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

**Kennzahl:** _____

**Kennwort:** _____

**Arbeitsplatz-Nr.:** _____

**Frühjahr 2013**

**66116**

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

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbanksysteme, Softwaretechnologie**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **13**

**Bitte wenden!**

# Thema Nr. 1

**Teilaufgabe 1:**

1. Allgemeinwissen

a) Definieren Sie den Begriff Datenbanksystem und grenzen Sie die Funktionalitäten eines Datenbanksystems von denen eines Dateisystems ab.
b) Nennen Sie die Phasen des Datenbankentwurfsprozesses und charakterisieren Sie diese kurz.
c) Nennen und beschreiben Sie kurz die vier wesentlichen Eigenschaften von Transaktionen.
d) Nennen und erläutern Sie kurz vier Beispiele für Integritätsbedingungen.

2. ER-Modellierung und Relationenmodell

a) Erstellen Sie ein ER-Diagramm bestehend aus Entitäts- und Beziehungstypen sowie Attributen für folgendes Szenario. 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 eine Mitarbeiterverwaltung, die folgendermaßen beschrieben ist:
   1. Mitarbeiter werden über ihre Sozialversicherungsnummer identifiziert. Sie haben einen Namen, bestehend aus Vor- und Nachnamen, sowie ein Geburtsdatum und ein Alter, das aus dem Geburtsdatum errechnet wird.
   2. Ein Mitarbeiter kann der Vorgesetzte anderer Mitarbeiter sein.
   3. Jeder Mitarbeiter ist einer Abteilung zugeordnet. Eine Abteilung wird über ihren Namen und eine Nummer identifiziert (z.B. „Entwicklung", „3"). Zu einer Abteilung werden mehrere Orte gespeichert, an denen die Abteilung angesiedelt ist.
   4. Mitarbeiter können Abteilungen leiten. Jede Abteilung wird von einem Mitarbeiter geleitet, aber nicht jeder Mitarbeiter muss eine Abteilung leiten.
   5. Abteilungen können Projekte realisieren. Ein Projekt wird durch eine Nummer identifiziert und findet an einem Ort statt.
   6. Jeder Mitarbeiter arbeitet an einem oder mehreren Projekten. An jedem Projekt muss mindestens ein Mitarbeiter arbeiten. Zusätzlich wird gespeichert, ab wann ein Mitarbeiter an einem Projekt arbeitet.

   **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 einfach uneingeschränkt bleiben (alle formulierten Einschränkungen sind zu erkennen).

b) Erstellen Sie zu dem E/R-Diagramm aus Teil (a) ein Relationenschema. Seien Sie dabei so vollständig wie möglich – bspw. Berücksichtigung totaler Partizipationen. Vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen.

**Fortsetzung nächste Seite!**

**Beispiel für die Notation:**

```
Relationenname (Primärschlüssel, Attribut1, Attribut2, …,
    Fremdschlüssel[AndereRelation],
    (FremdschlüsselattributA,
    FremdschlüsselattributB)[DritteRelation])
    Attribut2 NOT NULL
```

## 3. Normalformen

Gegeben sei die nachfolgende relationale Datenbank mit unterstrichenen Schlüsselattributen. Sie enthält Informationen über Opernaufführungen. Relation "Opernvorstellungen":

| Oper | Komponist | Datum | Ort | Opernhaus | Plätze | Dirigent | Dauer | Solist |
|---|---|---|---|---|---|---|---|---|
| Rigoletto | Verdi | 08.04.2013 | Wien | Staatsoper | 2298 | Lopez-Cobos | 2:00 | Polenzani |
| Don Carlos | Verdi | 09.04.2013 | Wien | Staatsoper | 2298 | de Billy | 4:00 | Youn |
| Rigoletto | Verdi | 11.04.2013 | Dresden | Semperoper | 1300 | Fisch | 2:00 | Berrugi |
| Macbeth | Verdi | 16.04.2013 | Mailand | Scala | 2000 | Gergiev | 2:45 | Vassallo |
| Carmen | Bizet | 20.05.2013 | Wien | Staatsoper | 2298 | de Billy | 2:45 | Garanca |
| Carmen | Bizet | 02.06.2013 | Wien | Staatsoper | 2298 | de Billy | 2:45 | Alagna |
| Don Carlos | Verdi | 12.10.2013 | Mailand | Scala | 2000 | Luisi | 4:00 | Pape |

Gegeben seien ebenso die folgenden funktionalen Abhängigkeiten:

FD1: Oper → Komponist, Dauer
FD2: Ort → Opernhaus, Plätze
FD3: Oper, Ort → Dirigent
FD4: Oper, Datum, Ort → Solist

a) Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.

b) Welcher Normalform genügt das angegebene Relationenschema? Begründen Sie Ihre Entscheidung.

c) Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung haben kann.

## 4. Anfrageverarbeitung

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

```
SELECT e1.name AS ename, e1.salary, dept.name AS dname,
       AVG (e2.salary) AS avg_salary
  FROM employee e1, employee e2, dept
 WHERE e1.dept_id = dept.id
   AND e2.dept_id = e1.dept_id
   AND dept.name LIKE 'Sales %'
 GROUP BY e1.name, e1.salary, dept.name
   HAVING AVG(e2.salary) > 2*e1.salary
```

b) Wie kann die relationale Algebra zur Optimierung von SQL-Anfragen eingesetzt werden?

**Fortsetzung nächste Seite!**

**Teilaufgabe 2:**

Aufgabe 1:

Sie sollen für eine Tankstelle eine Software zur Verwaltung von DVDs entwickeln. Jede DVD hat eine Nummer, einen Titel, eine Altersbeschränkung und kann entweder ausleihbar, für einen Kunden reserviert, oder von einem Kunden ausgeliehen sein. Eine Reservierung trägt ein Verfallsdatum; eine Ausleihe trägt ein Rückgabedatum.

Die beteiligten Personen können Kunden, Kassierer, oder Administratoren sein; der Einfachheit halber können Kassierer und Administratoren nicht Kunden sein.

Beteiligte Personen können sich beim System an- und wieder abmelden.

Kunden können DVDs reservieren, ausleihen und zurückgeben, und außerdem Gebühren schulden und bezahlen. Außerdem können Kunden den Bestand durchsuchen und Verfügbarkeitsdaten einsehen.

Kassierer können DVDs verleihen, sowie Rückgaben und Reservierungen entgegennehmen.

Außerdem können sie Kunden und DVDs neu aufnehmen oder löschen. Auch Kassierer können den Bestand durchsuchen, dabei aber im Gegensatz zu Kunden auch die ausleihenden Personen einsehen.

Administratoren haben alle Befugnisse von Kassierern und können außerdem Kassierer und Administratoren aufnehmen oder löschen.

- Entwickeln Sie aus der gegebenen Anwendungsbeschreibung ein UML-Klassendiagramm. Alle Angaben aus dem Text sollen sich in Ihrem Modell wiederfinden; weitere sollten nur dann hinzugenommen werden, wenn es aufgrund der von Ihnen gewählten Modellierung erforderlich erscheint.
- Verwenden Sie Vererbung und Interfaces, wo es sinnvoll möglich ist, und achten Sie auf eine angemessene Darstellung von Assoziationen und Aggregationen, wo erforderlich. Es geht hier nur um das UML-Klassendiagramm; insbesondere ist eine Modellierung als relationale Datenbank nicht verlangt und auch nicht im Sinne der Aufgabenstellung.
- Modellieren Sie den gesamten Anwendungsfall des Ausleihens einer DVD (Anmelden, Bestand anzeigen, Aussuchen, Datum eingeben, Bestätigen der Ausleihe, Abmelden) zunächst als UML-Anwendungsfall und dann als UML-Aktivitätsdiagramm. Achten Sie auch auf die Möglichkeit des Fehlschlagens einer Teilaktivität.
- Modifizieren Sie Ihr Modell so, dass Personen sowohl Administratoren bzw. Kassierer als auch Kunden sein können.

**Fortsetzung nächste Seite!**

Aufgabe 2:

Das Entwurfsmuster Kompositum dient der objektorientierten Modellierung hierarchischer, insbesondere rekursiver Datenstrukturen.

- Geben Sie ein allgemeines UML-Klassendiagramm für dieses Entwurfsmuster an.
- Geben Sie für ein konkretes Anwendungsbeispiel dieses Entwurfsmusters UML-Klassendiagramm und UML-Objektdiagramm für eine repräsentative Instanz an. Ihre Klassen sollten mindestens eine Methode beinhalten; Ihr Objektdiagramm sollte mindestens drei Objekte umfassen.
- Nennen Sie zwei weitere Entwurfsmuster und beschreiben Sie ihre Einsatzmöglichkeiten in jeweils ca. drei Sätzen.

# Thema Nr. 2

## Teilaufgabe 1:

**Aufgabe 1: Relationale Algebra und SQL**

Gegeben sei folgendes Schema:

- Züge:
  {[ZugNr: integer, AnzahlWagons : integer]}
- Städte:
  {[StadtNr: integer, Name: string]}
- Bahnhöfe:
  {[BahnhofNr: integer, Name: string, LiegtInStadtNr: integer]}
- Verbindungen:
  {[ZugNr: integer, BahnhofNrVon: integer, BahnhofNrNach: integer, AbfahrtsZeit: date, AnkunftsZeit: date]}

**Hinweis:**
Ein Element ∈ date hat der Einfachheit halber die Form: „5:32PM". Wir nehmen an, dass Verbindungen jeden Tag gelten.

**Aufgabe 1.1:**
Geben Sie tabellarisch eine Beispielausprägung mit mindestens zwei Zeilen pro Tabelle des angegeben Schemas an.

**Aufgabe 1.2**

1. Finden Sie die Züge, welche morgens um 5 Uhr von dem Bahnhof „Berlin Ostbahnhof" zu dem Bahnhof „München Hauptbahnhof" fahren **(in relationaler Algebra).**
2. Finden Sie die Städte, von denen mindestens ein Zug mit mehr als drei Wagons zu einem Bahnhof in Mannheim fährt **(in relationaler Algebra).**
3. In welcher Stadt liegt der Bahnhof „Coesfeld Schulzentrum"? **(in relationaler Algebra)**
4. Finden Sie die Namen der Städte an denen zur gleichen Uhrzeit mehr als ein Zug ankommt **(in relationaler Algebra).**

**Fortsetzung nächste Seite!**

**Aufgabe 1.3:**

Beantworten Sie folgende Fragen in SQL:

1. An welchen Uhrzeiten fahren Züge von „München Hauptbahnhof“ nach „Berlin Ostbahnhof“?
2. Finden Sie den Zug oder die Züge mit den meisten Wagons.
3. Welche Bahnhöfe (BahnhofNr angeben) sind mit dem Bahnhof “München Hauptbahnhof” direkt verbunden? Geben Sie die Bahnhöfe duplikatfrei aus. „München Hauptbahnhof“ selbst darf nicht im Ergebnis vorkommen.
4. Finden sie alle Bahnhöfe (BahnhofNr), die mit **genau zweimal** Umsteigen von dem Bahnhof „München Hauptbahnhof“ aus erreicht werden können.
5. Geben Sie die Namen der Bahnhöfe an, die überdurchschnittlich oft als Ziel vorkommen. Geben Sie außerdem zu jedem der Bahnhöfe die Anzahl der ankommenden Züge an. Beachten Sie hierbei, dass auch Bahnhöfe, die von keinem Zug erreicht werden können, den Durchschnitt beeinflussen.

**Aufgabe 1.4:**

Gegeben sei folgende SQL Anfrage:

```
SELECT z.ZugNr FROM
        Zug z, Bahnhof b1, Bahnhof b2, Verbindung v
WHERE
        v.BahnhofNrVon = b1.BahnhofNr AND
        v.BahnhofNrNach = b2.BahnhofNr AND
        b1.Name = “München” AND
        b2.Name = “Berlin” AND
        z.ZugNr = v.ZugNr
```

Übertragen Sie das gegebene SQL Statement in die relationale Algebra und optimieren Sie den Ausdruck logisch. Verwenden Sie die Operatorbaum-Syntax.

**Fortsetzung nächste Seite!**

**Aufgabe 2: ER Modellierung**

Sie sind beauftragt, bei der Entwicklung des ersten Prototypen eines sozialen Netzwerkes die Datenmodellierung zu übernehmen. Der erste Prototyp soll folgende Anforderungen erfüllen:

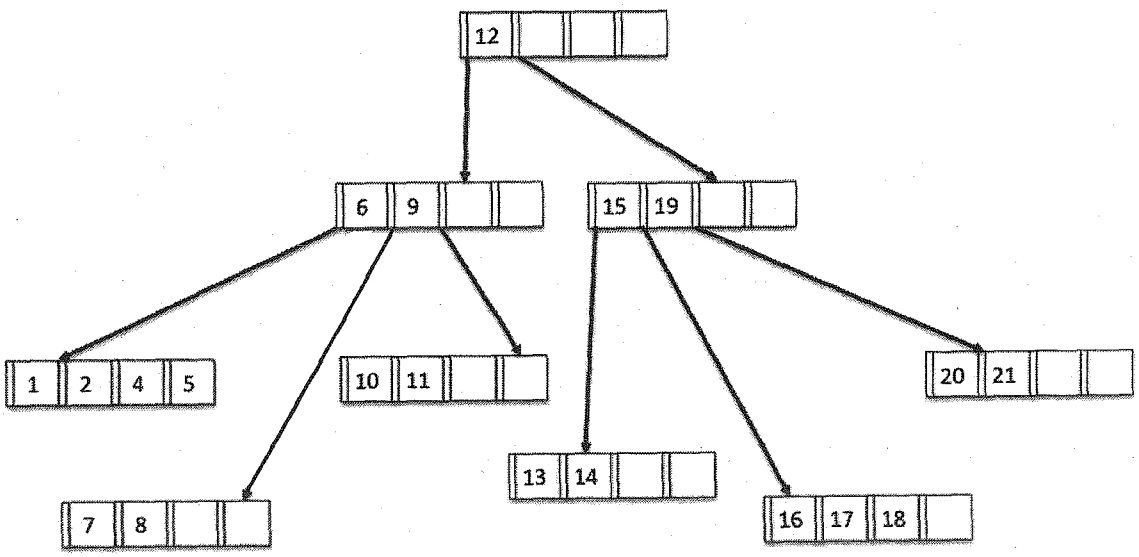
*Benutzer können sich mit ihrem Nutzernamen und Kennwort anmelden. Ein Benutzer hat eine sogenannte Wall, auf die andere Benutzer Texte schreiben können. Benutzer können Freunde von anderen Benutzern sein. Außerdem können Benutzer sich in verschiedenen Gruppen anmelden. Gruppen haben ebenfalls eine sogenannte Wall, auf der Gruppenmitglieder Nachrichten hinterlassen können.*

Erstellen Sie ein ER-Diagramm, welches die obengenannten Anforderungen erfüllt. Denken Sie daran Ihrem ER-Diagramm Funktionalitätsangaben hinzuzufügen.

**Fortsetzung nächste Seite!**

**Aufgabe 3: B-Baum**

Gegeben sei der folgende B-Baum




1. Was bedeutet $k$ bei einem B-Baum mit Grad $k$? Geben Sie $k$ für den obigen B-Baum an.
2. Was sind die Vorteile von B-Bäumen im Vergleich zu binären Baumen?
3. Wozu werden B-Bäume in der Regel verwendet und wieso?
4. Fügen Sie den Wert **3** in den B-Baum ein, und zeichnen Sie den vollständigen B-Baum nach dem Einfügen und möglichen darauf folgenden Operationen.
5. Entfernen Sie aus dem **ursprünglichen** B-Baum den Wert **19**. Zeichnen Sie das vollständige Ergebnis nach dem Löschen und möglichen darauf folgenden Operationen. Sollte es mehrere richtige Lösungen geben, reicht es eine Lösung zu zeichnen.

**Fortsetzung nächste Seite!**

**Teilaufgabe 2:**

**Aufgabe 1: Allgemeines**

a) Was ist unter Softwaretechnik zu verstehen?

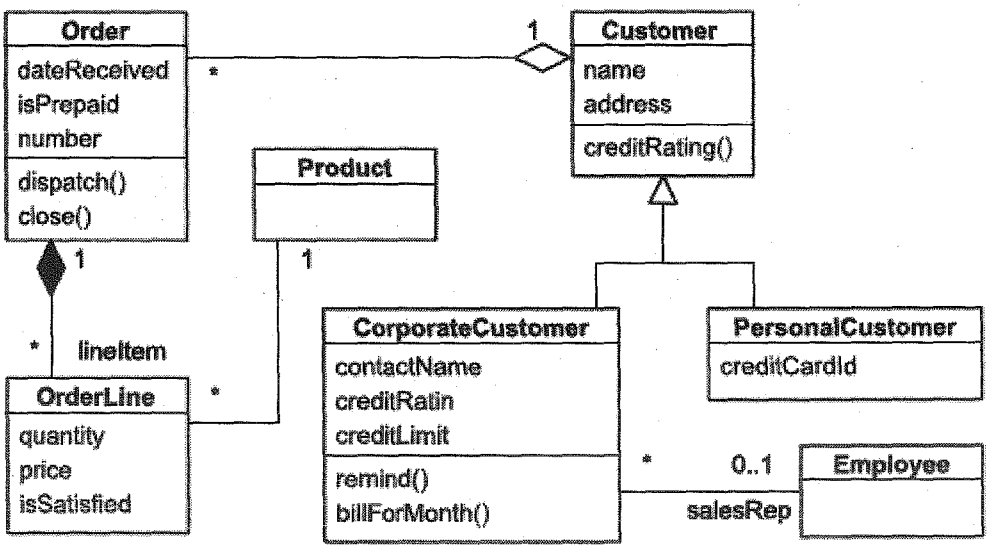
b) Nennen sie ein Vorgehensmodell (oder auch Prozess-Modell) in der Softwareentwicklung. Welcher ist der wesentliche Vorteil dieses Modells gegenüber anderen?

c) Nennen sie alle am Softwareentwicklungsprozess beteiligten Rollen (Akteure) und ordnen sie diesen mindestens eine Phase zu.

**Fortsetzung nächste Seite!**

**Aufgabe 2: UML**

Erstellen Sie ein Klassendiagramm, das UML-Klassendiagramme modelliert, d.h. ein Meta-Modell für UML-Klassendiagramme. Ein Klasse besitzt einen Namen, Methoden und Attribute. Bilden Sie mindestens die in dem unten angegebenen Klassendiagramm verwendeten UML-Klassendiagramm-Elemente ab. Die Modellierung von Typen und Anfangswerten ist nicht erforderlich.




**Aufgabe 3 (Architekturbeschreibung)**

Erstellen Sie eine Architekturbeschreibung für ein einfaches Studentenverwaltungssystem, das die Prüfungsbeteiligung (mit Status und Note) von Studenten (mit Name, Vorname, Matrikelnummer) an angebotenen Prüfungen (mit Bezeichnung, Ort, Zeit) verwalten soll. Eine Komponente „Prüfungsbeteiligung" soll folgende typische Aufgaben unterstützen:

1. Anmeldung und Abmeldung eines Studenten zu einer bestimmten Prüfung,
2. Abfrage des Status einer Prüfungsbeteiligung (angemeldet, abgemeldet, teilgenommen, bestanden, durchgefallen) zu einer bestimmten Prüfung durch einen Studenten,
3. Abfrage der Note eines Studenten zu einer Prüfung,
4. Eintragung der Teilnahme eines Studenten an einer Prüfung durch die Übungsleitung,
5. Eintragung der Note eines Studenten an einer Prüfung durch die Übungsleitung.

Nicht zu berücksichtigen sind dabei Authentifizierung und Autorisierung.

Geben Sie folgende Bestandteile einer Architekturbeschreibung des Systems an:

1. Datenmodell (als ER- oder Klassendiagramm),
2. Spezifikation der Methoden der Komponente „Prüfungsbeteiligung" nach „design by contract" (Vorbedingung, Nachbedingung, Invariante).

**Fortsetzung nächste Seite!**

## Aufgabe 4: Funktionale Programmierung

Wenn Sie die Sprache Haskell kennen, benutzen Sie sie zur Beantwortung dieser Frage. Ansonsten benutzen Sie die funktionale Sprache, in der Sie ausgebildet wurden. Bleiben Sie aber rein funktional, also vermeiden Sie Seiteneffekte oder wiederholte Zuweisungen.

In dieser Frage betrachten wir Funktionen, die eine Liste revertieren. D.h. die Ausgabeliste enthält dieselben Elemente wie die Eingabeliste, nur in umgekehrter Reihenfolge.

a) Einfache Verkettung. Benutzen Sie hier den in der Sprache mitgelieferten Listentyp (in Haskell `[a]`, in ML `'a list`, etc.). Definieren Sie eine Funktion `reverse`, die eine Liste revertiert. Dazu geben Sie

   i. die Signatur (also den Typ) der Funktion und

   ii. ihre definierenden Gleichungen an.

   iii. Welche asymptotische Zeitkomplexität in der Länge der Eingabeliste hat Ihre Funktion?

b) Doppelte Verkettung.

   i. Definieren Sie einen neuen Datentyp für Listen mit Elementen vom Typ `a`, bei denen Elemente sowohl am Anfang als auch am Ende eingefügt werden können, in Haskell `DList a`. Die entsprechenden Konstruktoren sollen `Cons` (für Anhängen am Anfang) und `Snoc` (für Anhängen am Ende) heißen.

   ii. Definieren Sie (mit Signatur und Gleichungen) eine Funktion `reverseDList`, die eine `DList a` revertiert.

   iii. Welche asymptotische Zeitkomplexität in der Länge der Eingabeliste hat diese Funktion?

**Fortsetzung nächste Seite!**

## Aufgabe 5: Verifikation

Diese Frage befasst sich mit dem Begriff der Abstiegsfunktion zum Beweis der Terminierung von Rekursionen.

a) Begriffsdefinition. Definieren Sie den Begriff der Abstiegsfunktion durch die Angabe folgender Dinge:

   i. ihrer Signatur (d.h. ihrer Urbild- und Bildmenge),

   ii. ihrer Eigenschaften.

b) Beispiel. Geben Sie eine Abstiegsfunktion für die folgende Funktionsdefinition an:

```
even(x) = if x<0 then even (-x)
                 else if x==0 then True
                              else if x==1 then False
                                           else even (x-2)
```

Verfahren Sie dazu, wie folgt:

   i. Definieren Sie die Abstiegsfunktion durch Angabe von Signatur und Gleichungen.

   ii. Weisen Sie mathematisch nach, dass die Funktion die Anforderungen an eine Abstiegsfunktion erfüllt.