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

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2021**

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**Teilaufgabe I: Softwaretechnologie****Aufgabe 1****(6 Punkte)**

- a) Erklären Sie ein Problem von Mehrfachvererbung in max. 2 Sätzen.
- b) Erklären Sie einen Vorteil von Mehrfachvererbung in max. 2 Sätzen.

Aufgabe 2**(13 Punkte)**

- a) Sie sind dabei, eine Softwareanwendung für Gärtner zu entwickeln. Konkret wollen Sie einen Garten modellieren in dem es v. a. verschiedene Baumarten gibt. Zeichnen Sie ein UML Klassendiagramm auf Basis der folgenden Beschreibung. Halten Sie sich dafür exakt an die beschriebenen Details.
- Die Klasse Baum ist eine abstrakte Klasse.
 - Die Klasse Baum hat eine abstrakte Methode reproduce.
 - Die Klasse Baum ist eine Oberklasse der Klasse Nussbaum.
 - Die Klasse Kiefer spezialisiert die Klasse Baum.
 - Die Klasse Kirschbaum ist eine Unterklasse der Klasse Baum.
 - Die Klasse Nuss ist eine Implementierung der Schnittstelle Samen.
 - Die Methode reproduce in Nussbaum instanziiert ein neues Nuss Objekt und gibt dieses zurück.

Ein Züchter kreuzt Kirschbaum und Nussbaum und erzeugt eine neue Baumart mit Namen KirschNussBaum. Tragen Sie auch diese Klasse in das Diagramm ein.

- b) Kann das Klassenmodell in Java so implementiert werden? Begründen Sie Ihre Entscheidung in max. 2 Sätzen.

Aufgabe 3**(8 Punkte)**

Der folgende Java-Programmcode berechnet die Anzahl an möglichen Permutationen für "dreimaliges Ziehen mit Zurücklegen". Der übergebende Parameter choices bestimmt dabei die Anzahl an Optionen pro Ziehung.

```
public class Permutate {  
  
    // Three draws with replacement  
    public int drawWithReplacement(int choices) {  
        return choices * choices * choices;  
    }  
}
```

Fortsetzung nächste Seite!

Schreiben Sie eine Methode in einer objektorientierten Programmiersprache, welche die obige Methode überlädt. Diese überladene Methode soll die Anzahl an Permutationen für z. B. 2, 4, und 11-maliges Ziehen mit Zurücklegen berechnen.

Aufgabe 4**(3 Punkte)**

Benennen Sie drei weitere grundlegende Prinzipien der Objektorientierung neben der Kapselung.

Aufgabe 5**(3 Punkte)**

Betrachten Sie den folgenden Java-Programmcode. (Die Zeilennummern gehören nicht dazu.)

```

1    public class Person {
2        private String firstname;
3        private String lastname;
4        private int age;
5
6        Person (String f, String l, int a) {
7            firstname = f;
8            lastname = l;
9            age = a;
10       }
11    }
```

- Benennen Sie die Art der Methode, die in den Zeilen 6-10 dargestellt ist.
- Geben Sie das objektorientierte Prinzip an, das in den Zeilen 2, 3 und 4 demonstriert wird.
- Welche Klassen werden im Beispielcode benannt?

Aufgabe 6**(8 Punkte)**

Betrachten Sie die folgende Tabelle und beziehen Sie sich auf diese in den zwei folgenden Teilaufgaben.

package A; class Aa {	package A; class Aa {	package A; class Ab {	package B; class Ac {	package B; class Ad extends Aa {
private int i;	?	?	?	?
int j;	?	?	?	?
protected int k;	?	?	?	?
public int l;	Ja	Ja	Ja	Ja
}	}	}	}	}

- Übernehmen Sie obige Tabelle in Ihre Bearbeitung und ersetzen Sie jedes '?' wie folgt: Wenn die in der jeweiligen Spaltenüberschrift genannte Klasse direkten Zugriff auf die in der jeweiligen Zeilenbeschriftung genannten Variable hat, schreiben Sie 'ja', ansonsten 'nein'.

Fortsetzung nächste Seite!

- b) Nehmen Sie nun an, dass Klasse Aa aus obiger Tabelle private Sichtbarkeit hätte. Beschreiben Sie, inwiefern sich für Klassen aus Paket B der Zugriff auf die Variablen damit ändern würde.

Aufgabe 7**(12 Punkte)**

- a) Erklären Sie in einem Satz, was Kapselung im objektorientierten Sinne bedeutet.
- b) Ist vereinfachte Wartung ein Vorteil von Kapselung? Begründen Sie in max. 2 Sätzen.
- c) Ist reduzierte Komplexität ein Vorteil von Kapselung? Begründen Sie in max. 2 Sätzen.
- d) Ist erhöhte Sicherheit (Safety) ein Vorteil von Kapselung? Begründen Sie in max. 2 Sätzen.

Aufgabe 8**(14 Punkte)**

Teilaufgaben beziehen sich auf das Strukturmuster Adapter.

- a) Erläutern Sie den Zweck (Intent) des Strukturmusters Adapter in max. 3 Sätzen, ohne dabei auf die technische Umsetzung einzugehen.
- b) Erklären Sie die zwei Varianten (objektorientiert, klassenbasiert), das Strukturmuster Adapter umzusetzen. Schreiben Sie pro Variante max. 5 Sätze. Benennen Sie hierbei die verwendeten Klassen, z. B. "A" für Adapterklasse.
- c) Nennen Sie einen Vorteil und einen Nachteil des Strukturmusters Adapter im Vergleich zu einer Implementierung, die nicht auf dieses Muster zurückgreift.

Fortsetzung nächste Seite!

Aufgabe 9**(8 Punkte)**

Skizzieren Sie den Softwarelebenszyklus des traditionellen (nicht-agilen) Softwarelebenszyklus (Software Development Life Cycle).

Aufgabe 10**(15 Punkte)**

- a) Geben Sie zwei traditionelle (“plan-getriebene”) Methoden der Softwareentwicklung an.
- b) Nennen Sie zwei Beispiele für eine agile Methode der Softwareentwicklung.
- c) Was unterscheidet die agilen Methoden von traditionellen Methoden? Antworten Sie in drei unterschiedlichen Paaren (also in Gegensätzen zwischen den beiden Vorgehensmodellen, z. B. “Flexibilität im Vorgehen **statt** lange, umfangreiche Vorausplanung”).

Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanken**Aufgabe 1 (Vermischte Fragen)****(24 Punkte)**

- a) Erläutern Sie die Begriffe Before und After Image. Erklären Sie deren Bedeutung für die Umsetzung des ACID-Prinzips.
- b) Erläutern Sie Backward und Forward Recovery und grenzen Sie diese voneinander ab.
- c) Beschreiben Sie ausführlich Funktionsweise und Aufgabe des Zwei-Phasen-Sperr-Protokolls.
- d) Definieren Sie die Begriffe Entität und Entitätstyp.
- e) Lässt sich jede tertiäre Beziehung durch drei binäre Beziehungen ausdrücken? Falls ja, begründen Sie dies. Falls nein, geben Sie ein Gegenbeispiel an.
- f) Erklären Sie, wie sich ein ER-Modell in ein relationales Modell überführen lässt.
- g) Beschreiben Sie, was man unter konkurrierenden Operationen versteht. Geben Sie ein Beispiel an.

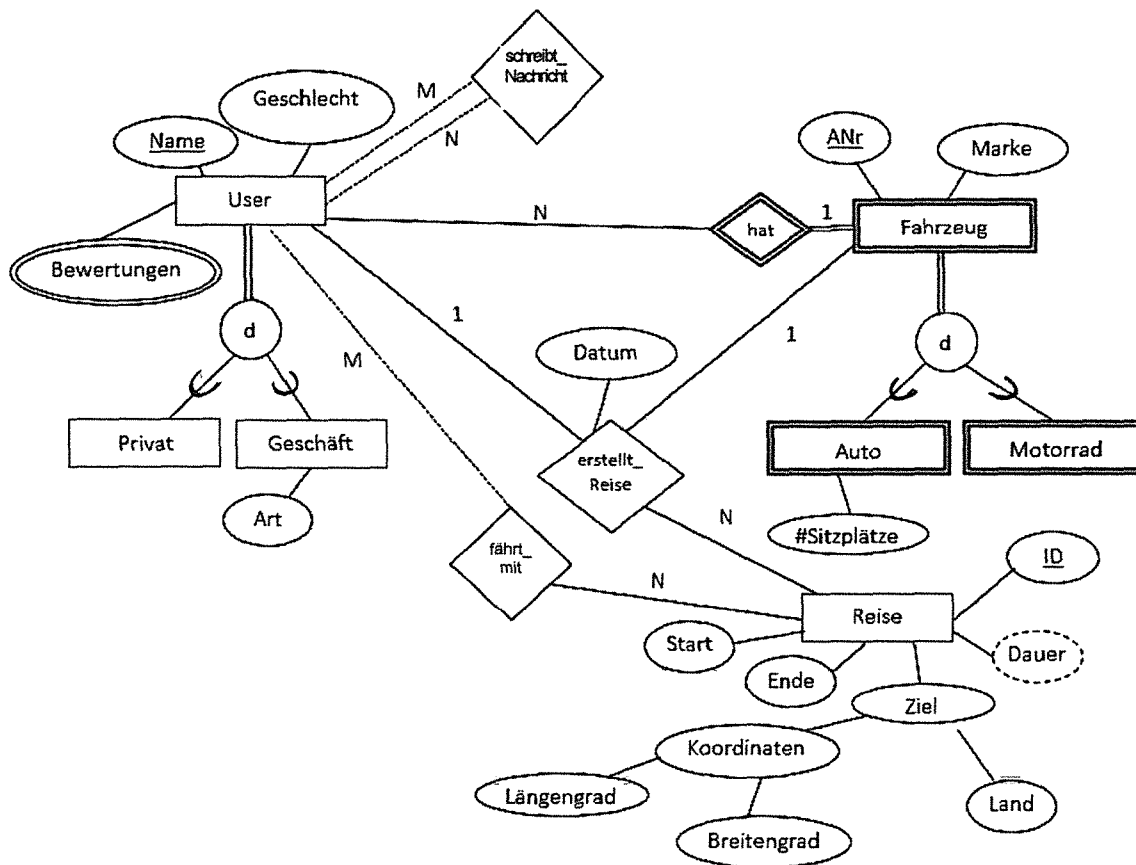
Aufgabe 2**(22 Punkte)**

Überführen Sie das im Folgenden gegebene ER-Modell in ein normalisiertes Relationenmodell. Vergessen Sie dabei nicht, Primär- und Fremdschlüssel entsprechend zu kennzeichnen. Geben Sie jeweils an, auf welche Spalte sich der Fremdschlüssel bezieht.

Erklärung der Notation:

- Teilmengen-Symbol: Stellt eine Vererbungsbeziehung dar, beispielsweise sind Privat und Geschäft Unterklassen von User.
- Das d bedeutet, dass die Vererbung disjunkt ist, d. h. ein Element der Oberklasse kann nicht zu mehreren Unterklassen gehören.
- Der Doppelstrich bei der Vererbung bedeutet, dass die Vererbung total ist, d. h. ein Element der Unterklasse ist zwingenderweise auch ein Element einer Oberklasse.
- Die doppelte Umrandung einer Entität und der dazugehörigen Relationship eine schwache Entität dar, d. h. diese Entität kann nicht ohne eine starke Entität existieren (Beispiel: Fahrzeug kann nicht ohne User existieren).
- Gestrichelt umrandete Attribute bezeichnen abgeleitete Attribute.
- Doppelt umrandete Attribute bezeichnen mehrwertige Attribute.
- Die hierarchische Struktur bei den Attributen stellen komposite Attribute dar.

Fortsetzung nächste Seite!

**Aufgabe 3 (SQL-Abfragen)****(24 Punkte)**

Gegeben sind folgende Relationenschemata. Die Primärschlüssel der einzelnen Tabellen sind jeweils durch Unterstrichungen gekennzeichnet.

```

Interpret (Interpret-ID, Name, Alter, Nationalität)
Lied (Titel, Laenge, Interpret-ID)
      FK Interpret-ID REF Interpret
CD (ID, Name, Jahr)
Lied_auf_CD (ID, Titel, Position)
      FK ID REF CD
      FK Titel REF Lied

```

- Schreiben Sie eine SQL-Anweisung, die die Namen aller Lieder des Künstlers „Luis Fonsi“ ausgibt. Die Ausgabe soll nach der Länge absteigend geordnet erfolgen.
- Schreiben Sie eine SQL-Anweisung, welche die ID aller CDs ausgibt, auf denen mindestens ein Interpret vertreten ist, der jünger als 16 Jahre ist.
- Formulieren Sie einen Ausdruck in relationaler Algebra, der die Namen aller Interpreten ausgibt, die mindestens einen Titel auf der CD mit der ID 15472 besitzen und jünger als 25 Jahre alt sind.
- Erklären Sie in Worten, was folgender Ausdruck in relationaler Algebra angibt:

$$\pi_{Name}(\sigma_{Alter \leq 20}(Interpret) \bowtie_{Interpret.Interpret_ID=Lied.Interpret_ID} \sigma_{Länge > 5}(Lied))$$

Fortsetzung nächste Seite!

Aufgabe 4 (Normalisierung)**(9 Punkte)**

Im Nachfolgenden ist ein Relationenschema in erster Normalform gegeben. In diesem wird der Sachverhalt wiedergegeben, dass ein Kunde (beschrieben durch KundenNr und KundenName) eine oder mehrere Versicherungsverträge (beschrieben durch VNr und VArt) bei einer Versicherungsgesellschaft (beschrieben durch VName und VSitz) hat.

Gegeben sind die folgenden funktionalen Abhängigkeiten, die in der Menge FD zusammengefasst werden:

- KundenNr \rightarrow KundenName
- VNr \rightarrow VArt
- VNr VSitz \rightarrow KundenNr
- KundenNr VNr \rightarrow VName
- VSitz \rightarrow VName

Die Menge FD ist bereits minimal.

- a) Bestimmen Sie alle Schlüsselkandidaten der Relation Kunde unter Verwendung von FD.
- b) Begründen Sie, weshalb das angegebene Schema Kunde nicht in dritter Normalform ist.
- c) Überführen Sie das Schema Kunde in die dritte Normalform! Benennen Sie den Namen des verwendeten Algorithmus.

Aufgabe 5 (Transaktionsmanagement)**(11 Punkte)**

- a) Zeichnen Sie den zugehörigen Serialisierungsgraph (es ist nicht notwendig, die Kanten des Graphen zu beschriften) zu folgender Historie:

$r_1(A) \ w_2(B) \ w_1(A) \ r_3(B) \ w_4(C) \ r_3(D) \ w_3(B) \ r_1(C) \ r_3(B) \ w_2(C) \ r_3(A) \ c_1 \ r_2(C) \ w_4(D) \ c_4 \ c_2 \ w_3(C) \ c_3$

- b) Formulieren Sie das Serialisierungstheorem.
- c) Ist die Historie aus Teilaufgabe a) serialisierbar? Begründen Sie kurz Ihre Antwort!
- d) Erläutern Sie das ACID Prinzip.

Thema Nr. 2**Teilaufgabe I: Softwaretechnologie****Aufgabe 1 (Projektmanagement)****(10 Punkte)**

Gegeben seien folgende Aufgaben mit ihren jeweiligen Abhängigkeiten und ihrer jeweiligen Dauer:

Aufgabe	Abhängigkeiten	Dauer (Tage)
A	/	8
B	A	4
C	B	3
D	A	7
E	D	2
F	A, C, E	10

- Zeichnen Sie ein CPM-Diagramm basierend auf der gegebenen Aufgabenliste. Zeichnen Sie hierbei auch den Start- und Endknoten ein.
- Als *Slack* bezeichnet man die Zeit, um die eine Aufgabe bezüglich ihres frühesten Startzeitpunktes verzögert werden kann, ohne dass es Probleme bei der fristgerechten Fertigstellung des Projektes gibt. Berechnen Sie den Slack für alle Aktivitäten und tragen Sie die jeweiligen Werte in Ihr Diagramm ein.
- Zeichnen Sie den kritischen Pfad ins Diagramm ein oder geben Sie die Aufgaben des kritischen Pfades in der folgenden Form an: **Start** → ... → **Ende**. Sollte es mehrere kritische Pfade geben, geben Sie auch diese an. Geben Sie die Dauer des kritischen Pfades an.

Aufgabe 2 (Entwicklungsprozesse)**(14 Punkte)**

- Zeichnen Sie das Wasserfallmodell und beschreiben Sie die einzelnen Phasen in kurzen Sätzen.
- Nennen und erklären Sie je zwei Vor- und Nachteile des Wasserfallmodells.

Aufgabe 3 (Anforderungsanalyse)**(15 Punkte)**

Erstellen Sie ein Use-Case-Diagramm zur Modellierung einer Zeitungsredaktion, die aus Redakteuren, Chefredakteuren und Lesern besteht. Dabei sind folgende Angaben zu beachten:

/F010/ Ein Leser kann Zeitungsartikel kaufen.

/F020/ Leser, Redakteure und Chefredakteure können Zeitungsartikel lesen.

/F030/ Redakteure und Chefredakteure können Artikel verfassen.

/F040/ Chefredakteure können Artikel freigeben.

/F050/ Für alle Aktionen muss der Nutzer authentifiziert werden.

Hinweis: Verwenden Sie, falls sinnvoll, Vererbung zwischen den Akteuren.

Fortsetzung nächste Seite!

Aufgabe 4 (Entwurfsmuster)**(15 Punkte)**

- a) Erstellen Sie ein Klassendiagramm, das verschiedene Strategien einen Baum zu travestieren, modelliert. Hierbei sollen genau die bekannten Strategien *Preorder* und *Postorder* angeboten werden. Der Rückgabebetyp dieser Operationen ist `void`. Sie dürfen annehmen, dass bereits eine entsprechende Datenstruktur für den Baum namens `Tree` vorhanden ist. Verwenden Sie für die Realisierung dieses Klassendiagramms das Entwurfsmuster *Strategy-Pattern*. Die Klasse `Serialization` (das Kontext-Objekt) kümmert sich um die Serialisierung des Baumes je nach ausgewählter Strategie. Die Auswahl der Strategie soll hierbei über eine `get`- und `set`-Methode geregelt werden. Außerdem sollten Sie in dieser Klasse mit einem Attribut die Baum-Instanz verwalten sowie eine Methode für die Ausführung des Algorithmus bereitstellen.

Hinweise:

- Die `get`- und `set`-Methode für die Baum-Instanz müssen nicht eingezeichnet werden.
 - Sie müssen keine Konstruktoren angeben.
- b) Geben Sie an, zu welcher Gruppe von Mustern das Entwurfsmuster *Strategy-Pattern* gehört.
- c) Nennen Sie einen Vorteil des *Strategy-Patterns*.

Aufgabe 5 (Implementierung)**(21 Punkte)**

Gegeben ist der jeweilige Java-Programmcode der Klassen `Verwaltung`, `Gehege` und `Tier`. Erstellen Sie unter Beachtung der Hinweise auf der nächsten Seite ein Sequenzdiagramm zur Methode `ausfuehren` der Klasse `Verwaltung`.

```
1 public class Verwaltung {
2
3     public static void ausfuehren() {
4         Gehege g1 = new Gehege();
5         Gehege g2 = new Gehege();
6
7         // ab hier soll modelliert werden
8         Tier t1 = new Tier();
9         t1.setzeTierart("Zebra");
10        t1.setzeNamen("Daisy");
11        Tier t2 = new Tier();
12        t2.setzeTierart("Nilpferd");
13        t2.setzeNamen ("Hippo");
14
15        g1.fuegeTierHinzu(t1);
16
17        if (g1.besitztTierVonArt("Zebra")) {
18            g1.fuegeTierHinzu(t2);
19        } else {
20            g2.fuegeTierHinzu(t2);
21        }
22    }
23
24 }
```

Fortsetzung nächste Seite!

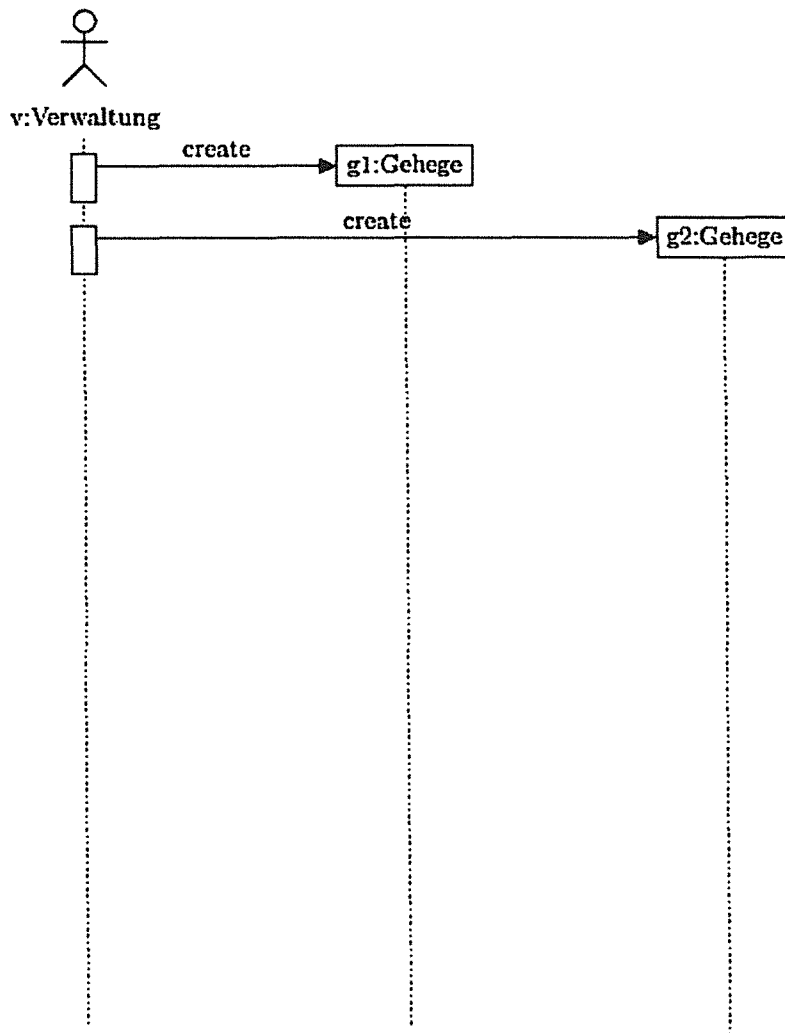
```
1 public class Gehege {
2
3     private List<Tier> enthalteneTiere = new ArrayList<>();
4
5     public void fuegeTierHinzu(Tier tier) {
6         enthalteneTiere.add(tier);
7     }
8
9     public boolean besitztTierVonArt(String art) {
10         for (Tier tier : enthalteneTiere) {
11             if (tier.nenneArt().equals(art)) {
12                 return true;
13             }
14         }
15         return false;
16     }
17
18 }
```

```
1 public class Tier {
2     private String tierart;
3     private String name;
4
5     public void setzeTierart(String tierart) {
6         this.tierart = tierart;
7     }
8
9     public void setzeNamen(String name) {
10         this.name = name;
11     }
12
13     public String nenneArt() {
14         return this.tierart;
15     }
16 }
```

Hinweise:

- Die Operationen auf Listen oder Strings müssen nicht gesondert dargestellt werden.
- Auf das Zeichnen einer passiven Lebenslinie muss nicht geachtet werden.
- Übertragen Sie das Diagramm von der nächsten Seite auf Ihren Bearbeitungsbogen und ergänzen Sie dieses im Folgenden.
- Lösen Sie if-Abfragen auf, sofern es möglich ist.
- Schleifen müssen nicht gesondert dargestellt werden, sondern betrachten Sie, falls notwendig, die konkrete Ausführung.
- Einzelne Rollen müssen nicht gesondert dargestellt werden.

Fortsetzung nächste Seite!

**Aufgabe 6 (Testing)****(15 Punkte)**

- Erklären Sie in wenigen Sätzen den Unterschied zwischen Fehlervermeidung und Fehlertoleranz.
- Nennen und beschreiben Sie kurz die vier verschiedenen Testebenen.
- Beschreiben Sie zwei Unterschiede zwischen Black-Box-Testing und White-Box-Testing. Nennen Sie sowohl zum Black-Box- als auch zum White-Box-Testing zwei Vorteile.

Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanken**Aufgabe 1 (Vermischte Fragen)****(14 Punkte)**

Beantworten Sie die folgenden Fragen und begründen bzw. erläutern Sie Ihre Antwort in jeweils ein bis zwei Sätzen.

- a) Welches SQL-Schlüsselwort kann verwendet werden, um sicherzustellen, dass jedes Tupel nur einmal im Ergebnis auftaucht?
- b) Welcher Unterschied besteht zwischen ALTER TABLE und UPDATE?
- c) Welche Auswirkung hat die Verwendung von ON DELETE RESTRICT bei einem Fremdschlüsselattribut?
- d) Kann eine abgeschlossene (*committed*) Transaktion wieder rückgängig gemacht werden? Warum (nicht)?
- e) Was versteht man unter einem *prepared statement* im Kontext einer Programmierschnittstelle für relationale Datenbanken (z. B. JDBC)?
- f) Worin besteht der Unterschied zwischen dem Zustand, dass ein Attribut vom Typ Integer 0 enthält, und dem Zustand, dass es NULL enthält?
- g) Kann ein Fremdschlüssel auch auf ein anderes Attribut als den Primärschlüssel der referenzierten Tabelle zeigen?

Aufgabe 2 (ER-Modellierung)**(23 Punkte)**

Im Folgenden finden Sie die Beschreibung einer Software zur Notenverwaltung. 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. Kennzeichnen Sie auch die totale Teilnahme von Entity-Typen an Beziehungstypen.

Es gibt Studenten. Diese haben einen Namen und eine eindeutige Matrikelnummer. Daneben gibt es Prüfungen, die über eine eindeutige Prüfungsnummer sowie über eine Prüfungsart (z. B. mündlich oder schriftlich) verfügen. Dozenten haben einen eindeutigen Namen, einen Titel und sind einem Lehrstuhl zugeordnet. Jeder Student ist in mindestens einem Studiengang eingeschrieben. Ein Studiengang hat eine eindeutige Bezeichnung.

Studenten können Prüfungen in einem bestimmten Semester ablegen und erhalten dafür eine Note. Ein Semester hat ein eindeutiges Kürzel. In jedem Semester finden Prüfungen statt. Jede Prüfung gehört zu genau einer Vorlesung. Eine Vorlesung hat einen eindeutigen Namen und wird in mindestens einem Semester von mindestens einem Dozenten angeboten. Ein Dozent bietet mindestens eine Vorlesung an. Es kann mehrere Prüfungen für eine Vorlesung geben (bspw. Klausuren in unterschiedlichem Umfang), aber jede Prüfung gehört nur zu einer Vorlesung.

Fortsetzung nächste Seite!

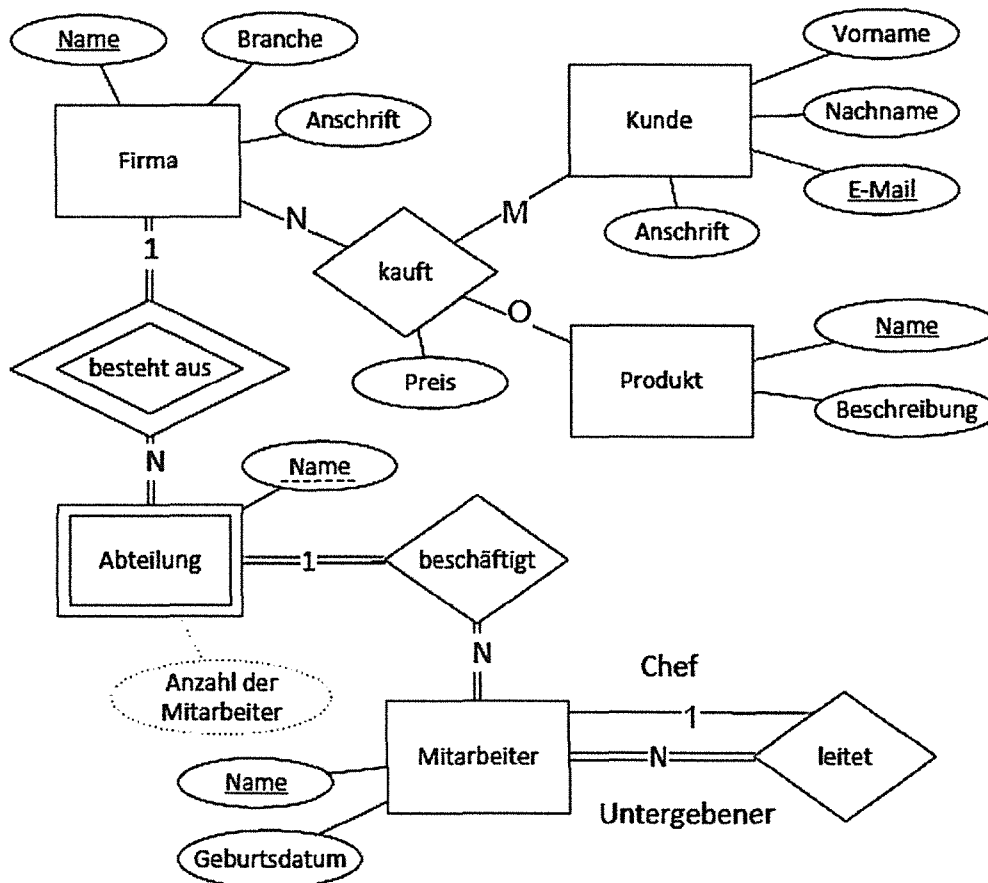
Aufgabe 3 (Relationaler Entwurf)**(19 Punkte)**

Entwerfen Sie zum untenstehenden ER-Diagramm ein Relationenschema in dritter Normalform (3. NF) mit möglichst wenigen Relationen.

Verwenden Sie dabei folgende Notation: Primärschlüssel werden durch Unterstreichen gekennzeichnet, Fremdschlüssel durch die Nennung der Relation, auf die sie verweisen, in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert. Wenn ein Attribut zur korrekten Abbildung des ER-Diagramms als UNIQUE oder NOT NULL ausgezeichnet werden muss, geben Sie dies an.

Beispiel:

```
Relation1 (Primärschlüssel, Attribut1, Attribut2,
Fremdschlüsselattribut1[Relation1],
(Fremdschlüssel2_Attribut1, Fremdschlüssel2_Attribut2)[Relation2]);
Attribut1 UNIQUE
Attribut2 NOT NULL
```



Fortsetzung nächste Seite!

Aufgabe 4 (SQL)**(25 Punkte)**

Gegeben sind folgende Relationen:

```
Verein(Name, Ort)
Athlet(ID, Verein[Verein], Vorname, Nachname)
Disziplin(Name, Beschreibung)
Wettbewerb(ID, Name, Ortsname[Ort]); Ortsname NOT NULL
Ort(Name, Land, Adresse)
Teilnahme(AthletID[Athlet], WettbewerbID[Wettbewerb], Name [Disziplin],
Ergebnis)
```

Verwenden Sie im Folgenden nur Standard-SQL und keine produktspezifischen Erweiterungen. Sie dürfen bei Bedarf Views anlegen. Geben Sie einen Datensatz nicht mehrfach aus.

- Schreiben Sie eine SQL-Anweisung, die die Tabelle *Wettbewerb* anlegt. Gehen Sie davon aus, dass alle anderen Tabellen bereits existieren.
- Schreiben Sie eine SQL-Anweisung, die Vorname und Nachname aller Athleten ausgibt, die zu einem Verein aus Erlangen gehören, aufsteigend sortiert nach Nachname und bei gleichen Nachnamen, absteigend sortiert nach Vorname.
- Schreiben Sie eine SQL-Anweisung, die für jede Disziplin das durchschnittliche Ergebnis ermittelt.
- Schreiben Sie eine SQL-Anweisung, welche alle Athleten löscht, die noch nie an einem Wettbewerb teilgenommen haben.
- Schreiben Sie eine SQL-Anweisung, welche die Athleten ausgibt, die an den meisten *unterschiedlichen* Wettbewerben bisher teilgenommen haben. Geben Sie zu jedem Athleten die Attribute AthletID, Vorname und Nachname an sowie die Anzahl der Wettbewerbe, an denen er teilgenommen hat.

Aufgabe 5 (Normalformen)**(9 Punkte)**

Gegeben ist die Relation $R(a_1, a_2, a_3, a_4, a_5)$. Sie besitzt den Schlüsselkandidaten $\{a_1, a_2\}$ und die funktionalen Abhängigkeiten $a_2 \rightarrow a_3$ und $a_4 \rightarrow a_5$.

- Begründen Sie in ein bis zwei Sätzen, warum R sich nicht in zweiter Normalform befindet. Zerlegen Sie R so, dass die Ergebnisrelationen die zweite Normalform erfüllen. Erzeugen Sie dabei so wenige Relationen wie möglich.
- Begründen Sie in ein bis zwei Sätzen, warum sich die Ergebnisrelationen der vorherigen Teilaufgabe nicht alle in der dritten Normalform befinden. Zerlegen Sie sie so weiter, dass alle Relationen die dritte Normalform erfüllen. Erzeugen Sie dabei so wenige Relationen wie möglich.