Anfrage, welche Vorname, Nachname und Dienstgrad von Polizisten mit Vorname, Nachname und Dienstgrad ihrer Vorgesetzten als ein Ergebnis-Tupel ausgibt (siehe Beispiel-Tabelle). Dabei sind nur Polizisten zu selektieren, die an Fällen gearbeitet haben, deren Titel den Ausdruck 'Fussball' beinhalten. An Vorgesetzte sind keine Bedingungen gebunden. Achten Sie darauf, dass Sie nicht nur direkte Vorgesetzte, sondern alle Vorgesetzte innerhalb der Vorgesetzten-Hierarchie betrachten. Ordnen Sie ihre Ergebnismenge alphabetisch (absteigend) nach Nachnamen des Polizisten.

VN	NN	DG	VN_VG	NN_VG	DG_VG
Hans	Müller	Polizeimeister	Andreas	Schmidt	Polizeikommissar
Hans	Müller	Polizeimeister	Stefan	Hoffmann	Polizeidirektor
Josef	Fischer	Polizeihauptmeister	Sebastian	Wagner	Polizeioberkommissar
Josef	Fischer	Polizeihauptmeister	Stefan	Hoffmann	Polizeidirektor

Abbildung 1: Ergebnistabelle

Hinweis: Sie dürfen Views verwenden, um Teilergebnisse auszudrücken.

Fortsetzung nächste Seite!

3. Normalformen

Gegeben sei ein Auszug der Relation S einer relationalen Datenbank in Abbildung 2. Darin sind Informationen zu Schrauben eines Baumarkts enthalten. Jeder Schraubentyp besitzt eine eindeutige Artikelnummer, sowie eine Kopfform, ein Formelement, einen Gewindetyp, das Material und die jeweilige Länge. Zusätzlich ist zu jeder Schraube gespeichert, welche DIN Norm sie erfüllt. Diese Normen standardisieren Kopfform, Formelement, Material und Gewinde. Zum Beispiel erfüllt eine Senk-Holzschraube mit Schlitz die DIN Norm 97.

ArtNr	DIN	Kopfform	Formelement	Gewinde	Material	Länge
SR2141	DIN571	Sechskant	Kreuz	Holzschraube	Holz	34mm
SR8923	DIN478	Vierkant	Trox-TR	Rechts	Stahl	41mm
SR0984	DIN7996	Rund	Schlitz	Links	Messing	80mm
SR7688	DIN571	Sechskant	Kreuz	Holzschraube	Holz	52mm
SR3579	DIN965	Senk	Kreuz	Rohr	Keramik	29mm
SR4139	DIN571	Sechskant	Kreuz	Holzschraube	Holz	78mm
SR6760	DIN97	Senk	Schlitz	Holzschraube	Holz	29mm

Abbildung 2: Relation S

Bearbeiten Sie mit diesen Informationen folgende Teilaufgaben. Vergessen Sie dabei nicht, Ihr Vorgehen stichpunktartig zu dokumentieren und zu begründen.

- Bestimmen Sie alle funktionalen Abhängigkeiten von S.
- Bestimmen Sie alle Kandidatenschlüssel von S.
- Zeigen Sie an einer geeigneten Instanz, dass in der Relation S eine Update-Anomalie auftreten kann.
Hinweis: Beachten Sie, dass Ihre Instanz alle Funktionalen Abhängigkeiten aus der Aufgabenstellung erfüllt.
- Welcher Normalform genügt die Relation S (1NF, 2NF, 3NF, BCNF)? Begründen Sie Ihre Aussagen ausführlich.

Fortsetzung nächste Seite!

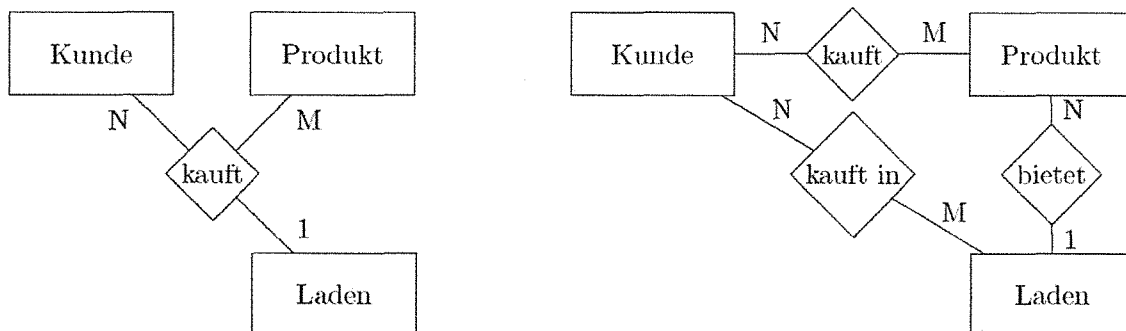
Teilaufgabe 2**1. ER-Diagramme**

- 1.1 Modellieren Sie den Aufbau eines Sportklubs mit einem ER-Diagramm. Kardinalitäten sind hierbei in der (min, max)-Notation anzugeben.

Personen unterteilen sich in Sportler und Betreuer. Personen haben einen Namen, eine Adresse, eine Telefonnummer sowie ein Geburtsdatum. Identifiziert werden sie über eine eindeutige Kennung. Zu den Sportlern ist ihr Gewicht sowie ihr Geburtsland bekannt, zu den Betreuern ihr Gehalt sowie eine Liste ihrer Führerscheinklassen.

Sportler können Mitglied eines oder mehrerer Teams sein, die über ihren Namen identifiziert werden. Jedes Team wird von mindestens einem Betreuer in mindestens einer Sportart betreut. Sportarten haben eine Kennung, einen Namen und eine Angabe zu den durchschnittlichen monatlichen Kosten. Sportler können mehrere Sportarten betreiben, wobei sie maximal eine Lieblingssportart haben.

- 1.2 Welche Aussagen lassen sich für folgende Modelle unter einer üblichen Interpretation treffen? Inwieweit unterscheiden bzw. gleichen sie sich? Es kommt hier vor allem auf die Gesamtaussage an. Formulieren Sie die Aussagen auch mit Hilfe partieller Funktionen.

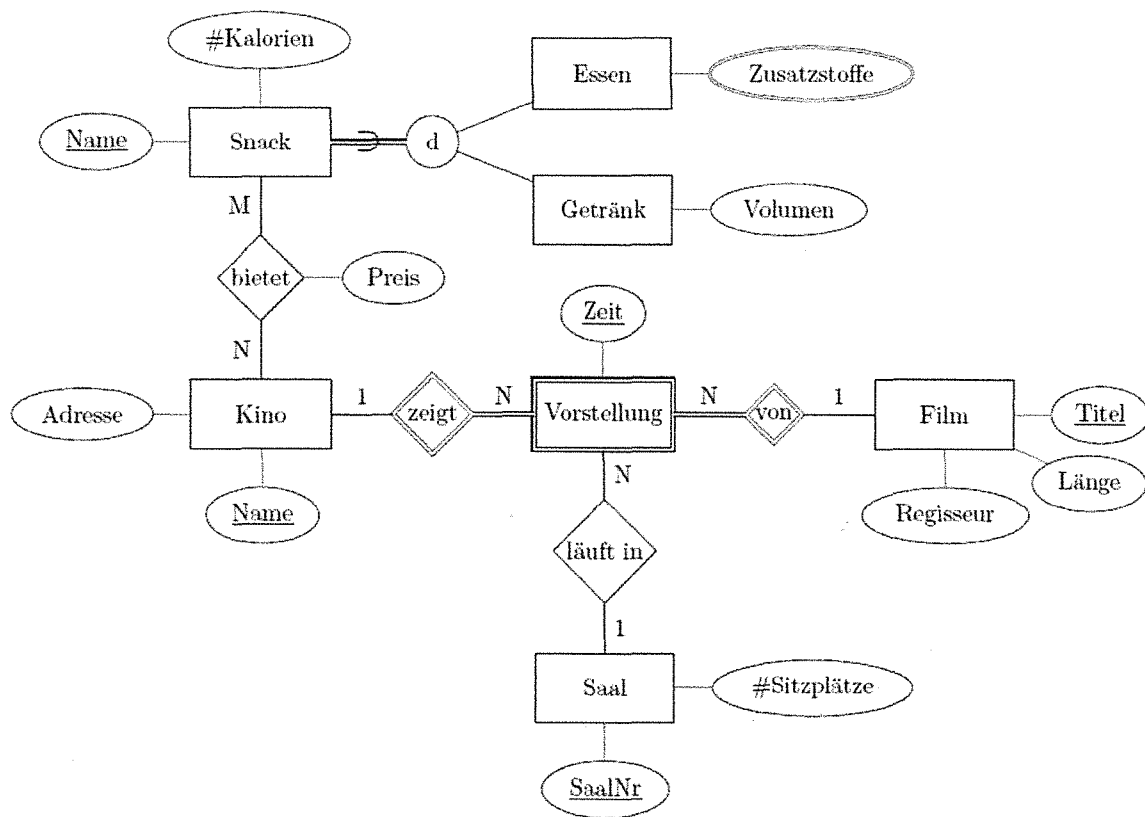


Bemerkung: Die Attribute wurden der Übersichtlichkeit wegen weggelassen.

Fortsetzung nächste Seite!

2. Relationales Modell

2.1 Überführen Sie das folgende ER-Modell in ein relationales Modell.



2.2 Macht es im ER-Modell von Aufgabe 2.1 einen Unterschied, zwischen Entitätstyp „Kino“ und Entitätstyp „Saal“ noch eine zusätzliche Beziehung einzuführen? Wenn ja, wo liegt der Unterschied? Wenn nein, warum nicht?

Fortsetzung nächste Seite!

3. Normalisierung

Gegeben ist eine Relation „Mitarbeiter“ sowie alle für die Attribute der Relation geltenden funktionalen Abhängigkeiten.

- 3.1 Ist die Relation in erster Normalform? Ist sie in zweiter Normalform? Ist sie in dritter Normalform? Geben Sie jeweils eine Begründung für Ihre Antwort mit an.

Mitarbeiter(MID, Vorname, Nachname, Geburtsdatum, Straße, Hausnummer, Postleitzahl, Ort, PrivatVorwahl, PrivatTelefonnummer, Abteilungsnummer, Abteilungsname, Bereichsnummer, Bereichsname, Eintrittsdatum, Dienstalter, Gehaltsklasse, Gehalt)

Zusätzlich zur Schlüsseleigenschaft des Attributs „MID“ ($MID \rightarrow \text{Mitarbeiter}$) gelten noch weitere funktionalen Abhängigkeiten. Alle Abhängigkeiten sind in der Menge FD zusammengefasst:

FD={
MID \rightarrow Vorname Nachname Geburtsdatum Straße Hausnummer Postleitzahl Ort PrivatVorwahl
PrivatTelefonnummer Abteilungsnummer Abteilungsname Bereichsnummer Bereichsname
Eintrittsdatum Dienstalter Gehaltsklasse Gehalt,
Postleitzahl \rightarrow Ort,
Postleitzahl \rightarrow PrivatVorwahl,
PrivatVorwahl \rightarrow Ort,
Ort \rightarrow Privatvorwahl,
Abteilungsnummer \rightarrow Abteilungsname,
Abteilungsname \rightarrow Abteilungsnummer,
Bereichsnummer \rightarrow Bereichsname,
Bereichsname \rightarrow Bereichsnummer,
Bereichsnummer \rightarrow Abteilungsnummer,
Eintrittsdatum \rightarrow Dienstalter,
Dienstalter \rightarrow Eintrittsdatum,
Eintrittsdatum Gehaltsklasse \rightarrow Gehalt
}

Bei der Bestimmung der kanonischen Überdeckung für die Menge FD muss eine Links- und eine Rechtsreduktion durchgeführt werden.

- 3.2 Warum führt eine Linksreduktion von FD zu keiner Veränderung?

Fortsetzung nächste Seite!

3.3 Welche Attribute auf der rechten Seite der funktionalen Abhängigkeit

MID → Vorname Nachname Geburtsdatum Straße Hausnummer Postleitzahl Ort
PrivatVorwahl PrivatTelefonnummer Abteilungsnummer Abteilungsname
Bereichsnummer Bereichsname Eintrittsdatum Dienstalter Gehaltsklasse
Gehalt

sind redundant, wenn noch keine der anderen Abhängigkeiten entfernt wurden?

4. Relationale Algebra und SQL

Nachfolgend ist das Datenbankschema einer Schule gegeben. Verschiedene Anfragen in SQL sollen auf diesem Schema durchgeführt werden.

Lehrkraft (Personalnummer, Name, Geschlecht, Wohnort, Geburtsjahr)

Schüler (Schülernummer, Name, Konfession, Klassennummer)

Fach (Fachname, Beschreibung)

unterrichtet (Personalnummer, Schülernummer, Fachname, Stundenzahl)

Bemerkung: Das Geschlecht einer Lehrkraft ist entweder 'm' oder 'w'.

In unterrichtet: Personalnummer ist Fremdschlüssel und referenziert den Primärschlüssel von Lehrkraft; Schülernummer ist Fremdschlüssel und referenziert den Primärschlüssel von Schüler; Fachname ist Fremdschlüssel und referenziert den Primärschlüssel von Fach.

- 4.1 Es soll eine Liste mit Schülernamen und den zugehörigen Klassen ausgegeben werden, wobei die Ausgabe absteigend nach Klassennummer und innerhalb der Klassen aufsteigend nach Name erfolgen soll.
 - 4.2 Aus welchen Wohnorten stammen die Lehrerinnen und wie viele kommen jeweils aus einem Ort?
 - 4.3 Wie lautet das durchschnittliche Geburtsjahr aller Lehrkräfte?
 - 4.4 Zu welchem Fach gibt es momentan nur eine unterrichtende Lehrkraft?
 - 4.5 Welche Lehrkräfte unterrichten mindestens zwei verschiedene Klassen im Fach Mathematik? Geben Sie Personalnummer und Name der entsprechenden Lehrkräfte aus.
- Nachfolgende Anfrage soll in relationaler Algebra formuliert werden.
- 4.6 Geben Sie alle Schüler mit Nummer aus, die von Lehrkraft Maier unterrichtet werden.

5. Transaktionen

- 5.1 Welche vier grundlegenden Eigenschaften soll eine Datenbanktransaktion erfüllen? Nennen Sie diese vier Eigenschaften und beschreiben Sie sie kurz.
- 5.2 Geben Sie zwei Fehler an, die im unkontrollierten (nicht synchronisierten) Mehrbenutzerbetrieb auftreten können und beschreiben Sie diese jeweils anhand eines Beispiels.

**Themenschwerpunkt B
(Betriebssysteme)**

Teilaufgabe 1

Aufgabe 1:

Speicherverwaltungstechniken

- a) Bei der dynamischen Speicherzuteilung muss der freie Speicher verwaltet werden. Im Folgenden sind vier verschiedene Techniken aufgelistet, die in unterschiedlichen Speicherverwaltungsverfahren eingesetzt werden. Beschreiben Sie jeweils kurz diese Techniken, ihre Eigenschaften sowie ihre Vor- und Nachteile.
- Freispeicherverwaltung mit einer Bitliste
 - Freispeicherverwaltung mit einer separat liegenden verketteten Liste
 - Freispeicherverwaltung mit der Verkettung von freien Blöcken
 - Freispeicherverwaltung mit dem Buddy-Verfahren
- b) Bei der Suche nach freien Speicherbereichen innerhalb einer Freispeicherverwaltung, die mit Verkettung arbeitet, gibt es vier unterschiedliche Vergabestrategien. Benennen Sie diese Verfahren und beschreiben Sie ihre Arbeitsweise. Geben Sie in O-Notation an, wie aufwändig jeweils die Suche nach einem freien Block ausreichender Größe, das Einsortieren eines Blocks in die Freispeicherliste und das Verschmelzen benachbarter Freispeicherblöcke sind. Wie verhalten sich die unterschiedlichen Verfahren in Bezug auf die Fragmentierung des Speichers?

Aufgabe 2:

Scheduling

- a) Geben Sie einen Überblick über drei Ihnen bekannte Schedulingstrategien. Beschreiben Sie jeweils die Funktionsweise sowie die Eigenschaften (Vor-/Nachteile und sonstige Besonderheiten).
- b) Welche Strategie eignet sich besonders gut für ein Mehrbenutzer-Betriebssystem, in dem gleichzeitig mehrere interaktive Anwendungen und sehr rechenzeitintensive Hintergrundaufgaben zu bearbeiten sind?
Die Gesamteffizienz der Strategie hängt sicherlich von der Wahl geeigneter Parameter ab. Welche Parameter sind bei der von Ihnen gewählten Strategie einstellbar und wie wirkt sich welche Art der Einstellung auf das Systemverhalten aus?

Fortsetzung nächste Seite!

Aufgabe 3:

Virtueller Speicher

- a) Bei Virtuellem Speicher kann es zu Seitenfehlern (Page-Faults) kommen. In welchen Situationen tritt ein Seitenfehler auf und was muss das Betriebssystem jeweils zur Behebung tun (beschreiben Sie drei solche Situationen)?
- b) Falls eine neue Seite allokiert werden soll, aber keine freie Seitenkachel im Hauptspeicher mehr verfügbar ist, muss eine Seite ausgelagert werden. Wie würde die optimale Seitenersetzungsstrategie funktionieren und warum ist sie in der Praxis nicht realisierbar?
- c) Eine mögliche Approximation der optimalen Strategie wäre LRU. Beschreiben Sie die Funktionsweise von LRU. Wie könnte man LRU implementieren? Ist LRU eine in der Praxis gut oder problematisch implementierbare Strategie? Warum?
- d) Clock (oder Second Chance) ist eine weitere Strategie, die eine möglichst optimale Seitenersetzung anstrebt. Beschreiben Sie die grundsätzliche Funktionsweise dieser Strategie. Hat sie gegenüber LRU Vor- bzw. Nachteile? Warum?

Fortsetzung nächste Seite!

Aufgabe 4:

Dateisysteme

- a) Was versteht man unter einem Inode in einem Unix/Linux-Dateisystem? Geben Sie vier Beispiele für Informationen, die darin gespeichert sind.
- b) Wie werden Dateiattribute und Dateiinhalte im Gegensatz zu Linux in einem Windows-NT-Dateisystem (NTFS) verwaltet?
- c) Gegeben ist die folgende Ausgabe des Kommandos `ls -alRi /tmp/sp` (rekursiv absteigende Ausgabe aller Dateien und Verzeichnisse unter `/tmp/sp` mit Angabe der Inode-Nummer) auf einem Linux-System:

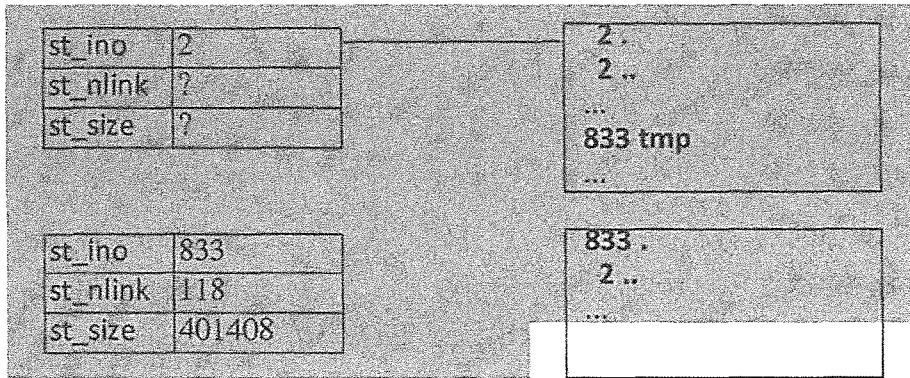
```
fauixx> ls -alRi /tmp/sp
/tmp/sp:
total 408
91 drwxrwxr-x 4 jklein i4staff 4096 Jul 19 15:17 .
833 drwxrwxrwt 118 root root 401408 Jul 19 15:30 ..
96 drwxrwxr-x 2 jklein i4staff 4096 Jul 19 15:21 dir1
98 drwxrwxr-x 2 jklein i4staff 4096 Jul 19 15:22 dir2

/tmp/sp/dir1:
total 68
96 drwxrwxr-x 2 jklein i4staff 4096 Jul 19 15:21 .
91 drwxrwxr-x 4 jklein i4staff 4096 Jul 19 15:17 ..
536 -rwxr--r-- 2 jklein i4staff 52224 Jul 19 15:20 datei1
258 -rw-rw-r-- 1 jklein i4staff 30 Jul 19 15:21 datei2

/tmp/sp/dir2:
total 68
98 drwxrwxr-x 2 jklein i4staff 4096 Jul 19 15:22 .
91 drwxrwxr-x 4 jklein i4staff 4096 Jul 19 15:17 ..
896 lrwxrwxrwx 1 jklein i4staff 14 Jul 19 15:22 dat1 -> ../dir1/datei2
536 -rwxr--r-- 2 jklein i4staff 52224 Jul 19 15:20 xxx
900 -rw-rw-r-- 1 jens i4staff 30 Jul 19 15:22 yyy
```

Ergänzen Sie im weißen Bereich die auf der folgenden Seite im grauen Bereich bereits angefangene Skizze der Inodes und Datenblöcke des Linux-Dateisystems um alle entsprechenden Informationen, die Sie aus obiger Ausgabe entnehmen können.

Fortsetzung nächste Seite!

Inodes**Inhalt der Datenblöcke**

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

st_ino	
st_nlink	
st_size	

- d) Beschreiben Sie kurz (in Stichworten) die grundsätzliche Funktionsweise eines *Journaling-File-Systems*.

Teilaufgabe 2

Aufgabe 1: Grundlagen zu Betriebssystemen

Beantworten Sie folgende allgemeine Fragen aus dem Bereich der Threads:

- a) Erläutern Sie 2 wesentliche Unterschiede zwischen Threads und Prozessen.
- b) In welchen 3 grundsätzlichen Zuständen kann sich ein Thread befinden?
- c) Es gibt 2 wesentliche Arten von Threads. Nennen Sie beide Arten und geben Sie jeweils einen Vorteil an.
- d) Was versteht man unter Multithreading?
- e) Nennen Sie 3 Bestandteile, die jedem Thread innerhalb eines Prozesses zugeordnet werden.

Beantworten Sie folgende allgemeine Fragen aus dem Bereich der Zustands-Prozessmodelle:

- a) Benennen Sie die 5 grundsätzlichen Zustände des 5-Zustands-Prozessmodells.
- b) Das 7-Zustandsmodell ist eine Erweiterung des 5-Zustandsmodells. Nennen Sie nur die zusätzlichen Zustände im Vergleich zum 5-Zustandsmodell und geben Sie jeweils den Grund dafür an, warum ein solcher zusätzlicher Zustand in einem Betriebssystem gebraucht wird.
- c) Welcher Zustandsübergang wird durch den Dispatcher realisiert?
- d) Müsste man das Zustands-Prozessmodell anpassen, wenn man es auf eine Multiprozessorumgebung anwenden will? Begründen Sie kurz Ihre Antwort.

Fortsetzung nächste Seite!

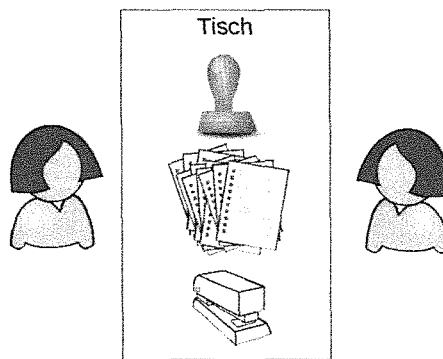
Aufgabe 2: Petrinetz und Erreichbarkeitsgraphen

Gegeben Sei folgendes Szenario, welches als Petrinetz modelliert werden soll:

2 Sekretärinnen eines Instituts müssen einen riesigen Stapel an Unterlagen bearbeiten. Dazu benötigen sie gleichzeitig 2 Gegenstände, um zusammengehörende Unterlagen aus dem Gesamtstapel zu verarbeiten. Die beiden Gegenstände sind:

- Der offizielle Institutsstempel
- Der Tacker

Beide Sekretärinnen sitzen zusammen an einem Tisch gegenüber. Auf dem Tisch liegt in der Mitte der Stapel an Unterlagen, sowie der Stempel und der Tacker, wie in folgender Abbildung gezeigt:



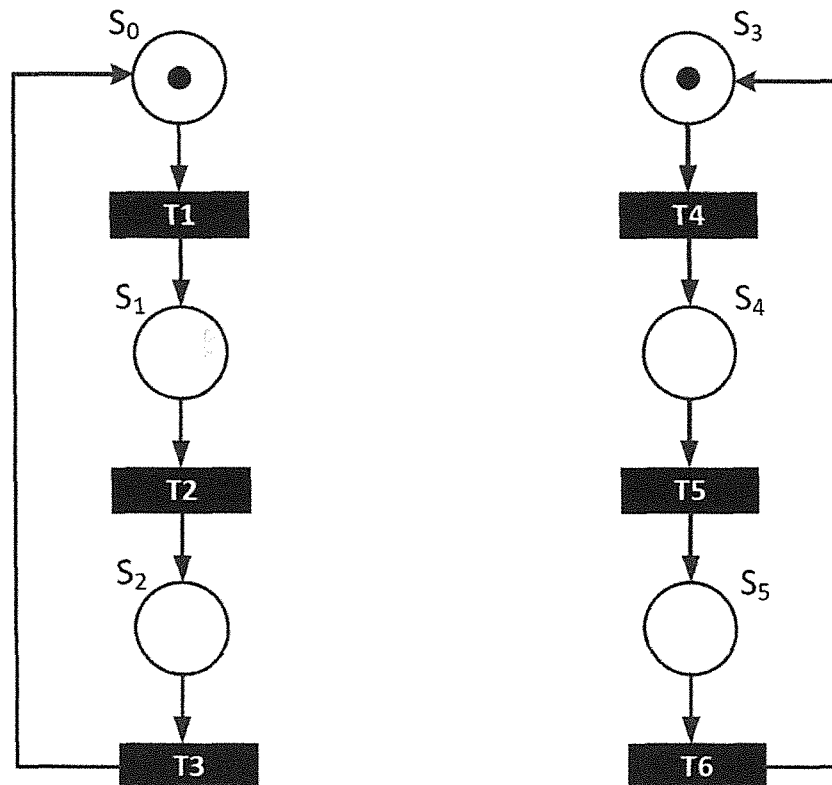
In diesem Szenario sind beide Sekretärinnen Rechtshänder. So greift jede zuerst nach dem Gegenstand, der rechts von ihr liegt, und dann zu dem anderen Gegenstand. Die eine greift somit immer zuerst nach dem Tacker und dann zum Stempel und die andere genau anders herum. Beachten Sie bitte, dass bei diesem Szenario eine Sekretärin immer beide Gegenstände gleichzeitig benutzen muss, damit sie einen Teil der Unterlagen verarbeiten kann. Nachdem eine Sekretärin die zusammengehörenden Unterlagen verarbeitet hat (getackert und gestempelt), legt sie beide Gegenstände wieder genau so auf den Tisch wie zuvor und ruht einen kurzen Moment. Eine Sekretärin hat somit drei Zustände:

- Ruhen
- Hat einen Gegenstand
- Verarbeitet Unterlagen.

Fortsetzung nächste Seite!

Bearbeiten Sie zu diesem Szenario die folgenden Teilaufgaben:

- a) Übertragen Sie das folgende Petrinetz auf Ihr Blatt und ergänzen Sie es durch Einfügen einer minimalen Anzahl von Stellen, Transitionen, Kanten oder Marken, so dass das oben beschriebene Szenario korrekt modelliert wird. Beschriften Sie außerdem alle neu eingezeichneten Stellen bzw. Transitionen. Geben Sie zusätzlich für alle Stellen und Transitionen in dem kompletten Petrinetz eine kurze Beschreibung an:



- b) Erstellen Sie den Erreichbarkeitsgraphen zu dem modellierten Petrinetz. Geben Sie hierbei auch an, wie in den Markierungen des Erreichbarkeitsgraphen die Stellen angeordnet sind. Beschriften Sie alle Übergänge zwischen Markierungen mit der Bezeichnung der Transition, die hierfür feuern muss.
- c) Begründen Sie mittels des Erreichbarkeitsgraphen jeweils separat, ob in dem angegebenen Szenario ein Deadlock bzw. ein partieller Deadlock auftreten kann.

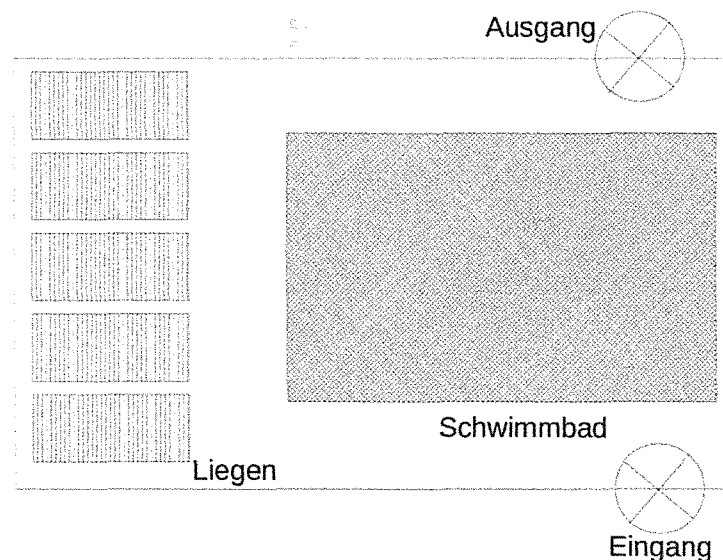
Fortsetzung nächste Seite!

Aufgabe 3: Koordination von Threads

In dieser Aufgabe sollen Sie Badegäste eines kleinen Schwimmbads synchronisieren. Die Badegäste sollen dabei als Java-Threads realisiert werden. Betrachten Sie dazu folgendes Szenario:

- Das kleine Schwimmbad besitzt maximal *maxLiegen*, die von den Badegästen genutzt werden können.
- Die Badegäste können das Schwimmbad über ein Drehkreuz betreten und über ein anderes Drehkreuz verlassen.
- Zu jedem Zeitpunkt kann immer nur eine Person durch das jeweilige Drehkreuz gehen.
- Es dürfen sich stets auch nur maximal so viele Personen im Schwimmbad befinden (inklusive der beiden Drehtüren), wie Liegen vorhanden sind.

Die nachfolgende Abbildung zeigt den schematischen Aufbau des beschriebenen Szenarios für *maxLiegen=5* und *anzahlGaeste=0*:



Übertragen Sie den folgenden Coderahmen der Klasse *Schwimmbad* und vervollständigen Sie ihn. Die Beispielimplementierungen der Klassen *Badegast* und *Simulation* sollen Ihnen verdeutlichen, wie die Klasse *Schwimmbad* verwendet werden kann.

- a) Überlegen Sie sich die notwendigen Attribute der Klasse *Schwimmbad* und implementieren Sie den Konstruktor der Klasse an der entsprechenden Stelle, die im Coderahmen dafür vorgesehen ist. Kommentieren Sie Ihre Lösung ausführlich.

Fortsetzung nächste Seite!

- b) Implementieren Sie die Methode *betreten()*, welche das Passieren des Eingang-Drehkreuzes modelliert, sowie die Methode *verlassen()*, welche im Gegenzug das Verlassen des Schwimmbads über das Ausgangs-Drehkreuzes modelliert. Kommentieren Sie Ihre Lösung wieder ausführlich. Beachten Sie bei der Implementierung, dass alle oben genannten Bedingungen eingehalten werden.
- c) Was sind die kritischen Bereiche bei diesem Problem?
- d) Erläutern Sie kurz (2 Sätze), warum Ihre Lösung des wechselseitigen Ausschlusses die Bedingung der Mutual Exclusion erfüllt.
- e) Beschreiben Sie ausgehend von Java die Einschränkungen bei der Verwendung von *wait()*, *notify()* und *notifyAll()* gegenüber dem Monitorkonzept von C. Hoare (1974). **Hinweis:** Überlegen Sie sich dazu, wie im Falle des Monitorkonzepts von Hoare bzw. im Falle des Java Synchronisierungsmechanismus der nächste aktive Thread selektiert wird.

Fortsetzung nächste Seite!

```
public class Schwimmbad {  
    private static int maxLiegen;  
    private int anzahlGaeste;
```

```
    //Schreiben Sie hier den Konstruktor:
```

```
    // Implementierung der Methode "betreten":
```

```
    // Implementierung der Methode "verlassen":
```

```
}
```

Fortsetzung nächste Seite!

Die Klasse Simulation:

```
1 public class Simulation {
2     private Schwimmbad my_schwimmbad;
3
4     public Simulation() {
5         my_schwimmbad = new Schwimmbad(5);
6
7         for(int i=0; i< 50; i++)
8         {
9             Badegast gast = new Badegast(my_schwimmbad);
10            gast.start();
11        }
12    }
13    public static void main(String[] args) {
14        new Simulation();
15    }
16 }
```

Die Klasse Badegast:

```
1 import java.util.Random;
2
3 public class Badegast extends Thread {
4     private Schwimmbad my_schwimmbad;
5     private Random generator;
6
7     public Badegast(Schwimmbad my_schwimmbad) {
8         this.my_schwimmbad = my_schwimmbad;
9         generator = new Random();
10    }
11
12    public void run() {
13        try {
14            Thread.sleep(generator.nextInt(2000) + 500);
15            my_schwimmbad.betreten();
16            //Liege belegen, schwimmen usw.
17            Thread.sleep(generator.nextInt(2000) + 500);
18            my_schwimmbad.verlassen();
19        }
20        catch(InterruptedException ie) {}
21    }
22 }
```