
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2012**

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: 10

Bitte wenden!

Thema Nr. 1**Teilaufgabe 1:****1. ER-Modellierung**

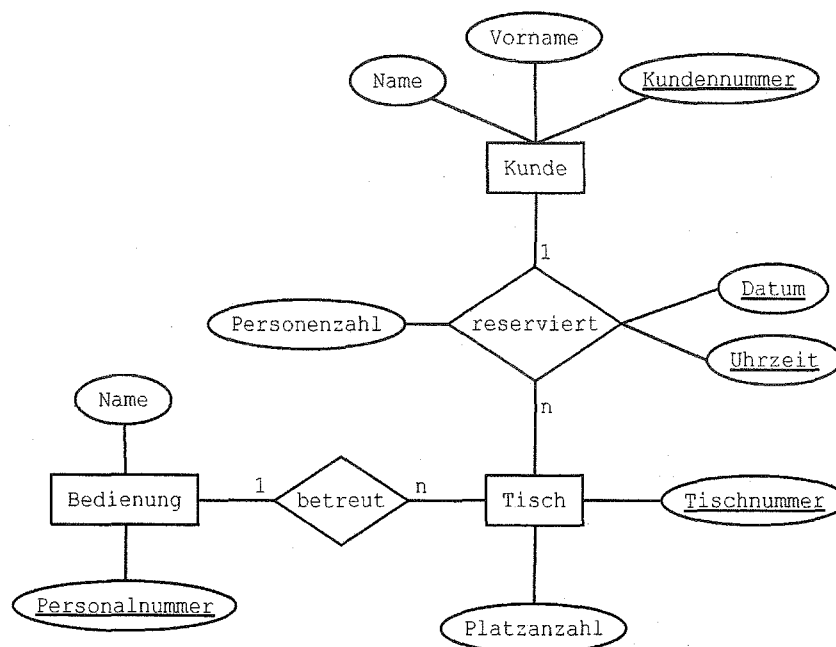
Im Folgenden wird eine vereinfachte Kinoverwaltung beschrieben.

Ein Kino hat einen Namen und eine Anschrift (bestehend aus Straße, Postleitzahl und Ort), durch die es eindeutig beschrieben wird. Jedes Kino besitzt eine bestimmte Anzahl an Sälen, in welchen Filme vorgeführt werden. Jeder Saal hat eine eigene Nummer und eine bestimmte Anzahl an Sitzplätzen. In den Sälen werden regelmäßig Filme gezeigt. Ein Film zeichnet sich durch seinen Titel sowie seine Kopie-Nummer aus und hat eine bestimmte Spieldauer. Jede Filmvorführung hat einen bestimmten Preis und findet an einem bestimmten Datum zu einer bestimmten Uhrzeit statt. In jedem Film spielen Schauspieler mit. Ein Schauspieler hat einen Namen und kommt aus einem Land. Es können Schauspieler mit demselben Namen im selben Land beheimatet sein.

Erstellen Sie zum obigen Szenario ein ER-Diagramm. Geben Sie dabei alle notwendigen Entitäten und Beziehungen sowie sämtliche Kardinalitäten an. Kennzeichnen Sie jeweils die Schlüssel. Sollte es nötig sein, führen Sie Surrogatschlüssel (künstliche Schlüssel) ein.

2. Relationale Datenbankmodellierung

Das folgende Diagramm zeigt die Modellierung der Tischreservierung in einem Lokal.



Geben Sie zu diesem ER-Modell ein Relationenschema an und vereinfachen Sie es anschließend so weit wie möglich.

Fortsetzung nächste Seite!

3. SQL

Gegeben ist folgendes einfache Datenbankschema zur Personalverwaltung (Schlüsselattribute unterstrichen, Fremdschlüssel überstrichen):

Angestellter (PersNr, Name, Gehalt, Beruf, AbtNr, Ort)

Abteilung (AbtNr, Name, Ort)

Formulieren Sie folgende Datenbankoperationen in SQL:

- a) Welche Angestellten sind von Beruf Koch und verdienen mehr als 5000 €?
- b) Welche Angestellten der Abteilung B17 sind aus München?
- c) Hans Meier aus der Abteilung C4 zieht von München nach Erlangen um.
- d) Abteilung C4 wird aufgelöst.

Formulieren Sie folgende SQL-Anfrage umgangssprachlich und so exakt wie möglich.

- e) SELECT AbtNr
FROM Abteilung a, (SELECT PersNr, Name, AbtNr
FROM Angestellter
WHERE Gehalt < 2000)
WHERE a.AbtNr = t.AbtNr
AND a.Ort = 'Nuernberg';

4. Normalformenlehre

- a) Nennen und erläutern Sie kurz die Ziele der Normalisierung einer Datenbank.
- b) Gegeben ist das Datenbankschema in 3. Normalform aus der vorhergehenden Teilaufgabe:

Angestellter (PersNr, Name, Gehalt, Beruf, AbtNr, Ort)

Abteilung (AbtNr, Name, Ort)

Zu jedem Ort sollen nun auch die Postleitzahl und die Straße gespeichert werden. Dabei soll die dritte Normalform erhalten bleiben. Ein Firmenmitarbeiter schlägt folgende zwei Lösungen vor:

I. Speicherung der Postleitzahl und Straße zusammen mit dem Ort (z. B. „91058 Erlangen Martensstraße 3“ als Eintrag unter Ort)

II. Folgende Erweiterung des Datenbankschemas:

Angestellter (PersNr, Name, Gehalt, Beruf, AbtNr, Ort, Straße, Postleitzahl)

Abteilung (AbtNr, Name, Ort, Straße, Postleitzahl)

Begründen Sie, warum beide Lösungen die 3. Normalform verletzen.

- c) Geben Sie einen eigenen Vorschlag zur Lösung des Problems an.

Fortsetzung nächste Seite!

Teilaufgabe 2:**Aufgabe 1**

Erläutern Sie zwei verschiedene Vorgehensmodelle der Softwareentwicklung und diskutieren Sie jeweils Vor- und Nachteile.

Aufgabe 2

Erläutern Sie die Drei-Schichtenarchitektur für betriebliche Informationssysteme (jeweils mit den Zuständigkeiten der einzelnen Schichten).

Aufgabe 3

Es seien folgende Anforderungen gegeben: Eine Bank hat Kunden und sie führt Konten. Ein Konto gehört genau einem Kunden, aber nicht jeder Kunde muss ein Konto besitzen. Es gibt Spar- und Girokonten.

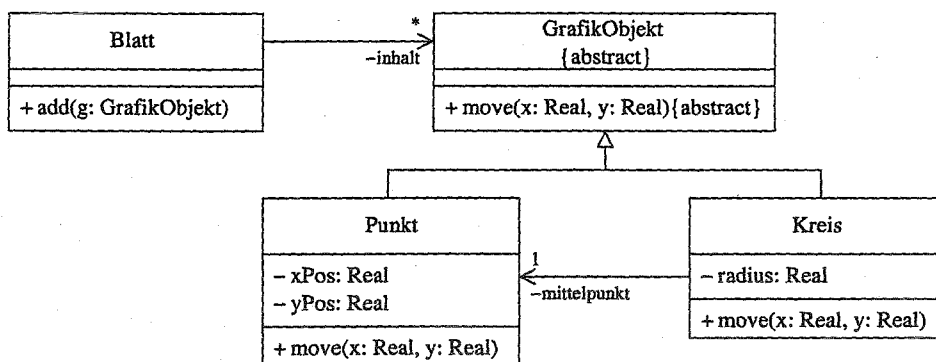
Zu einem Zeitpunkt soll die folgende Situation bestehen: Die Bank mit Namen XY hat als Kunden Frau *Sandra Huber* mit der Kundennummer 1001 und Herrn *Peter Meier* mit der Kundennummer 2002. Frau Huber besitzt ein Girokonto mit Kontonummer 304 und einem Kontostand von -2500,30 Euro sowie ein Sparkonto mit Kontonummer 507 und einem Kontostand von 4500,00 Euro. Herr Meier besitzt ein Girokonto mit Kontonummer 811 und einem Kontostand von 3700,00 Euro. Für jedes Girokonto wird monatlich eine Gebühr von 4,09 Euro erhoben, für das Sparkonto gibt es 2,75% Zinsen im Jahr.

- (a) Erstellen Sie ein UML-Klassendiagramm, das die beschriebene Situation zulässt. Insbesondere sind Multiplizitäten und die Typen von Attributen anzugeben. Assoziationen sollen bidirektional sein. Operationen sind (noch) nicht zu berücksichtigen.
- (b) Erstellen Sie ein UML-Objektdiagramm (Instanzendiagramm), das die oben beschriebene Situation darstellt.
- (c) Ein Bankangestellter soll einen Monatsabschluss bei einer Bank durchführen können. Wird ein Monatsabschluss veranlasst, dann führt die Bank bei jedem ihrer Konten eine Monatsabrechnung durch, wobei bei einem Sparkonto die monatlichen Zinsen gutgeschrieben werden und bei einem Girokonto die monatliche Gebühr abgezogen wird.
Konstruieren Sie ein Sequenzdiagramm, das die Abfolge der Nachrichten bei der Durchführung eines Monatsabschlusses für die oben genannten Objekte zeigt.
- (d) Ergänzen Sie die betreffenden Klassen um die Operationen, die sich aus dem Sequenzdiagramm ergeben. Welche Operationen sind abstrakt? Für nicht abstrakte Operationen soll die Wirkung mit Hilfe eines Programmcodes einer objektorientierten Programmiersprache Ihrer Wahl (Sprache angeben) oder mittels (intuitiv klarem) Pseudo-Code formuliert werden.

Fortsetzung nächste Seite!

Aufgabe 4

Das folgende Klassendiagramm ist in einer objektorientierten Programmiersprache Ihrer Wahl (Sprache angeben) zu realisieren. Der Rumpf der beiden konkreten move-Methoden braucht nicht angegeben zu werden. Die Operation `add(g:GrafikObjekt)` soll ein Grafikobjekt `g` zum Inhalt eines Blattes hinzufügen.



Thema Nr. 2

Teilaufgabe 1:

1 ER-Modellierung und Relationenmodell

1.1

Sie sollen die Verwaltung einer Fluggesellschaft modellieren. Erstellen Sie unter Berücksichtigung der unten beschriebenen Kriterien ein ER-Diagramm bestehend aus Entitäts- und Beziehungstypen sowie Attributen. Geben Sie auch die Kardinalitäten mit an.

Verwenden Sie bei Entitäten und Attributen ausschließlich die beschriebenen, soweit dies möglich ist. Es ist ebenfalls nicht erlaubt, künstliche Schlüssel zu erfinden. Die Kardinalitätseinschränkungen können Sie entweder in der (min,max)-Notation oder der einfachen Notation nach Chen (1:1, 1:N, M:N) angeben.

Ein Flugzeug hat eine eindeutige Kennzeichnung sowie eine Beschreibung.

Ein Flugzeugtyp hat eine eindeutige ID, einen Namen, Kapazität und Reichweite.

Jedes Flugzeug ist von genau einem Flugzeugtyp.

Flugzeuge werden auf Routen eingesetzt. Natürlich können mehrere Flugzeuge dieselbe Route befliegen. Ebenso kann ein Flugzeug auf verschiedenen Routen eingesetzt werden. Eine Route hat einen eindeutigen Namen sowie einen Start- und Zielflughafen. Zu einer Route wird außerdem die Länge in Meilen gespeichert.

Passagiere werden durch ihre Ausweisnummer unterschieden. Sie haben einen Vor- und einen Nachnamen sowie eine Kreditkartennummer.

Passagiere können an bestimmten Daten mit einem Flugzeug auf einer Route mitfliegen. Ein Datum besteht aus Jahr, Monat und Tag. Natürlich muss es möglich sein, dass ein Passagier eine Route mit einem Flugzeug an unterschiedlichen Daten nutzt.

1.2

Erstellen Sie zu dem ER-Diagramm aus Aufgabe 1.1 ein Relationenschema.

Berücksichtigen Sie dabei totale Partizipationen und vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen.

Beispiel für die Notation:

```
Relationenname (Primärschlüssel, Attribut1, Attribut2, ...,  
    Fremdschlüssel[AndereRelation],  
    (FKAttrA, FKAttrB)[DritteRelation])  
Attribut2 NOT NULL
```

Fortsetzung nächste Seite!

2 Normalformen

2.1

Die Normalisierung von Relationenschemata dient der Vermeidung von Redundanzen und dadurch bedingter Anomalien. Geben Sie ein Beispiel für eine nicht-normalisierte Relation an und erläutern Sie zwei mögliche Anomalien an diesem Beispiel.

2.2

Normalisierung beruht auf dem Erkennen und Eliminieren von funktionalen Abhängigkeiten. Erläutern Sie in diesem Zusammenhang kurz die folgenden Begriffe:

- a) Funktionale Abhängigkeit
- b) Voll-funktionale Abhängigkeit
- c) Transitive funktionale Abhängigkeit
- d) Superschlüssel
- e) Determinante

2.3

In welcher Normalform ist das folgende Beispiel? Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind! Welche Bedingung der nächsthöheren Normalform ist verletzt? Berücksichtigen Sie, dass eine Relation nur dann in n-ter Normalform ist, wenn sie die Bedingungen aller m-ten Normalformen mit $m \leq n$ erfüllt.

PersNr	Name	Strasse	Ort	PLZ	Vorwahl
1	Meier	Abbestr.	München	80999	089
2	Meier	Allacher Str.	München	80997	089
3	Müller	Zieglhof	Regensburg	93055	0941
4	Schmidt	Bucher Hauptstr.	Nürnberg	90427	0911
5	Schmid	Osserweg	Passau	94034	0851
6	Meier	Alfred-Nobel-Str.	Würzburg	97080	0931

Fortsetzung nächste Seite!

3 Anfrageverarbeitung

3.1

Nennen und charakterisieren Sie die wesentlichen Schritte, die eine SQL-Anfrage bei der Verarbeitung durchläuft.

3.2

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

```
SELECT R1.id AS r1, R2.id AS r2, R3.id AS r3
  FROM R1, R2, R3
 WHERE R1.R2_id = R2.id AND R3.R2_id = R2.id
```

Fortsetzung nächste Seite!

Teilaufgabe 2:**1. UML - Klassendiagramm**

Erstellen Sie zu der folgenden Beschreibung eines Systems zur Organisation eines Hotels ein Klassendiagramm, das Attribute und Assoziationen mit Kardinalitäten sowie Methoden enthält. Setzen Sie dabei das Konzept der Vererbung sinnvoll ein.

- Ein Hotel besteht aus genau einer Küche und mehreren Gästezimmern.
- Es hat einen Namen, eine Adresse und kann Werbeaktionen durchführen.
- In der Küche arbeitet eine bestimmte Anzahl von Mitarbeitern. Die Küche kann geöffnet oder geschlossen sein.
- Gästezimmer haben eine Nummer und eine bestimmte Anzahl an Betten, die ausgegeben werden kann.
- In diesen Gästezimmern wohnen Gäste, die von einer oder mehreren Servicekräften umsorgt werden.
- Servicekräfte sind Angestellte des Hotels und sind nur für die ihnen zugeordneten Gästezimmer verantwortlich.
- Jede Servicekraft hat einen Namen und eine Personalnummer sowie einen Vorgesetzten, der auch eine Servicekraft ist.
- In der Küche arbeiten Küchenmitarbeiter, die einen Namen haben und ein Gehalt bekommen.
- Küchenmitarbeiter sind entweder Köche oder Aushilfen. Köche können zudem Sterneköche sein, also mit einem oder mehreren Sternen dekoriert sein.
- Aushilfen bauen die Büffets für die Mahlzeiten auf.
- Gäste können sich unterhalten. Jeder Gast hat einen Namen und eine Adresse und ist seinem Zimmer zugeordnet.

Fortsetzung nächste Seite!

2. Objektorientierte Programmierung

Ein Versicherungsunternehmen bietet KFZ-Versicherungen mit verschiedenen Jahresbeiträgen und Selbstbeteiligungen im Schadensfall an. Lösen Sie die folgenden Aufgaben in einer objektorientierten Programmiersprache Ihrer Wahl (Sprache vorher angeben):

- a) Alle Versicherungsverträge des Unternehmens sollen fortlaufende Vertragsnummern erhalten. Erstellen Sie eine Klasse Vertrag. Die eindeutigen Vertragsnummern sollen über ein Objektfeld, das im Standardkonstruktor die nächste freie Vertragsnummer zugewiesen bekommt, an ihre Unterklassen vererbt werden.
- b) Erstellen Sie nun für KFZ-Versicherungsverträge eine weitere Klasse Vollkasko als Unterklasse von Vertrag mit Objektfeldern für
 - den Jahresbeitrag (für 100%)
 - die Schadensklasse (in Prozent)
 - die Höhe der Selbstbeteiligung

Geben Sie einen Konstruktor an, der alle Objektfelder auf die übergebenen Werte setzt und implementieren Sie folgende öffentliche Methoden:

- `double berechneBeitrag()`
gibt den zu zahlenden Jahresbeitrag entsprechend der Schadensklasse zurück.
 - `double schadensregulierung(double schaden)`
berechnet den Betrag, den die Versicherung bei einem Schadensfall abzüglich Selbstbeteiligung übernehmen muss. Falls der Schadenswert die Selbstbeteiligung überschreitet, soll die Schadensklasse um 15 Prozentpunkte erhöht werden.
- c) Implementieren Sie ein kurzes Testprogramm: Deklarieren und erzeugen Sie ein Objekt der Klasse Vollkasko mit Beispielwerten, berechnen Sie dessen Jahresbeitrag und geben Sie diesen auf der Standardausgabe aus.
 - d) Das Unternehmen führt einen weiteren Vollkaskovertrag ein, dessen Schadensklasse erst bei jedem zweiten Schadensfall steigt. Implementieren Sie dafür eine Klasse VollkaskoPlus als Unterklasse von Vollkasko mit einem zusätzlichen Objektfeld, das als Wahrheitswert angibt, ob bereits ein Schaden gemeldet worden ist, der die Selbstbeteiligung übersteigt.
Die Klasse soll neben einem Konstruktor, der alle benötigten Objektfelder sinnvoll setzt, auch die folgende Methode enthalten:
`double schadensregulierung(double schaden)`
soll die Schadensklasse nur dann erhöhen, falls bereits ein Schaden vorlag. Kam es zu einer Erhöhung, muss der Wahrheitswert wieder zurückgesetzt werden, da der nächste Schadensfall wieder „frei“ ist.
 - e) Erklären Sie kurz die Begriffe *Redefinieren* und *dynamisches Binden* von Methoden in einer objektorientierten Programmiersprache. Was muss beim Redefinieren von Methoden in Hinblick auf die Sichtbarkeiten beachtet werden?