
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Herbst
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: 14

Bitte wenden!

Thema Nr. 1 (Aufabengruppe)

Es sind alle Aufgaben dieser Aufabengruppe zu bearbeiten!

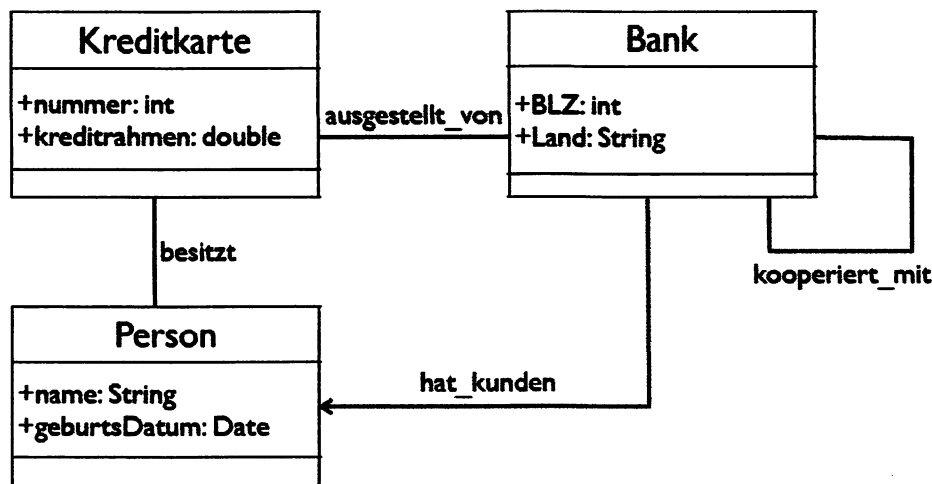
Teilaufgabe I: Softwaretechnologie

Aufgabe 1: Grundwissen Softwaretechnik

1. Welche zwei Eigenschaften unterscheiden die Komposition von der Aggregation?
2. Nennen Sie drei Merkmale, die Schnittstellen (*interface*) und abstrakte Klassen (*abstract class*) voneinander unterscheiden!
3. Geben Sie die Laufzeit für die lineare Suche an. Nennen Sie ein effizienteres Verfahren unter der Annahme, dass die Eingabedaten sortiert sind. Was ist dessen Laufzeit?
4. Nennen Sie je einen Vorteil für verkettete Listen und für Arrays!

Aufgabe 2: UML

Betrachten Sie das folgende UML-Diagramm:



1. Geben Sie zu jeder Beziehung die Multiplizitäten der Beziehungen an. (Nicht ins Diagramm eintragen!)
2. Setzen Sie Person, Kreditkarte und Bank mit Attributen und Beziehungen in einer objektorientierten Programmiersprache Ihrer Wahl um. Sie können Typen aus der jeweiligen Standard-Bibliothek verwenden (zum Beispiel: Set). Methoden sind nicht verlangt.

Fortsetzung nächste Seite!

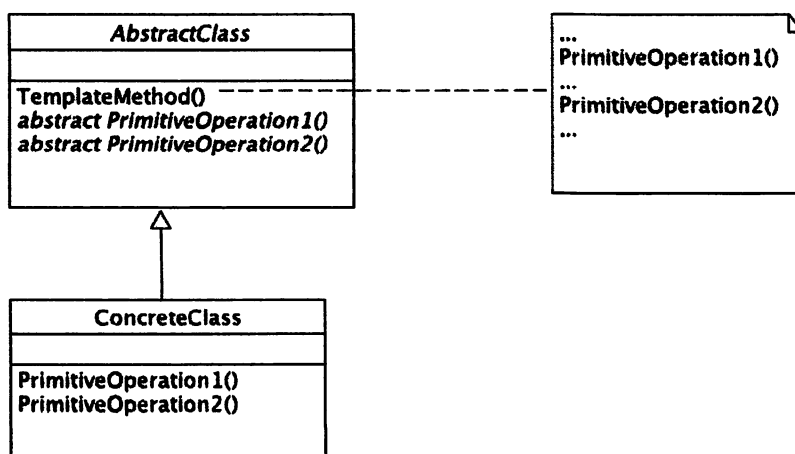
Aufgabe 3: Systementwicklung

Sie wurden beauftragt, ein neues Bibliotheksverwaltungssystem zu entwickeln. Dieses System soll den Bücherbestand verwalten sowie Ausleih- und Vormerkungsfunktionalität bieten.

Rahmendaten zum Projekt: Von Seiten des Auftraggebers wird großer Wert auf besonders hohe Sicherheitseigenschaften des Systems gelegt. Im Rahmen des Projekts sollen von Anfang an Prototypen den Projektfortschritt dokumentieren. Zur Entwicklung dieses Systems steht Ihnen ein Team von sechs Entwicklern zur Verfügung. Das System muss innerhalb von 8 Monaten fertig gestellt sein.

1. Nennen Sie drei Stakeholder des Bibliotheksverwaltungssystems und jeweils zwei Anwendungsfälle!
2. Welches der Vorgehensmodelle V-Modell XT, Spiralmodell und Extreme Programming würden Sie unter den geschilderten Umständen für die Entwicklung dieses Systems anwenden? Treffen Sie eine eindeutige Entscheidung und begründen Sie diese mit drei schlüssigen Argumenten!
3. Das zu entwickelnde Softwaresystem sollte die Qualitätseigenschaften Benutzbarkeit, Zuverlässigkeit und Wartbarkeit erfüllen. Geben Sie zu jeder dieser Qualitätseigenschaften je zwei Qualitätssicherungsmaßnahmen an.
4. Das von Ihnen zu entwickelnde System soll ein derzeit in Betrieb befindliches System ablösen. Das bisherige System verwendet ein Chipkartensystem zur Identifikation der Bibliotheksbenutzer. Im Zuge der Systemumstellung soll ein neues Kartensystem eingeführt werden. Um nicht sofort alle alten Karten ersetzen zu müssen, soll das neue Bibliotheksverwaltungssystem die alten Karten weiterhin unterstützen. Zeichnen Sie ein Klassendiagramm, um unter Verwendung des Patterns TemplateMethod (s.u.) das konkrete Einlesen einer Karte mit beiden Systemen zu ermöglichen.

Intention des TemplateMethod-Patterns: Definiere das Gerüst eines Algorithmus in einer Methode wobei die Implementierung einiger Schritte in Unterklassen ausgelagert wird. Template-Method lässt die Unterklassen bestimmte Schritte in einem Algorithmus redefinieren, ohne die Struktur des Algorithmus zu verändern.



Fortsetzung nächste Seite!

Aufgabe 4: Qualitätssicherung

Ein gängiger Ansatz zur Messung der Qualität von Software ist das automatisierte Testen von Programmen. Im Folgenden werden praktische Testmethoden anhand des nachstehend angegebenen Sortieralgorithmus diskutiert.

Algorithmus 1 Bubble Sort

```
void bubblesort(int[] array, int len) {  
1   for (int i = 0 ; i < len - 1 ; i++) {  
2       for (int j = 0 ; j < len - 1 ; j++) {  
3           if ( array[j] > array[j + 1] ) {  
4               int temp = array[j];  
5               array[j] = array[j + 1];  
6               array[j + 1] = temp;  
           }  
       }  
   }  
}
```

- Nennen Sie eine Art des Black-Box-Testens und beschreiben Sie deren Durchführung anhand des vorgegebenen Algorithmus.
- Zeichnen Sie ein mit Zeilennummern beschriftetes Kontrollflussdiagramm für den oben angegebenen Sortieralgorithmus.
- Erklären Sie, ob eine vollständige Pfadüberdeckung für die gegebene Funktion möglich und sinnvoll ist.

Teilaufgabe II: Datenbanksysteme**Aufgabe 1: Grundwissen Datenbanksysteme**

- Nennen Sie den Namen **des Algorithmus**, der ein Relationenschema in einer Normalform in eine höhere Normalform überführt und dabei **abhängigkeitsbewahrend** ist.
- Nennen Sie drei Gründe für den Einsatz eines Datenbanksystems. (3 Stichworte)
 - _____
 - _____
 - _____
- Geben Sie außerdem drei Gründe an für den Einsatz eines Datenbanksystem **für die Datenverwaltung**.
- Geben Sie die Laufzeit von Einfüge- und Löschoptionen in erweiterbaren Hashtabellen an. (Stichwort)
- Nennen Sie eine Indexstruktur, die sich anbietet, wenn viele Bereichsanfragen auf der zu indizierenden Spalte (Range Queries) erwartet werden. (Stichwort)

Fortsetzung nächste Seite!

Aufgabe 3: Relationale Anfragesprachen

Gegeben ist folgendes Universitätsschema (eine beispielhafte Ausprägung hängt der Klausur an):

Studenten : {[MatrNr : integer, Name : string, Semester : integer]}

Vorlesungen : {[VorlNr : integer, Titel : string, SWS : integer, gelesenVon : integer]}

Professoren : {[PersNr : integer, Name : string, Rang : string, Raum : integer]}

Assistenten : {[PersNr : integer, Name : string, Fachgebiet : string, Boss : integer]}

hören : {[MatrNr : integer, VorlNr : integer]}

voraussetzen : {[Vorgänger : integer, Nachfolger : integer]}

prüfen : {[MatrNr : integer, VorlNr : integer, PersNr : integer, Note : integer]}

1. Finden Sie alle Studenten (**nur Name** ausgeben), die in keinem Fach geprüft wurden. Formulieren Sie die Anfrage in **relationaler Algebra**.
2. Finden Sie alle Studenten (**nur Name** zurückgeben), die in mindestens zwei verschiedenen Fächern bestanden (Note ≤ 4.0) haben! Formulieren Sie die Anfrage in **Tupelkalkül**.

Aufgabe 4: SQL

Gegeben sei das folgende Universitätsschema (eine beispielhafte Ausprägung hängt der Klausur an):

Studenten : {[MatrNr : integer, Name : string, Semester : integer]}

Vorlesungen : {[VorlNr : integer, Titel : string, SWS : integer, gelesenVon : integer]}

Professoren : {[PersNr : integer, Name : string, Rang : string, Raum : integer]}

Assistenten : {[PersNr : integer, Name : string, Fachgebiet : string, Boss : integer]}

hören : {[MatrNr : integer, VorlNr : integer]}

voraussetzen : {[Vorgänger : integer, Nachfolger : integer]}

prüfen : {[MatrNr : integer, VorlNr : integer, PersNr : integer, Note : integer]}

Geben Sie jeweils ein *SQL*-92 Statement an, welches die gestellte Frage beantwortet.

1. Wie viele Vorlesungen hört der Student mit dem Namen Fichte? (Ausgabe der Anfrage soll die Anzahl der Vorlesungen sein.)
2. Welche Studenten außer Fichte haben eine Vorlesung gehört, die auch Fichte gehört hat? Geben Sie die Namen dieser Studenten (und nicht Fichte) duplikatfrei aus. (Im Beispiel sind dies Feuerbach, Theophrastos und Schopenhauer.)
3. Finden Sie alle Studenten (**MatrNr und Name duplikatfrei ausgeben**), die in jeder Prüfung, die sie geschrieben haben, eine 1 erhalten haben. Geben Sie hierbei nur Studenten aus, die mindestens in einem Fach geprüft wurden.

Fortsetzung nächste Seite!

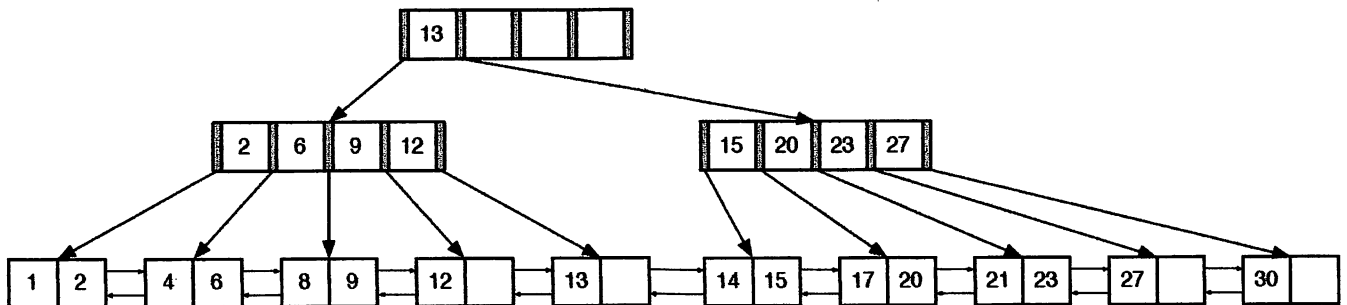
4. Geben Sie eine absteigend sortierte Liste aus allen Vorlesungen und der Anzahl pro Vorlesung abgehaltener Prüfungen aus, etwa so:

VorlNr	Anzahl
4630	2
5001	1
5041	1
5022	0
5043	0
⋮	⋮

Formulieren Sie Ihre Lösung auf zwei verschiedene Varianten: Einmal mit korrelierter Unterabfrage und einmal ohne.

- a) Variante 1: mit korrelierter Unterabfrage
- b) Variante 2: ohne korrelierter Unterabfrage

Aufgabe 5: Physische Datenorganisation



1. Wie heißt die gezeigte Datenstruktur?
2. Geben Sie den Wert von k (minimale Belegung von inneren Knoten) für das Beispiel an.
3. Geben Sie den Wert von k^* (minimale Belegung von Blättern) für das Beispiel an.
4. Fügen Sie die 5 in die Datenstruktur ein. Zeichnen Sie die vollständige Datenstruktur nach dem Einfügen und allen möglicherweise notwendigen Ausgleichsoperationen.

Beispielausprägung

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesungen			
VorlNr	Titel	SWS	gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

Assistenten			
PersNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2134

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2
25403	4630	2137	5

Abbildung 1: Beispielausprägung für eine Universitäts-Datenbank

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1: Projektmanagement

Gegeben ist die folgende **Tabelle** zur Grobplanung eines hypothetischen Softwareprojekts:

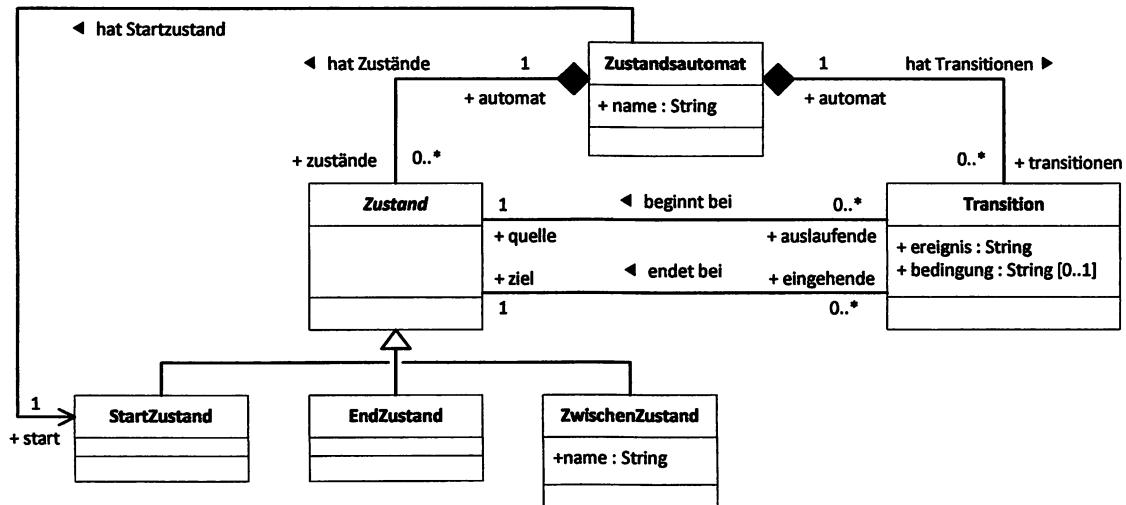
Aktivität	Dauer	Einschränkungen
<i>Anforderungs-analyse</i>	4 Monate	Endet frühestens zwei Monate nach dem Start der Entwurfsphase.
<i>Entwurf</i>	3 Monate	Startet frühestens einen Monat nach dem Start der Anforderungsanalyse.
<i>Implementierung</i>	7 Monate	Endet frühestens drei Monate nach dem Ende der Entwurfsphase. Darf erst starten, nachdem die Anforderungsanalyse abgeschlossen ist.

- a) Geben Sie ein **CPM-Netzwerk** an, das die Aktivitäten und Abhängigkeiten des obigen Projektplans beschreibt. Gehen Sie von der Zeiteinheit „Monate“ aus.
- Das Projekt hat einen Start- und einen Endknoten.
 - Jede Aktivität wird auf einen Start- und einen Endknoten abgebildet.
 - Die Dauer der Aktivitäten sowie Abhängigkeiten sollen durch Kanten dargestellt werden.
 - Der Start jeder Aktivität hängt vom Projektstart ab, das Projektende hängt vom Ende aller Aktivitäten ab. Modellieren Sie diese Abhängigkeiten durch Pseudoaktivitäten mit Dauer null.
- b) Berechnen Sie für jedes Ereignis (d.h. für jeden Knoten) die **früheste Zeit**, die **späteste Zeit** sowie die **Pufferzeit**. Beachten Sie, dass die Berechnungsreihenfolge einer topologischen Sortierung des Netzwerks entsprechen sollte.
- c) Geben Sie einen **kritischen Pfad** durch das CPM-Netzwerk an. Welche **Aktivität** darf sich demnach wie lange verzögern?

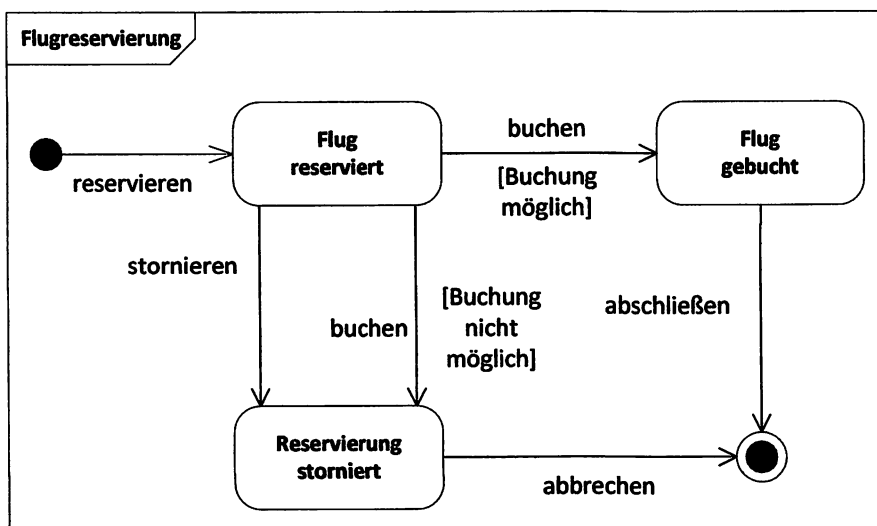
Fortsetzung nächste Seite!

Aufgabe 2: Objektorientierte Modellierung mit UML

Gegeben sei das folgende **UML-Klassendiagramm**, welches die statische Struktur von einfachen **Zustandsautomaten** beschreibt:



- a) Geben Sie ein **Glossar** an, welches die im Klassendiagramm definierten Konzepte und Beziehungen natürlicher Sprache beschreibt, ohne auf die grafische Notation einzugehen. Ergänzen Sie die Definitionen gegebenenfalls durch Angaben zur Semantik von Zustandsautomaten. Verwenden Sie ein bis zwei Sätze pro Klasse.
- b) Nachfolgend ist ein Beispiel eines **Zustandsautomaten** in der gängigen grafischen Notation abgebildet. Stellen Sie den Zustandsautomaten als **UML-Objektdiagramm** konform zum obigen **UML-Klassendiagramm** dar. Referenzieren Sie die dort definierten Klassen und Assoziationen; auf Objektbezeichner und Rollennamen dürfen Sie verzichten.



Fortsetzung nächste Seite!

Aufgabe 3: Entwurfsmuster

Verwenden Sie geeignete **Entwurfsmuster**, um die folgenden Sachverhalte mit Hilfe von **UML-Klassendiagrammen** zu beschreiben. Nennen Sie das zu verwendende Entwurfsmuster namentlich, wenden Sie es zur Lösung der jeweiligen *Fragestellung* an und erstellen Sie damit das problemspezifische UML-Klassendiagramm. Beschränken Sie sich dabei auf die statische Sicht, d.h. definieren Sie keinerlei Verhalten mit Ausnahme der Definition geeigneter Operationen.

- a) Es gibt unterschiedliche Arten von Bankkonten: Girokonto, Bausparkonto und Kreditkarte. Bei allen Konten ist der Name des Inhabers hinterlegt. Girokonten haben eine IBAN. Kreditkarten sind immer mit einem Girokonto verknüpft. Bei Bausparkonten werden ein Sparzins sowie ein Darlehenszins festgelegt. Es gibt eine *zentrale* Klasse, die die *Erzeugung* unterschiedlicher Typen von Bankkonten steuert.
- b) Beim Ticker für ein Hockeyspiel können sich verschiedene Geräte registrieren und wieder abmelden, um auf *Veränderungen* des Spielstands zu *reagieren*. Hierzu werden im Ticker die Tore der Heim- und Gastmannschaft sowie die aktuelle Spielminute vermerkt. Als konkrete Geräte sind eine Smartphone-App sowie eine Stadionuhr bekannt.
- c) Dateisysteme sind *baumartig* strukturiert. Verzeichnisse können wiederum selbst Verzeichnisse und/oder Dateien beinhalten. Sowohl Dateien als auch Verzeichnisse haben einen Namen. Das jeweilige Elternverzeichnis ist eindeutig. Bei Dateien wird die Art (Binär, Text oder andere) sowie die Größe in Byte, bei Verzeichnissen die Anzahl enthaltener Dateien hinterlegt.

Teilaufgabe II: Datenbanksysteme

Aufgabe 1: ER-Modellierung

Für nachfolgend gegebene Miniwelt soll ein ER-Modell erstellt werden. Geben Sie Kardinalitäten in Chen-Notation an. Vergessen Sie nicht, dass zur Chen-Notation nicht nur Funktionalitäten, sondern auch Partizipitäten, also die Angabe von Existenzabhängigkeit bzw. totaler Teilnahme, zählen.

Der örtliche Blasmusikverein möchte seine Verwaltung mithilfe einer Datenbank regeln. Zum einen möchte er seine Mitglieder mit Namen, Adresse - bestehend aus Straße, Hausnummer, PLZ und Ort, und Geburtsdatum festhalten. Jedes Mitglied bekommt bei Vereinseintritt eine laufende Nummer zugewiesen. Mitglieder teilen sich komplett auf in aktive Musiker und passive Vereinsmitglieder. Die aktiven Musiker spielen alle mindestens eine Art von Instrument. Ein Instrument hat einen eindeutigen Namen (z. B. Trompete oder Flöte) und einen Typ.

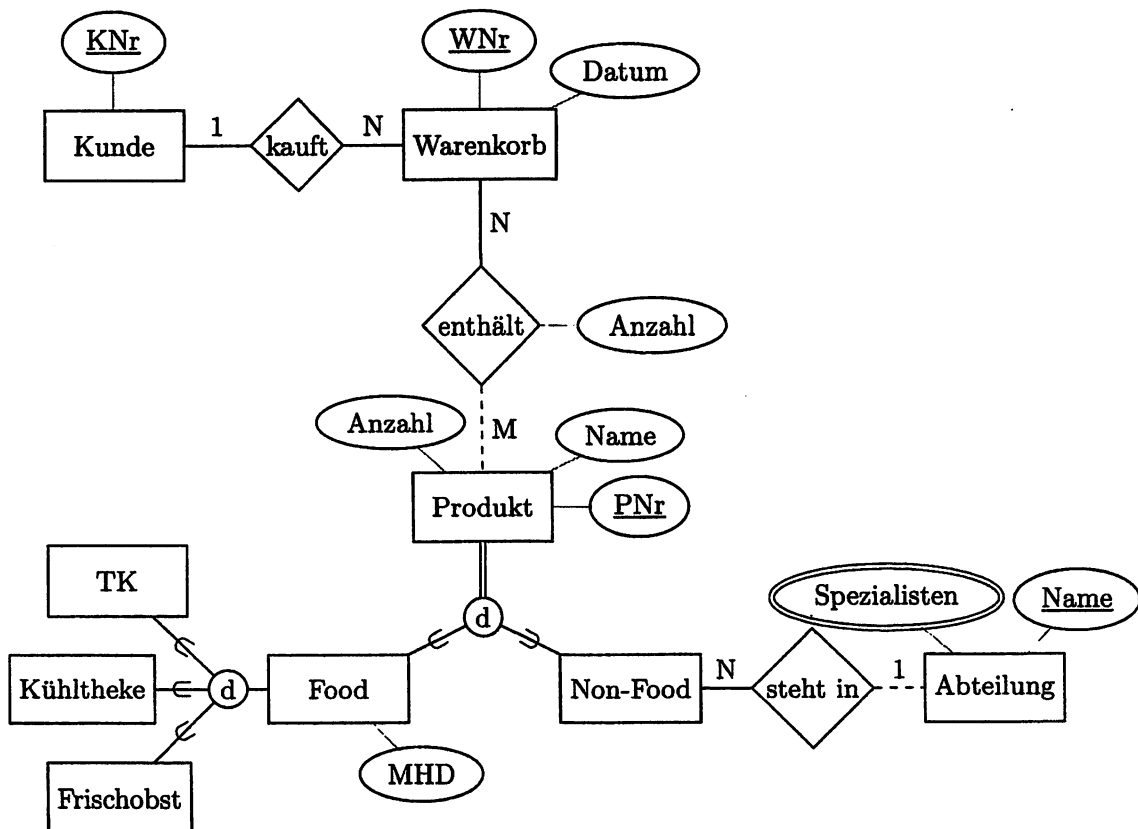
Ein paar der Mitglieder sind außerdem im Vorstand des Vereins. Im Vorstand können sowohl aktive Musiker als auch passive Mitglieder sein. Für die Vorstandsmitglieder wird der jeweilige Posten notiert.

Fortsetzung nächste Seite!

Um die Aktivität der Musiker zu kontrollieren, wird festgehalten, welcher Musiker bei welchem Auftritt anwesend war. Ein Auftritt wird identifiziert über Datum und Uhrzeit, außerdem hat er einen Namen und einen Ort. Während eines Auftritts werden verschiedene Stücke gespielt. Um Wiederholungen in zukünftigen Auftritten am selben Ort zu vermeiden, werden für jeden Auftritt die Stücke, die eine eindeutige Nummer im Musikverein haben, notiert. Musikstücke haben außerdem einen Titel sowie einen oder mehrere Komponisten. Jedes Stück ist für gewisse Instrumente geschrieben. Da sich die Anzahl der Stimmen für Stück und Instrument unterscheiden kann, wird für jede Zuordnung von Musikstück zu Instrument die Anzahl der Stimmen festgehalten.

Aufgabe 2: Relationenmodell

Überführen Sie das folgende ER-Modell in ein Relationenschema, das die Bedingungen der dritten Normalform erfüllt. Geben Sie bei jedem Fremdschlüssel an, worauf er verweist. Überlegen Sie sich bei der Transformation von Generalisierung bzw. Spezialisierung, welche der zur Verfügung stehenden Möglichkeiten die beste ist. Beantworten Sie anschließend noch kurz die Frage, warum Sie sich für die von Ihnen gewählte Methode entschieden haben.



Fortsetzung nächste Seite!

Aufgabe 3: Normalisierung**Gegeben ist die Relation**

Reparatur(KID, Kundenname, Adresse, Geburtsdatum, Autokennzeichen, Modell, Baujahr, MitarbeiterID, Reparaturdatum, Reparaturbeschreibung, Reparaturkosten) .

Sie beschreibt den Sachverhalt, dass Autos, die einem bestimmten Kunden gehören, von einem Mitarbeiter zu einem bestimmten Zeitpunkt repariert werden. Zum Kunden wird eine eindeutige ID (KID), ein Name, eine Adresse und sein Geburtsdatum gespeichert. Das Auto wird über sein Kennzeichen identifiziert und kann eindeutig einem Kunden zugeordnet werden. Zusätzlich wird das Modell und das Baujahr aufgenommen. Für die Identifikation eines Reparaturvorgangs muss das Autokennzeichen und das Reparaturdatum bekannt sein. Dazu können dann die Beschreibung, die Kosten und der Mitarbeiter, der die Reparatur durchgeführt hat, bestimmt werden.

3.1 Funktionale Abhängigkeiten

Bestimmen Sie für die obige Beschreibung die funktionalen Abhängigkeiten. Verwenden Sie hierfür nur die Informationen aus der Beschreibung, wobei Sie Zusammenhänge, die sich aus Transitivität ergeben, nicht angeben müssen.

3.2 Schlüsselkandidaten

Bestimmen Sie die Schlüsselkandidaten für die gegebene Relation mit Hilfe der funktionalen Abhängigkeiten aus Aufgabe 3.1. Begründen Sie Ihre Ergebnisse stichhaltig und nennen Sie die verwendeten Methoden.

3.3 Normalformen

Welche Normalform wird von der gegebenen Relation maximal erfüllt? Begründen Sie Ihre Entscheidung in kurzen Sätzen.

Aufgabe 4: SQL

Für die bayerische Meisterschaft im Turmspringen ist folgendes Datenbankschema angelegt:

Springer (Startnummer, Nachname, Vorname, Geburtsdatum, Körpergröße)

Sprung (SID, Beschreibung, Schwierigkeit)

springt (SID, Startnummer, Durchgang)

FK (SID) referenziert Sprung (SID)

FK (Startnummer) referenziert Springer (Startnummer)

Fortsetzung nächste Seite!

Das Attribut Schwierigkeit kann die Werte 1 bis 10 annehmen, das Attribut Durchgang ist positiv und ganzzahlig. Die Körpergröße der Springer ist in Zentimeter angegeben.

- 4.1 Welche Springer sind größer als 1,80 m? Schreiben Sie eine *SQL*-Anweisung, welche in der Ausgabe mit dem größten Springer beginnt.
- 4.2 Welche Springer haben im ersten Durchgang einen Sprung mit einer Schwierigkeit von unter 6 gezeigt? Schreiben Sie eine *SQL*-Anweisung, welche Startnummer und Nachname dieser Springer ausgibt.
- 4.3 Formulieren Sie in Umgangssprache, aber trotzdem möglichst präzise, wonach mit folgender Abfrage gesucht wird:

```
SELECT springt.Startnummer, s.Nachname, s.Vorname, MAX(springt.Durchgang)
FROM springt, Springer s
WHERE springt.Startnummer=s.Startnummer
GROUP BY springt.Startnummer, s.Nachname, s.Vorname;
```
- 4.4 Gesucht ist die „durchschnittliche Körpergröße“ all der Springer, die vor dem 01.01.2000 geboren wurden. Formulieren Sie eine *SQL*-Anweisung, welche die entsprechenden Springer ausgibt, wobei die Spalte mit der durchschnittlichen Körpergröße genau diesen Namen „durchschnittliche Körpergröße“ haben soll.

Aufgabe 5: Grundlagen

- 5.1 Durch Anwendung der Drei-Schichten-Architektur nach ANSI/SPARC erreicht man Datenunabhängigkeit. Erklären Sie diesen Begriff und die zwei Ausprägungen in der Drei-Schichten-Architektur nach ANSI/SPARC.
- 5.2 Erläutern Sie die ACID-Eigenschaft einer Transaktion. Nennen Sie hierfür, welche Eigenschaften mit dem Begriff „ACID“ zusammengefasst sind und erklären Sie diese Eigenschaften jeweils in ein bis zwei Sätzen!