
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Frühjahr
2015

Kennwort: _____

46116

Arbeitsplatz-Nr.: _____

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: **11**

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe 1: Softwaretechnologie

1. Objektorientierung

- a) Eine Volkshochschule bietet verschiedene Kurse an. Implementieren Sie in einer objektorientierten Programmiersprache Ihrer Wahl eine Klasse Kurs, wobei eine Instanz der Klasse folgende Eigenschaften besitzen soll:

- eine eindeutige Kursnummer
- das Thema des Kurses (z.B. Spanisch, Yoga)
- die Höhe der Kursgebühr
- die maximale Teilnehmerzahl
- die Anzahl der gemeldeten Teilnehmer

Geben Sie auch einen geeigneten Konstruktor an, der auch die eindeutige Kursnummer festlegt. Verwenden Sie dazu einen geeigneten Zähler für die nächste zu vergebende Nummer.

- b) Schreiben Sie eine Methode, die die Auslastung des Kurses, d.h. das Verhältnis zwischen der gemeldeten und möglichen Teilnehmerzahl, berechnet.
- c) Implementieren Sie für die Klasse Kurs eine Methode zur Kursbuchung, die als Übergabeparameter die Anzahl der zu buchenden Kursplätze übergeben bekommen soll. Die Methode soll vor der Buchung zunächst prüfen, ob noch genügend Plätze frei sind. Der Rückgabewert ist der zu zahlende Gesamtbetrag an Kursgebühren, falls die Buchung erfolgreich war.
- d) Schreiben Sie ein Testprogramm, in dem Sie ein Objekt vom Typ Kurs erzeugen, Kursplätze buchen und den Rechnungsbetrag z. B. auf der Standardausgabe ausgeben.
- e) Die Kurse sollen mit Hilfe einer dynamischen Liste verwaltet werden. Ergänzen Sie dazu in der Klasse Kurs ein Attribut, das auf den nachfolgenden Kurs in der Liste verweist. Schreiben Sie eine Klasse KursSystem, die den Anfang der Liste speichert.
- f) Implementieren Sie in der Klasse KursSystem eine Suchmethode, die einen Kurs anhand einer Kursnummer aus der Liste sucht und - falls er vorhanden ist - als Rückgabewert zurückgibt.

Fortsetzung nächste Seite!

2. UML

Ein Tennis-Turnier lässt sich folgendermaßen beschreiben:

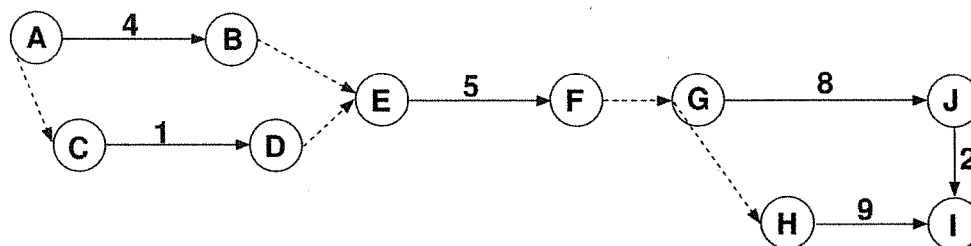
- Zu einem Turnier gehören Spiele, Spieler und Personal. Überlegen Sie hierbei, welche Elemente einzigartige Teile des Turniers sind und welche in mehreren Turnieren vorkommen können.
- Ein Turnier findet in genau einem Land statt.
- Das Personal besteht aus Schiedsrichtern und Hilfsrichtern.
- Jeder Schieds- bzw. Hilfsrichter besitzt einen Namen und stammt aus genau einem Land.
- Jeder Spieler tritt für genau ein Land an.
- Jedes Land hat einen Namen.
- In einem Spiel treten zwei Spieler gegeneinander an. Es wird von einem Schiedsrichter und bis zu 9 Hilfsrichtern begleitet.
- Ein Schiedsrichter ist der Oberschiedsrichter des Turniers.
- Jedes Spiel wird gespielt und dabei von einem Spieler gewonnen.

Erstellen Sie ein UML-Klassendiagramm, das den angegebenen Anforderungen entspricht. Wählen Sie dabei geeignete Datentypen für die Attribute und passende Signaturen für Methoden (üblicherweise parameterlos und leerer Rückgabotyp).

Hinweis: Geben Sie auf jeden Fall **Rollennamen** an! Die Assoziationen müssen selbst **keine Namen** haben. Tragen Sie bitte auch **keine Konstruktoren**, bzw. get- und set-Methoden für Attribute ein.

3. Projektmanagement

Betrachten Sie folgendes CPM-Netzwerk:



- Berechnen Sie die früheste Zeit für jedes Ereignis, wobei angenommen wird, dass das Projekt zum Zeitpunkt 0 startet.
- Setzen Sie anschließend beim letzten Ereignis die späteste Zeit gleich der frühesten Zeit und berechnen Sie die spätesten Zeiten.
- Berechnen Sie nun für jedes Ereignis die Pufferzeiten.
- Bestimmen Sie den kritischen Pfad.
- Was ist ein Gantt-Diagramm? Worin unterscheidet es sich vom CPM-Netzwerk?

Fortsetzung nächste Seite!

Teilaufgabe 2: Datenbanksystem**1. Relationale Anfragesprachen**

Gegeben sei folgendes relationales Schema, dessen Attribute nur atomare Attributwerte besitzen.

Computer {*IP*, *Name*, *Hersteller*, *Modell*, *Standort*}

- Geben Sie für die folgenden Anfragen einen relationalen Ausdruck an:
 - a) Geben Sie die IP-Adresse des Computers mit Namen "Chiemsee" aus.
 - b) Geben Sie 2er-Tupel von IP-Adressen der Computer am selben Standort aus.
- Formulieren Sie die folgenden Anfragen in SQL:
 - a) Geben Sie die IP-Adressen der Rechner am Standort "Büro2" aus.
 - b) Geben Sie alle Computer-Namen in aufsteigender Ordnung mit ihren IP-Adressen aus.
 - c) Geben Sie für jeden Hersteller die Anzahl der unterschiedlichen Modelle aus.
 - d) Geben Sie für jeden Hersteller, welcher mindestens 2 unterschiedliche Modelle hat, die Anzahl der unterschiedlichen Modelle aus.

2. Normalformen

Gegeben sei folgendes relationales Schema, dessen Attribute nur atomare Attributwerte besitzen.

R : {[*A*, *B*, *C*, *D*, *E*]}

Es gelte die folgende Menge *FD* von funktionalen Abhängigkeiten:

FD = {

$ABC \rightarrow DE,$

$E \rightarrow BCD$

}

- a) Bestimmen Sie alle Superschlüssel und Kandidatenschlüssel von *R*.
- b) Welcher Normalform genügt *R*?

Begründen Sie Ihre Aussagen in beiden Teilaufgaben.

Fortsetzung nächste Seite!

3. Modellierung

Es sind folgende Informationen zu einer Datenbank für Konsulate gegeben:

- Jedes Konsulat hat einen Sitz in einer Stadt.
- Zu einem Konsulat sollen ein eindeutiger Name (z.B. Konsulat Bayern), die Adresse, und der Vor- bzw. Nachname des Konsuls gespeichert werden.
- Für jede Stadt sollen der Name, die Anzahl der Einwohner, sowie das Land, in dem es liegt, festgehalten werden. Gehen Sie davon aus, dass eine Stadt nur in Zusammenhang mit dem dazugehörigen Land identifizierbar ist.
- Für ein Land sollen der Name in Landessprache, der Name des Staatspräsidenten und eine eindeutige ID gespeichert werden.

a) Entwerfen Sie für das obige Szenario ein ER-Diagramm 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.

Hinweis: Achten Sie darauf, alle Totalitäten einzutragen.

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.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe 1: Softwaretechnologie

Aufgabe 1:
Grundlagen

Übertragen Sie die Buchstaben a) - q) auf Ihr Blatt und notieren Sie dahinter, ob die entsprechende Aussage wahr oder falsch ist.

Allgemein:

- a) Im Software Engineering geht es vor allem darum, qualitativ hochwertige Software zu entwickeln.
- b) Software Engineering ist gleichbedeutend mit Programmieren.

Vorgehensmodelle:

- c) Das Spiralmodell ist ein Vorläufer sog. Agiler Methoden.
- d) Agile Methoden eignen sich besonders gut für die Entwicklung komplexer und sicherer Systeme in verteilten Entwicklerteams.
- e) Die Erhebung und Analyse von Anforderungen sind nicht Teil des Software Engineerings.

Anforderungserhebung:

- f) Bei der Anforderungserhebung dürfen in keinem Fall mehrere Erhebungstechniken (z. B. Workshops, Modellierung) angewendet werden, weil sonst Widersprüche in Anforderungen zum Vorschein kommen könnten.
- g) Ein Szenario beinhaltet eine Menge von Anwendungsfällen.
- h) Nicht-funktionale Anforderungen sollten, wenn möglich, immer quantitativ spezifiziert werden.

Architekturmuster:

- i) Schichtenarchitekturen sind besonders für Anwendungen geeignet, in denen Performance eine wichtige Rolle spielt.

UML:

- j) Sequenzdiagramme beschreiben Teile des Verhaltens eines Systems.
- k) Komponentendiagramme beschreiben die Struktur eines Systems.
- l) Zustandsübergangsdiagramme beschreiben das Verhalten eines Systems.

Fortsetzung nächste Seite!

Enwurfsmuster:

- m) Das MVC Pattern verursacht eine starke Abhängigkeit zwischen Datenmodell und Benutzeroberfläche.
- n) Das Singleton Pattern stellt sicher, dass es zur Laufzeit von einer bestimmten Klasse höchstens ein Objekt gibt.

Testen:

- o) Testen ermöglicht sicherzustellen, dass ein Programm absolut fehlerfrei ist.
- p) Validation dient der Überprüfung von Laufzeitfehlern.
- q) Verifikation dient der Überprüfung, ob ein System einer Spezifikation entspricht.

Fortsetzung nächste Seite!

Aufgabe 2:

Zeichnen Sie den Ablauf des V-Modells und beschreiben Sie die einzelnen Phasen in kurzen Sätzen.

Aufgabe 3:**Modellierung**

Sie sollen ein Car-Sharing System modellieren. Das System soll die Verwaltung von Buchungen ermöglichen. Beachten Sie dabei folgendes:

1. Eine Buchung hat eine BuchungsNR, ein startDatum und ein endDatum. Außerdem beinhaltet sie zugehörige Fahrzeuge und einen Kunden. Zusätzlich soll es möglich sein, eine potentielle Buchung zu prüfen (ob die Fahrzeuge für besagten Zeitraum noch verfügbar sind) und evtl. zu bestätigen.
2. Ein Kunde hat eine ID, einen Namen und eine Adresse und beliebig viele Buchungen.
3. Ein Fahrzeug hat eine ID, einen Hersteller, ein Baujahr und eine Farbe. Außerdem gehört es einer Kategorie an.
4. Eine Kategorie hat einen Namen und einen Preis pro Tag.

Aufgabe a)

Erstellen Sie ein Datenmodell in Form eines UML-Klassendiagramms.

Aufgabe b)

Erstellen Sie ein UML-Sequenzdiagramm, welches die Kommunikation zwischen einem Kunden, dem System und einem Objekt Buchung (welches die Details zur Buchung enthält) modelliert. Dabei soll ein Anwendungsfall modelliert werden, der aus den nachfolgend genannten Schritten besteht.

1. Ein Kunde sucht nach Fahrzeugen einer bestimmten Kategorie für einen bestimmten Zeitraum.
2. Der Kunde kann dann eine neue Buchung tätigen.
3. Das System erstellt dabei eine neue (potentielle) Buchung.
4. Danach wird geprüft, ob die Buchung möglich ist (die Prüfung erfolgt über eine Methode des Objektes Buchung).
5. a) Wenn die Buchung möglich ist, wird sie bestätigt und der Kunde darüber informiert.
b) Ist die Buchung nicht möglich, wird die Buchung gelöscht und der Kunde darüber informiert.

Aufgabe c)

Implementieren Sie in einer gängigen objektorientierten Programmiersprache eine Methode "kosten" für das Objekt Buchung, welche die gesamten Kosten für eine Buchung berechnet.

Fortsetzung nächste Seite!

Teilaufgabe 2: Datenbanksysteme

Aufgabe 1:

Datenbankentwurf

Versicherung

Für die fiktive Versicherung "Safe" soll ein Informationssystem aufgebaut werden. Fertigen Sie, um die dafür benötigten Daten speichern zu können, ein ER-Schema an. Geben Sie dabei Attribute von Entitäten und Beziehungen an; kennzeichnen Sie Schlüsselattribute durch Unterstreichen. Die Kardinalitäten von Beziehungen sollen in der (min, max)-Notation in das Modell aufgenommen werden. Führen Sie Surrogatschlüssel (künstliche Schlüsse) nur ein, falls es nötig ist.

Die zu modellierende Versicherung betreibt mehrere Filialen, die eine eindeutige Adresse besitzen und von denen der jährliche Umsatz erfasst werden soll. In einer Filiale arbeiten verschiedene Mitarbeiter der Versicherung, die einer Tarifgruppe zugeordnet werden. Jeder Mitarbeiter ist genau einer Filiale zugewiesen.

Eine weitere wichtige Personengruppe stellen Kunden dar, die über eine Kundennummer verfügen und von denen die Berufsgruppe bedeutsam ist. Für jede Person, die in der Versicherung von Bedeutung ist, sollen der Name, das Geburtsdatum, die Adresse (Straße, Hausnummer, PLZ) und die eindeutige Identifikationsnummer abgespeichert werden.

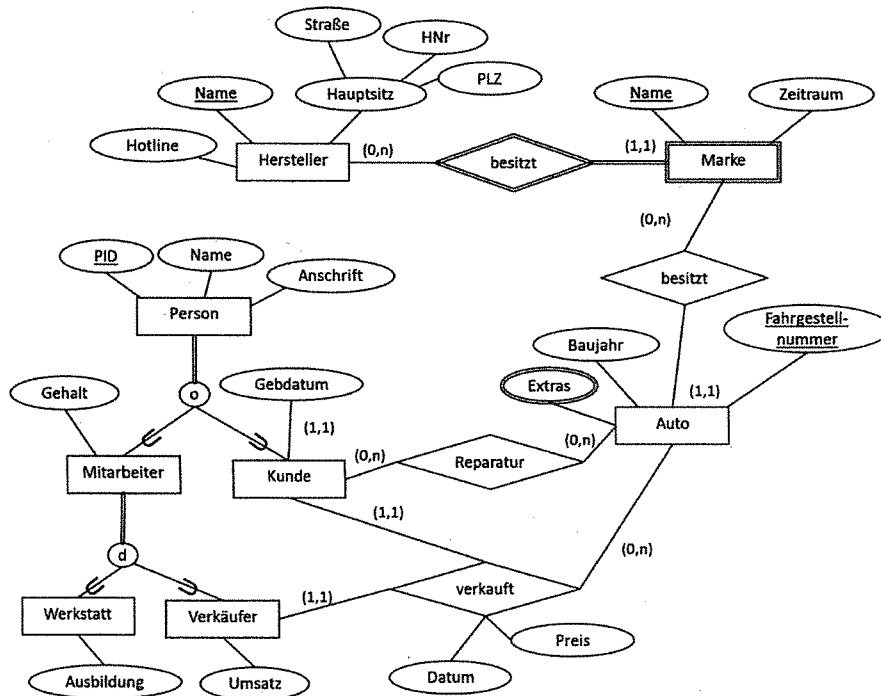
Für die Berechnung bestimmter tariflicher Optionen ist es wichtig, dass für jeden Kunden Angehörige in der Datenbank hinterlegt werden können. Diese besitzen einen Namen und ein Geburtsdatum.

Die Versicherung bietet verschiedene Produkte an, die eindeutig über eine Nummer identifiziert werden und über einen bestimmten Zeitraum angeboten werden. Die wichtigsten (aber nicht alle) Produktgruppen sind: KFZ-, Hausrat- und Lebensversicherungen. Bei KFZ-Versicherungen ist die Höhe des versicherten Personenschadens von Bedeutung. Außerdem sollen bestimmte (potentiell mehrere) Zusatzleistungen abgespeichert werden. Hausratversicherungen besitzen einen gewissen Umfang und Lebensversicherungen eine Auszahlungsfrist. Produkte werden in der Versicherung von einem Mitarbeiter an einem gewissen Datum zu einem gewissen Preis an einen Kunden vermittelt.

Aufgabe 2:

Relationen

Transformieren Sie das unten angegebene ER-Schema in ein Relationenschema. Kennzeichnen Sie Primärschlüssel durch Unterstreichen. Geben Sie bei Fremdschlüsseln deren Herkunft an.



Hinweis: Sie dürfen für FOREIGN KEY (Attributmenge) IS KEY OF Relation die Abkürzung FK (Attributmenge) iKO Relation verwenden.

Aufgabe 3:

SQL

Gegeben ist das Schema eines Turn- und Sportvereins (TSV). Teilnehmer besuchen Kurse (gegen monatliche Gebühr), welche von einem Kursleiter gehalten werden. Ein Teilnehmer kann einen Kurs noch aktiv besuchen (aktiv="ja") oder bereits ausgetreten sein (aktiv="nein"). Die entrichtete monatliche Kursgebühr kommt zu 80 % dem Kursleiter zu.

Kursleiter (PersNr, Name, Adresse, Ausbildung)

Teilnehmer (MitgliedsNr, Name, Adresse, Geburtsjahr)

Kurs (KursID, Name, Gebühr, Beschreibung, Ort, Leiter)

- Leiter ist Fremdschlüssel und bezieht sich auf den Primärschlüssel von Kursleiter

besucht (Kurs, Mitglied, aktiv, Eintrittsjahr, Austrittsjahr)

- Kurs ist Fremdschlüssel und bezieht sich auf den Primärschlüssel von Kurs
- Mitglied ist Fremdschlüssel und bezieht sich auf den Primärschlüssel von Teilnehmer

Fortsetzung nächste Seite!

- a) Gesucht werden alle Teilnehmer, die dem Karatekurs mit einem Alter von maximal 12 Jahren (bezogen auf die Jahreszahl) beigetreten sind. Auszugeben sind dabei deren Name, Adresse, Eintrittsalter, sowie ob sie noch aktiv an diesem Kurs teilnehmen. Geben Sie das entsprechende SQL-Statement an.
- b) Die Gebühren des Tenniskurses sind um 10 % erhöht worden. Der entsprechende Eintrag ist in der Datenbank auszubessern. Geben Sie das entsprechende SQL-Statement an.
- c) Gesucht werden alle Mitglieder, die momentan mindestens zwei verschiedene Kurse besuchen, wobei alle Attribute des jeweiligen Mitglieds inklusive der Anzahl der besuchten Kurse ausgegeben und die Mitglieder mit den meisten Kursen zuerst aufgeführt werden sollen. Geben Sie das entsprechende SQL-Statement an.
- d) Formulieren Sie umgangssprachlich, wonach mit folgendem SQL-Statement gesucht wird.

```
SELECT Leiter, SUM(0.8*Gebühr) AS Gesamtsumme  
      FROM Kurs  
      GROUP BY Leiter  
      ORDER BY Gesamtsumme ASC
```