
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2017**

46116

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

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Softwaretechnologie/Datenbanksysteme**

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

Anzahl der Druckseiten dieser Vorlage: **15**

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1: Begriffe und Konzepte

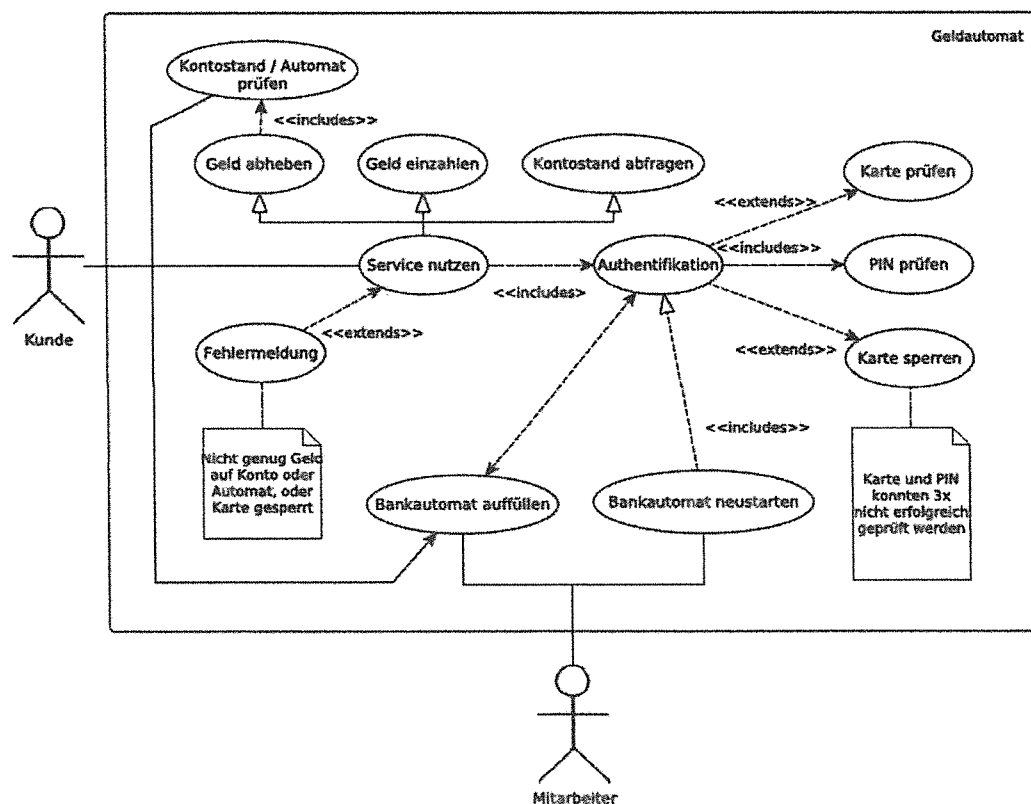
- a) Erläutern Sie kurz den Unterschied zwischen starker und schwacher Kohäsion einer Softwarekomponenten.
- b) Erläutern Sie kurz den Unterschied zwischen starker und schwacher Kopplung zwischen Softwarekomponenten.
- c) Wie wirkt sich der Einsatz von public-Instanzvariablen in einer Klasse auf die Kopplung im System aus und warum?
- d) Wie wirkt sich der Einsatz des Entwurfsmusters MVC auf Kopplung und Kohäsion des Systems aus und warum?
- e) Nennen Sie drei Merkmale des Extreme Programming.
- f) Was versteht man unter dem Liskovschen Substitutionsprinzip?
- g) Welche Arten von Softwarewartung werden unterschieden?

Fortsetzung nächste Seite!

Aufgabe 2: Use-Case-Diagramm

Im Folgenden sehen Sie ein fehlerhaftes Use-Case-Diagramm für das System „Geldautomat“. Geben Sie ein korrektes Use-Case-Diagramm (Beziehungen und Beschriftungen) entsprechend der folgenden Beschreibung an (nicht auf dem Aufgabenblatt berichtigen). Sollte es mehrere Möglichkeiten geben, begründen Sie Ihre Entscheidung kurz.

Kunden können am Geldautomaten verschiedene Services nutzen, es kann Geld abgehoben und eingezahlt sowie der Kontostand abgefragt werden. Für jeden dieser Services ist eine Authentifizierung notwendig. Diese besteht aus der Prüfung der Bankkarte und des eingegebenen PINs. Manche Bankautomaten können Karten bei zu vielen Fehlversuchen (3) bei der Anmeldung sperren, andere geben Fehlermeldungen aus, falls die Karte gesperrt wurde oder nicht mehr genügend Geld auf dem Konto oder im Bankautomaten vorhanden ist. Mitarbeiter bzw. Mitarbeiterinnen der Bank können sich ebenfalls über PIN und Karte authentifizieren, um dann den Bankautomaten neu zu starten oder mit Geld aufzufüllen. Bevor Geld abgehoben werden kann, ist sicherzustellen, dass auf dem Konto und im Bankautomaten genügend Geld vorhanden ist.

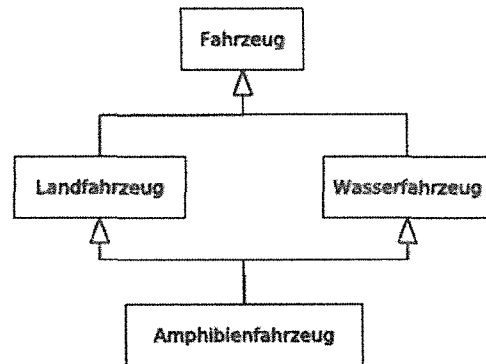
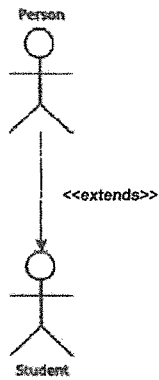


Fortsetzung nächste Seite!

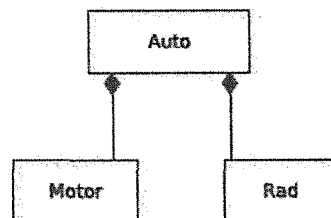
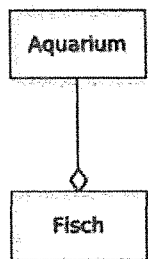
Aufgabe 3: UML Diagramme

Betrachten Sie die folgenden UML-Diagramme. Sind diese korrekt? Falls nein, begründen Sie warum nicht. Geben Sie in diesem Fall außerdem eine korrigierte Version an.

a)



b)

**Fortsetzung nächste Seite!**

Aufgabe 4: Entwurfsmuster

In einem Abenteuerspiel soll ein Held Monster bekämpfen. Zu Beginn ist die Spielfigur schwach und kann als Attacke nur mit den Fäusten boxen. Im Laufe des Spiels erhält der Charakter diverse Items und wird dadurch zu einer stärkeren Figur. Bei jeder Attacke auf ein Monster werden alle Items, die er erworben hat, der Reihe nach eingesetzt.

Der folgende Code zeigt einen Ausschnitt aus der Implementierung.

Welches Designpattern wurde hier eingesetzt? Zeichnen Sie das zugehörige UML-Diagramm, das genau die im Code beschriebene Situation wiedergibt.

```
interface Spielfigur {
    public void attackieren();
}

class EinfacheFigur implements Spielfigur {
    public void attackieren() {
        // boxen();
    }
}

abstract class BewaffneteFigur implements Spielfigur {
    protected Spielfigur figur;

    public BewaffneteFigur (Spielfigur figur) {
        this.figur = figur;
    }
    public void attackieren() {
        figur.attackieren();
    }
}

class FigurMitHammer extends BewaffneteFigur {
    public FigurMitHammer (Spielfigur figur) {
        super(figur);
    }
    public void attackieren() {
        // mitHammerZuschlagen();
        super.attackieren();
    }
}

class FigurMitTraenengas extends BewaffneteFigur {
    public FigurMitTraenengas (Spielfigur figur) {
        super(figur);
    }
    public void attackieren() {
        // spruehen();
        super.attackieren();
    }
}

class FigurMitWasserpistole extends BewaffneteFigur {
    public FigurMitWasserpistole (Spielfigur figur) {
        super(figur);
    }
    public void attackieren() {
        // nassspritzen();
        super.attackieren();
    }
}
```

Fortsetzung nächste Seite!

Aufgabe 5: Projektmanagement

Betrachten Sie die folgende Tabelle zum Projektmanagement:

Arbeitspaket	Dauer (Tage)	Abhängig von
A1	5	
A2	10	
A3	5	A1
A4	15	A2, A3
A5	10	A1
A6	10	A1, A2
A7	10	A2, A4
A8	15	A4, A5

Tabelle 1: Übersicht Arbeitspakete

- Erstellen Sie ein Gantt-Diagramm, das die in der Tabelle angegebenen Abhängigkeiten berücksichtigt.
- Wie lange dauert das Projekt mindestens?
- Geben Sie den oder die kritischen Pfad(e) an.
- Konstruieren Sie ein PERT-Chart zum obigen Problem.

Aufgabe 6: Entwurfsmuster Abstract Factory

Zeichnen Sie das Strukturdiagramm des Entwurfsmusters Abstract Factory. Erklären Sie die Wirkungsweise und den Vorteil der Verwendung dieses Entwurfsmusters.

Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanksysteme**Aufgabe 1: Grundlagen**

- a) Grenzen Sie die Begriffe Datenbanksystem, Datenbankverwaltungssystem und Datenbank in ein bis zwei Sätzen voneinander ab.
- b) Erläutern Sie in ein bis zwei Sätzen, was man unter physischer Datenunabhängigkeit versteht.
- c) Erläutern Sie in ein bis zwei Sätzen, was ein Trigger ist. Nennen Sie weiterhin einen beispielhaften Einsatzzweck eines Triggers.

Aufgabe 2: Konzeptioneller Entwurf

Der Metadatenkatalog eines Datenbankverwaltungssystems wird meist ebenfalls in Relationen abgelegt. Im Folgenden finden Sie die Beschreibung eines Ausschnitts des Metadatenkatalogs eines Datenbankverwaltungssystems. Erstellen Sie zu dieser Beschreibung ein ER-Diagramm. Kennzeichnen Sie die Primärschlüssel durch passendes Unterstreichen und geben Sie die Kardinalitäten in Chen-Notation (= Funktionalitäten) an. Modellieren Sie keine Attribute oder Entitytypen, die nicht aus dem Text hervorgehen.

Die Datenbank enthält Datenobjekte, die genau einem Benutzer gehören und durch den Benutzer zusammen mit einem für diesen Benutzer eindeutigen Namen identifiziert werden. Ein Benutzer hat ein Passwort und einen eindeutigen Benutzernamen. Benutzer können anderen Benutzern Zugriffsrechte auf Datenobjekte geben. Dabei wird auch die Berechtigungsart gespeichert. Datenobjekte sind entweder Tabellen oder Indexe. Eine Tabelle kann verschiedene Spalten enthalten. Jede Spalte gehört zu genau einer Tabelle und hat einen Datentyp sowie einen innerhalb ihrer Tabelle eindeutigen Namen. Indexe haben eine Gültigkeit. Es gibt B-Baum- und Hash-Indexe. Zu einem B-Baum-Index wird die aktuelle Höhe abgelegt, zu einem Hash-Index der aktuelle Belegungsfaktor. Jeder Index und jede Tabelle ist in genau einem eigenen Segment abgelegt. Segmente werden durch eine eindeutige Nummer identifiziert und haben eine Größe. Jedes Segment ist in einer Datei abgelegt. Eine Datei kann mehrere Segmente enthalten. Dateien haben einen eindeutigen Pfad.

Aufgabe 3: SQL

Gegeben sind folgende Relationen aus der Verwaltungssoftware eines Fahrradverleihs:

Fahrrad (FahrradID, Marke, Modell, Tagesgebuehr)

Kunde (KundenID, Vorname, Nachname)

Ausleihe (KundenID[Kunde], FahrradID[Fahrrad], Entleihdatum, Rueckgabedatum)

- a) Schreiben Sie eine SQL-Abfrage, die zu allen Ausleihen das Entleihdatum zusammen mit dem Nachnamen des Kunden ausgibt, der die Ausleihe getätigt hat. Die Ausgabe soll absteigend nach Nachnamen sortiert sein.

Fortsetzung nächste Seite!

- b) Schreiben Sie eine SQL-Abfrage, die für jeden Kunden die Ausleihe mit der längsten Dauer (= Rueckgabedatum – Entleihdatum) ermittelt und deren Dauer zusammen mit der KundenID ausgibt. Die Ausgabe soll nur Ausleihdauern größer 7 enthalten.
- c) Schreiben Sie eine SQL-Abfrage, die FahrradID und Entleihdatum aller Ausleihen zusammen mit der für die jeweilige Ausleihe insgesamt anfallenden Ausleihgebühr ausgibt. Für den Entleihtag wird keine Gebühr berechnet, für den Rückgabetag und alle anderen Tage wird jeweils die Tagesgebühr berechnet.
- d) Es soll ermittelt werden, ob die Tabelle Ausleihe Datensätze mit unmöglichen Ausleihvorgängen enthält.

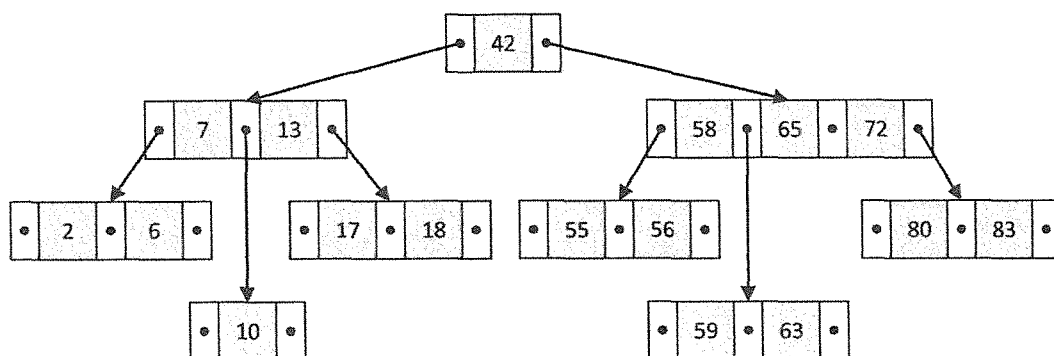
Hierzu sollen FahrradID und Entleihdatum aller Datensätze der Tabelle Ausleihe ausgegeben werden, bei denen sich die Entleihzeit mit der Entleihzeit anderer Datensätze für die gleiche FahrradID überschneidet. Das Entleihen eines Fahrrads am gleichen Tag seiner Rückgabe aus einem anderen Ausleihvorgang gilt nicht als Überschneidung.

Geben Sie eine entsprechende SQL-Abfrage an.

- e) Schreiben Sie eine SQL-Abfrage, die für alle Fahrräder der Marke „Schnelles Rad“ die Tagesgebühr um 3% erhöht.

Aufgabe 4: Physische Datenorganisation

- a) Erläutern Sie in ein bis zwei Sätzen, aus welchen zwei Teilen sich ein TID (Tupel-Identifikator) zusammensetzt.
- b) Erläutern Sie in ein bis zwei Sätzen das Vorgehen, wenn ein durch einen TID adressierter Satz innerhalb einer Seite verschoben werden muss.
- c) Erläutern Sie in ein bis zwei Sätzen das Vorgehen, wenn ein durch einen TID adressierter Satz erstmalig in eine andere Seite verschoben werden muss.
- d) Erläutern Sie in zwei bis drei Sätzen das Vorgehen, wenn ein durch einen TID adressierter und bereits einmal über Seitengrenzen hinweg verschobener Satz erneut in eine andere Seite verschoben werden muss.
- e) Nennen Sie zwei Gründe, warum der folgende Baum kein gültiger B-Baum ist. Beziehen Sie sich dabei auf konkrete Stellen im gegebenen Baum.



Fortsetzung nächste Seite!

Aufgabe 5: Transaktionen

- a) Nennen Sie die Bezeichnungen der vier wesentlichen Eigenschaften des Transaktionskonzepts und erläutern Sie jede Eigenschaft in ein bis zwei Sätzen.
- b) Erläutern Sie in ein bis zwei Sätzen, ob innerhalb einer Transaktion ein inkonsistenter Zustand vorliegen darf.
- c) Gegeben ist die folgende Historie (Schedule) dreier Transaktionen:
 $w_1(O) \rightarrow w_1(O) \rightarrow r_3(P) \rightarrow r_1(P) \rightarrow c_1 \rightarrow r_2(T) \rightarrow w_3(T) \rightarrow w_2(P) \rightarrow w_2(O) \rightarrow c_2 \rightarrow c_3$

Zeichnen Sie den Serialisierbarkeitsgraphen zu dieser Historie und begründen Sie, warum die Historie serialisierbar ist oder nicht.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1: „Begriffe und Konzepte“

- a) Nennen Sie vier Qualitätsmerkmale von Software.
- b) Was versteht man unter dem Begriff „Refactoring“?
- c) Wie stehen „Refactoring“ und „Regression-Testing“ in Beziehung?
- d) Nennen Sie drei Vorteile für den Einsatz von Software zur Versionsverwaltung.

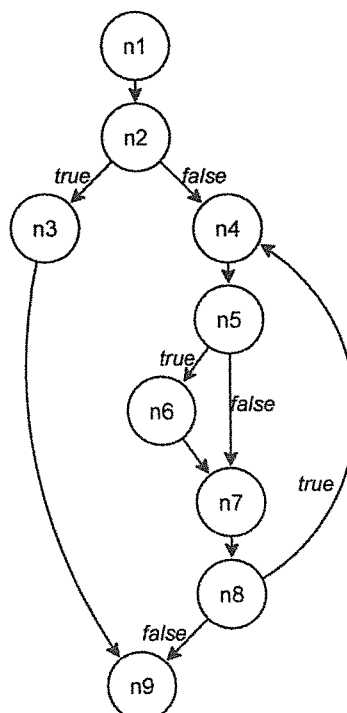
Fortsetzung nächste Seite!

Aufgabe 2: „Testen“

Gegeben sei folgende Java-Methode:

```
static int test(int x) {  
    int y = 0;  
    if (x < 0) {  
        x = -x;  
    }  
    do {  
        if (x == 0) {  
            x++;  
        } else if (x % 2 == 1) {  
            x--;  
        } else {  
            x = x/2;  
            y++;  
        }  
    } while(x > 0);  
    return y;  
}
```

- Konstruieren Sie den Kontrollflussgraphen des obigen Code-Fragments.
- Nennen Sie die maximale Anzahl linear unabhängiger Programmpfade, also die zyklomatische Komplexität nach McCabe, und geben Sie an, wie Sie den Wert ermittelt haben.
- Geben Sie einen Testdatensatz mit möglichst wenigen Testfällen (d.h. jeweils den Aufrufparameter der Methode) an, der eine 100%-ige Verzweigungsüberdeckung der Methode **test(x)** erzielt. Welche Pfade werden dabei jeweils durchlaufen?
- Welche Pfade sind **im folgenden** Kontrollflussgraphen einer anderen Methode zu überdecken, um eine vollständige „Schleife-Inneres-Überdeckung“ („boundary interior coverage“) zu erreichen?



Fortsetzung nächste Seite!

- e) Beschreiben Sie kurz, welche Eigenschaften eine Testfallmenge allgemein haben muss, damit sie das datenflussorientierte Überdeckungskriterium „all-uses“ erfüllt.

Aufgabe 3: „Entwurfsmuster“

Beschreiben Sie das Entwurfsmuster „Zustandsmuster“, das zur Kapselung unterschiedlicher, zustandsabhängiger Verhaltensweisen eines Objektes eingesetzt wird. Geben Sie dazu ein Klassendiagramm an (mindestens drei verschiedene Zustände). Zeichnen Sie in diesem auch die Interaktion mit der das Muster nutzenden Klasse ein. Was ist der Hauptvorteil im Vergleich zu einer Umsetzung von zustandsabhängigen Operationen mittels Bedingungen?

Aufgabe 4: „Zustandsdiagramm und Klassendiagramm“

- a) Beschreiben Sie einen Protokollzustandsautomat für die Klasse Buch mit den Operationen ausleihen(Leser), zurückgeben(), vorbestellen(Leser) und den Buchzuständen BuchPräsent, BuchAusgeliehen, BuchVorbestellt, BuchReserviert (d.h. ein vorbestelltes Buch wird für den Leser, der es als erstes vorbestellt hat, zur Abholung bereitgestellt). Es kann beliebig viele Vorbestellungen geben. Der Fall, dass ein reserviertes Buch nicht abgeholt wird, ist nicht zu betrachten. In Ihrem Diagramm sollen die Zustandswechsel bei den Operationen vollständig dargestellt werden. Geben Sie auch an, welche Attribute und Assoziationen für die Buchzustände erforderlich sind, um die Zuordnung zu den Lesern zu verwalten.
- b) Geben Sie zu dem Protokollzustandsautomat das zugehörige Klassendiagramm gemäß Anwendung des Zustandsmusters aus der ersten Teilaufgabe an. Ein Buch hat dabei Titel und Autor. Der Leser ist als eigener Typ zu modellieren.
- c) Geben Sie weiterhin für eine Operation ausleihen(Leser) im Zustand BuchPräsent den zugehörigen Code in einer Notation Ihrer Wahl an (z.B. PseudoCode, Java, Aktivitätsdiagramm).

Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanksysteme**1. Aufgabe**

Gegeben sei eine Relation GRUPPEN(GruppenId, MatrNr, Datum) in der vermerkt wird, welche Studierende an welchem Termin an welcher Übungsgruppe teilgenommen haben. Folgende Bedingungen sollen gelten:

- Dieselbe Gruppe trifft sich nie zwei Mal am gleichen Tag.
- Eine Gruppe wird dann abgehalten, wenn mindestens ein Teilnehmer anwesend ist.
- Studierende dürfen die Gruppe nicht wechseln.

Beispielausprägung:

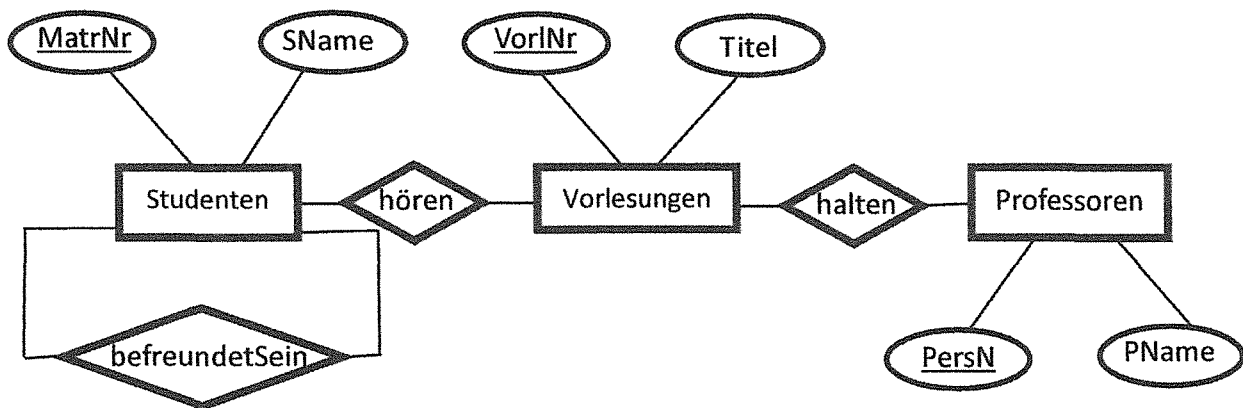
GRUPPEN		
GruppenID	MatrNr	Datum
1	123	2014-10-01
1	456	2014-10-01
1	456	2014-10-08
2	147	2014-10-02
2	258	2014-10-09
2	369	2014-10-09
3	159	2014-10-10

- Schreiben Sie eine SQL-Abfrage, die überprüft, dass die letztgenannte Bedingung (Studierende dürfen die Gruppe nicht wechseln) auch tatsächlich eingehalten wurde. Falls nicht, sollen die Studenten ausgegeben werden, die die Bedingung verletzt haben; ansonsten ein leeres Ergebnis.
- Schreiben Sie eine SQL-Abfrage, die für alle Studierenden (MatrNr) ausgibt, wie häufig sie an einer Übungsgruppe teilgenommen haben.
- Schreiben Sie eine SQL-Abfrage, die für alle Studierenden ausgibt, wie häufig sie in ihrer Übungsgruppe gefehlt haben.
- Geben Sie eine SQL-Abfrage an, welche die Gruppen ausgibt, bei denen nie ein Studierender gefehlt hat, wenn die Gruppe abgehalten wurde.

Fortsetzung nächste Seite!

2. Aufgabe

Folgendes ER-Diagramm sei gegeben:



- a) Übernehmen Sie das ER-Diagramm in Ihre Aufgabenbearbeitung und vervollständigen Sie es dort um Funktionalitätsangaben.

Ein „Datenbank-Laie“ hat das obige ER-Diagramm in folgendes Relationenschema übersetzt:

UniWelt: {[MatrNr, SName, FreundMatrNr, VorlNr, Titel, PersNr, PName]}

- b) Bestimmen Sie alle funktionellen Abhängigkeiten, geben Sie alle Schlüsselkandidaten der Relation UniWelt an und normalisieren Sie das Relationsschema UniWelt.

Dokumentieren Sie alle Schritte Ihres Vorgehens nachvollziehbar und geben Sie an, welche funktionalen Abhängigkeiten (FDs) in den generierten Schemata gelten.

- c) Formulieren Sie auf der Basis der normalisierten Relationen folgende Abfragen in SQL:

- Welche befreundeten Studenten-Paare hören dieselbe Vorlesung?
- Wer hört mindestens zwei unterschiedliche Vorlesungen, die der Professor namens „James Bond“ anbietet?

Fortsetzung nächste Seite!

3. Aufgabe

Fügen Sie in einen anfangs leeren B-Baum mit der Knotenkapazität 4 folgende Daten ein:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

Geben Sie auch die relevanten Zwischenergebnis-Bäume an.

Löschen Sie anschließend zuerst die 12 und danach die 4 aus dem Baum und geben Sie den Baum nach dem jeweiligen Löschvorgang an.

Beantworten Sie zusätzlich folgende Fragen:

- Welche Vorteile bietet ein B-Baum gegenüber einer Hash-basierten Indexierung?
- Welchen Vorteil bieten Hash-basierte Indexe im Vergleich zum B-Baum?

4. Aufgabe

Erklären Sie in kurzen Texten folgende Begriffe:

- WAL-Prinzip
- ACID
- Serialisierbarkeit
- physische Datenunabhängigkeit
- Sichten-Konsolidierung