Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 2011

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:

Datenmodellierung: ER–Modell, SQL

In einer *Fußballdatenbank* sollen folgende Informationen gespeichert werden:

- Zu jeder *Fußballweltmeisterschaft* werden *Jahr*, *Austragungsland* und *Weltmeisternation* gespeichert.
- Zu jeder *Nation* werden der *Sitz* des nationalen Fußballverbandes und die Anzahl der bisher errungenen *Weltmeistertitel* gespeichert.
- Zu jeder *Fußballweltmeisterschaft* werden die teilnehmenden Nationen gespeichert sowie die Spieler, die für die Nation an der Weltmeisterschaft teilgenommen haben.
- Zu jedem *Spieler* werden *Name*, *Verein* und *Geburtsdatum* gespeichert.

Nationen und Spieler sollen jeweils noch ein Attribut *Nnr* bzw. *Snr* mit einer eindeutigen Identifikationsnummer haben.

a) Erstellen Sie ein ER–Modell, welches das oben dargestellte Szenario vollständig abbildet. Verwenden Sie wann immer möglich (binäre oder auch höherstellige) Relationships.

b) Übertragen Sie Ihr ER–Modell ins relationale Modell. Erstellen Sie dazu Tabellen mit Hilfe von CREATE TABLE–Statements in SQL. Berücksichtigen Sie die Fremdschlüsselbeziehungen.

c) Erweitern Sie den Entity–Typ *Spieler* um eine *Rückennummer* und machen Sie diese – anstelle der künstlichen Identifikationsnummer *Snr* – zum „schwachen Schlüssel". Diese Rückennummer ist innerhalb einer Nation für jeden Spieler nur innerhalb einer Weltmeisterschaft eindeutig.

   Geben Sie auch das entsprechend veränderte CREATE TABLE–Statement für die Relation *Spieler* an.

d) Geben Sie geeignete INSERT-Statements an, die in alle beteiligten Tabellen jeweils mindestens ein Tupel einfügen, so dass nach Ausführung aller Einfügungen alle Integritätsbedingungen erfüllt sind.

Fortsetzung nächste Seite!

Datenbankanfragen in SQL

Gegeben seien die folgenden drei Tabellen:

- *liefert (LNr, Teil, Anzahl)*
- *Lieferant (LNr, Name, Branche)*
- *Filialen (LNr, Ort),*

wobei sich das Fremdschlüsselattribut *LNr* aus *liefert* und *Filialen* auf das Schlüsselattribut *LNr* aus *Lieferant* bezieht. Formulieren Sie auf den obigen Tabellen folgende SQL-Anfragen:

a) Bestimmen Sie die Namen aller Lieferanten aus *München.*

b) Bestimmen Sie die Namen aller Lieferanten, die das Teil T_1 liefern.

c) Bestimmen Sie, wie viele Teile vom Lieferanten L_1 geliefert werden.

d) Bestimmen Sie für jeden Lieferanten die Summe der Anzahlen der von ihm gelieferten Teile.

Funktionale Abhängigkeiten und Zerlegungen

Gegeben sei das Relationsschema $R = (U, F)$ mit der Attributmenge $U = \{A,B,C,D,E\}$ und der folgenden Menge F von funktionalen Abhängigkeiten:

$$F = \{ A \rightarrow \mathrm{B}, AB \rightarrow C, AD \rightarrow BE, E \rightarrow D\}$$

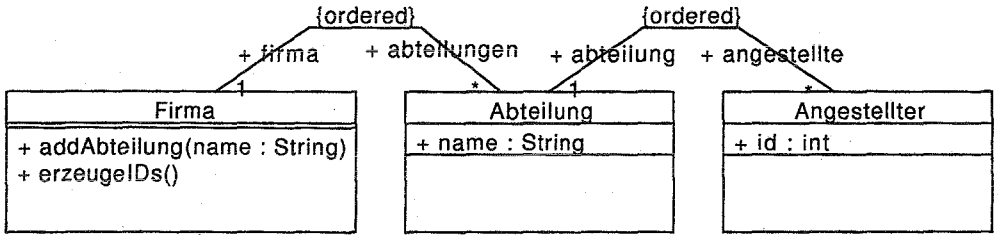
a) Geben Sie alle Schlüssel für R an.

b) Ist R in 3. Normalform bzw. in Boyce-Codd-Normalform?

Fortsetzung nächste Seite!

Teilaufgabe 2:

1 Firmenstruktur



Eine Firma besteht aus null oder mehr Abteilungen, von denen jede null oder mehr Angestellte hat. Da sowohl die Abteilungen als auch deren Angestellten geordnet sind, sind Angestellte insgesamt geordnet. Sie haben durchgehende, ganzzahlige IDs, die bei 1 beginnen.

a) Erstellen Sie exemplarisch ein Objektdiagramm: Stellen Sie eine Firma mit dem Instanznamen `f` und den zwei Abteilungen „Produktion" (Name `p`) und „Marketing" (Name `m`) dar. Die Produktion hat zwei Angestellte, Marketing hat einen Angestellten. Die Angestellten haben die Namen `a1`, `a2`, und `a3`.

b) Implementieren Sie das Klassendiagramm in Java oder in einer anderen geeigneten objektorientierten Programmiersprache Ihrer Wahl. Beachten Sie, dass die Assoziationen bidirektional und geordnet sind. Die beiden Methoden der Klasse `Firma` sollen dabei folgendes Verhalten haben:

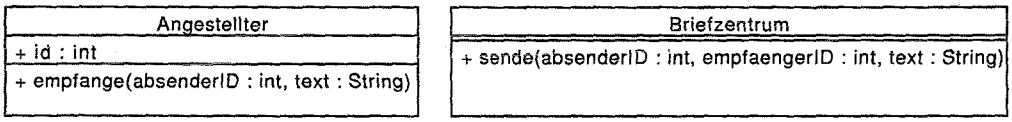
- Die Methode `erzeugeIDs` sorgt dafür, dass die IDs wieder korrekt zugewiesen sind. Die alten IDs können beliebig geändert werden, solange das Endergebnis wieder den obenstehenden Kriterien genügt.

c) Angestellte sollen in Manager und einfache Angestellte unterteilt werden. Zeichnen Sie ein Klassendiagramm mit der Oberklasse `Angestellter` und den zwei Unterklassen `Manager` und `EinfacherAngestellter`. Die Klasse `Angestellter` soll nicht instantiierbar sein und erzwingen, dass die Methode `getPosition()` (öffentlich, ohne Argumente, Rückgabewert `String`) von allen konkreten Unterklassen implementiert wird. `Manager` und `EinfacherAngestellter` sollen instantiierbar sein.

d) Wie lautet der Fachbegriff dafür, dass eine Methode in einer Klasse und in deren Unterklassen dieselbe Signatur hat, aber in den Unterklassen unterschiedlich implementiert ist?

Fortsetzung nächste Seite!

2 Briefe versenden



Ein Angestellter verwendet das Briefzentrum, um einem anderen Angestellten eine Nachricht zu schicken. Absender und Empfänger werden als ganzzahlige IDs angegeben, die Nachricht selbst ist eine Zeichenkette.

a) Erstellen Sie ein Sequenzdiagramm für das folgende Szenario:

- Ein Angestellter (Instanzname `abs`) mit der ID 1 verwendet die Methode `sende()` des Briefzentrums (Name `bz`), um eine Nachricht mit dem Text „Anfrage" an einen anderen Angestellten (Name `empf`) mit der ID 2 zu senden.
- `empf` sendet daraufhin eine Nachricht mit dem Text „Bestätigung" an `abs` zurück.
- Alle Methodenaufrufe sind synchron, „Bestätigung" wird also geschickt, noch während `abs` auf die Beendigung von `sende()` wartet.

3 Telefon

In jedem Büro der Angestellten steht ein Telefon. Es soll als UML-Zustandsdiagramm modelliert werden.

- Dabei gibt es folgende Ereignisse: `abheben`, `auflegen`, `ziffer`.
- Das Diagramm soll *mindestens* die folgenden Zustände enthalten: `verbunden`, `aufgelegt`, `abgehoben`.
- Der Hörer ist anfangs aufgelegt. Dann hebt man ihn ab, wählt genau drei Ziffern und ist verbunden. Der Hörer kann jederzeit aufgelegt werden.

a) Erstellen Sie ein UML-Zustandsdiagramm.

b) Nennen Sie zwei Techniken, mit denen man ein UML-Zustandsdiagramm in Java oder in einer anderen objektorientierten Programmiersprache Ihrer Wahl implementieren kann.

c) Verwenden Sie eine der beiden in Teilaufgabe (b) erwähnten Techniken, um das Zustandsdiagramm aus Teilaufgabe (a) in Java oder einer anderen geeigneten objektorientierten Programmiersprache Ihrer Wahl zu implementieren.

- Das Telefon soll sich die gewählten Ziffern merken, aber nur bis zum Auflegen. Dann werden sie wieder gelöscht.
- Bei einem illegalen Ereignis soll die Ausnahme `IllegalStateException` (in Java eine Unterklasse von `RuntimeException`) geworfen werden.

Thema Nr. 2

Teilaufgabe 1:

Aufgabe 1) Relationales Modell, SQL Data Definition Language (DDL)

Eine Krankenversicherung will Daten über die ärztlichen Behandlungen ihrer Versicherten in einer relationalen Datenbank speichern. Zusätzlich zu den Patientendaten der Versicherten sollen die Daten jeder Behandlung, welche durch einen bestimmten Arzt in einem bestimmten Krankenhaus an einem bestimmten Tag durchgeführt wurde, gespeichert werden. Es gibt also Ärzte, die an einem bestimmten Tag einen oder mehrere Patienten behandelt haben. Ein Arzt arbeitet in genau einem Krankenhaus.

a) Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank.

b) Setzen Sie das gegebene E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Identifizieren Sie dazu zunächst weitere Attribute, die in obigem Diagramm noch nicht enthalten sind und die gegebenen Entities in sinnvoller Weise beschreiben. Es sollen mindestens zwei Attribute pro Entity angegeben werden. Geben Sie die resultierenden Relationenschemata in folgender Schreibweise an:
   Relation (Attribut1, Attribut2, ..., AttributN)
   Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen. Achten Sie auf eine geeignete Modellierung der Relationships.

c) Geben Sie die Anweisungen in SQL DDL an, die notwendig sind, um die Relationen aus Teilaufgabe (b) in einer relationalen Datenbank zu erzeugen. Kennzeichnen Sie dabei die Primär- und Fremdschlüssel der Relationen.

Fortsetzung nächste Seite!

Aufgabe 2) SQL Data Manipulation Language (DML)

Gegeben sei das folgende Datenbank-Schema mit den zugehörigen Ausprägungen. Die Primärschlüssel-Attribute sind dabei jeweils unterstrichen.

Lieferant

LNr	LName	LStadt
L1	Huber	München
L2	Müller	Köln
L3	Meier	München
L4	Weiß	Hamburg
L5	Schwarz	Köln

Ware

WNr	Bezeichnung	Preis
W1	Milch	12
W2	Mehl	15
W3	Brot	32
W4	Zucker	58

Kunde

KNr	KName	KStadt
K1	Fuchs	Hamburg
K2	Wolf	München
K3	Vogel	Köln
K4	Wurm	München

Auftrag

LNr	WNr	KNr	Menge
L1	W1	K1	4
L1	W2	K3	21
L2	W2	K3	3
L2	W4	K3	9
L3	W2	K4	18
L4	W2	K1	27
L4	W4	K4	54
L5	W1	K3	7
L5	W2	K3	4

Fortsetzung nächste Seite!

Formulieren Sie die folgenden Anfragen in SQL:

a) Finden Sie die Namen aller Lieferanten, für welche ein Auftrag zur Milchlieferung gespeichert ist.
b) Erstellen Sie eine Liste mit den Namen von Kunden, welche mindestens zwei Aufträge erteilt haben.
c) Finden Sie zu jeder Ware (WNr) die Anzahl der Aufträge, bei denen diese Ware auftaucht. Sortieren Sie das Ergebnis absteigend nach dieser Anzahl.
d) Errechnen Sie für jeden Lieferanten den Gesamtpreis aller in Auftrag gegebenen Waren. Das Ergebnis soll die Lieferantennummer, den Lieferantennamen und die errechnete Summe enthalten.

Wie sieht die Ergebnisrelation zu folgenden Anfragen aus?

e) SELECT LNr, KNr FROM Lieferant, Kunde WHERE LStadt = KStadt;
f) SELECT * FROM Kunde WHERE KNr NOT IN (SELECT KNr FROM AUFTRAG NATURAL JOIN LIEFERANT WHERE LName = 'Schwarz');

Formulieren Sie die folgenden Anfragen in relationaler Algebra:

g) Finden Sie die Namen aller Lieferanten aus Köln.
h) Geben Sie Nummern und Bezeichnungen aller Waren aus, die nach Hamburg geliefert werden.

i) Geben Sie das Ergebnis des folgenden Ausdrucks der relationalen Algebra als Tabelle an:

$$\pi_{WNr}(\text{Auftrag}) \bowtie \text{Ware}$$

Aufgabe 3) Entwurfstheorie

Gegeben sei die nachfolgende relationale Datenbank mit unterstrichenen Schlüsselattributen. Sie enthält die Daten der ausgeliehenen Filme einer Videothek. Dabei kann ein Film in mehreren Kopien vorhanden sein. Ein Kunde kann aber nur eine Kopie eines Filmes ausleihen:

Relation „Ausleihe":

FilmNr	KdNr	Name	Adresse	Titel	Leihgebühr
1	1	Müller	München	Hangover	3
2	1	Müller	München	Transformers 2	6
3	2	Huber	Nürnberg	Avatar	4
2	2	Huber	Nürnberg	Transformers 2	6
5	3	Meier	Hamburg	Zweiohrküken	4
6	4	Meier	München	13 Semester	2

a) Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können.
b) Geben Sie für das obige Datenbankschema alle funktionalen Abhängigkeiten (inkl. der transitiven) an.
c) Überführen Sie das obige Relationenschema in die dritte Normalform.

Fortsetzung nächste Seite!

Teilaufgabe 2:

1. Lebenszyklus und Vorgehensmodelle

Welche der folgenden Aussagen sind richtig? Geben Sie jeweils eine kurze, stichpunktartige Begründung an!

A	Das Wasserfallmodell sollte nur für große Projekte eingesetzt werden, da der Einarbeitungsaufwand sehr groß ist.
B	Das oberste Ziel des Spiralmodells ist die Minimierung des Risikos durch wiederkehrendes Durchlaufen der einzelnen Phasen.
C	Bei der inkrementellen Entwicklung werden Kundenwünsche explizit berücksichtigt.
D	Die Anforderungsbeschreibung legt fest, was ein Produkt „können soll" und „wie es realisiert ist", aber nicht: wie es aussieht.
E	Die Anforderungen an das zu entwickelnde System werden vom Auftraggeber klassifiziert und hierarchisiert.
F	Eine gute Anforderungsspezifikation muss vor allem für Ingenieure verständlich sein, da die Anforderungsspezifikation die Grundlage der Systementwicklung bildet.
G	Der häufigste Fehler bei der Softwareentwicklung besteht darin, das zu entwickeln, was der Kunde braucht, und nicht das, was er will.
H	Verifikation ist der Prozess der Beurteilung eines Systems mit dem Ziel festzustellen, ob die spezifizierten Anforderungen erfüllt sind.
I	Durch Validierung kann überprüft werden, ob das Produkt den Erwartungen des Kunden entspricht.
J	Das Ziel von Black-Box-Tests ist die umfassende Überprüfung der spezifizierten Funktionalität.
K	Die Definition des Softwareproduktes in der Definitionsphase ist ein iterativer Prozess.
L	Nicht-funktionale Anforderungen sind sowohl Teil des Pflichtenhefts als auch des Lastenhefts.

2. Funktionale Programmierung

Schreiben Sie in einer funktionalen Programmiersprache (oder in entsprechendem Pseudocode) folgende Funktionen:

a) Die Funktion *leneo* soll prüfen, ob die Länge eines Strings gerade oder ungerade ist! Verwenden Sie beim Rückgabeparameter einen möglichst sinnvollen Datentyp.

b) Die Funktion *codetext* soll einen String mit gerader Länge umkehren und einen String ungerader Länge zweimal aneinander hängen.

z. B.: *codetext*(„ADAM") ergibt „MADA", *codetext*(„EVA") ergibt „EVAEVA"

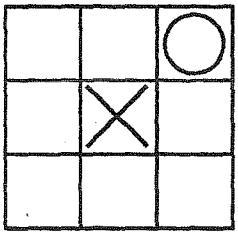
Fortsetzung nächste Seite!

3. Objektorientierter Entwurf

Beim Spiel „TicTacToe" belegen zwei Spieler auf einem Spielfeld von 3x3 Kästchen abwechselnd ein Feld mit ihrem „Symbol". Gewonnen hat derjenige, dem es als erstes gelingt, drei seiner Symbole senkrecht, waagerecht oder diagonal zusammenhängend zu setzen. Das Spiel verfügt über eine Methode zum Spielen. Zu jedem Spieler sollen der Name und sein Symbol gespeichert werden. Nach jedem Setzen eines Symbols prüft das Spiel, ob bereits einer der Spieler gewonnen hat, und gibt ggf. eine entsprechende Meldung zum Spielausgang aus.

a) Erstellen Sie für dieses Szenario ein Klassendiagramm, das mindestens die Klassen Spiel und Spieler enthält. Verwenden Sie nur die nötigen Attribute und Methoden.

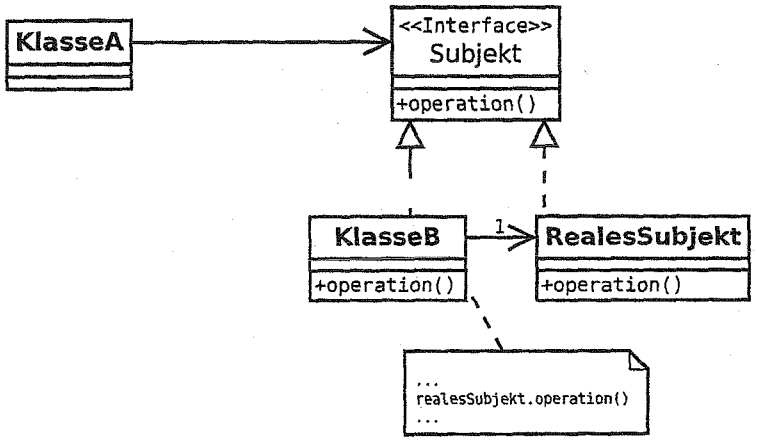
b) Erstellen Sie ein Objektdiagramm für ein Spiel zwischen Peter und Paula, bei dem folgender Spielstand entstanden ist:



c) Erstellen Sie ein Sequenzdiagramm für die Erzeugung eines Spiels sowie die erste Spielrunde, bei der jeder der Spieler einmal an die Reihe kommt.

d) Implementieren Sie die Klasse „Spiel" in einer gängigen objektorientierten Programmiersprache (oder in entsprechendem Pseudocode).

4. Entwurfsmuster



Welches Entwurfsmuster ist in der obigen Abbildung dargestellt? Erläutern Sie dessen Funktionsweise und geben Sie ein Beispiel für seinen Einsatz an.