
Prüfungsteilnehmer**Prüfungstermin****Einzelprüfungsnummer**

Kennzahl: _____**Kennwort:** _____**Arbeitsplatz-Nr.:** _____**Herbst
2015****66116**

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
— Prüfungsaufgaben —**

Fach: Informatik (vertieft studiert)**Einzelprüfung: Datenbanksysteme, Softwaretechnologie****Anzahl der gestellten Themen (Aufgaben): 2****Anzahl der Druckseiten dieser Vorlage: 12**

Bitte wenden!

Thema Nr. 1

Teilaufgabe 1

1. Modellierung

Wir wollen eine relationale Datenbankstruktur für ein Online-Auktionshaus modellieren.

Das Auktionshaus hat Mitglieder. Diese Mitglieder haben Kundennummern, Namen und Adressen. Sie können Verkäufer und/oder Käufer sein. Verkäufer können eine Laden-URL (eine Subdomain) erhalten, die zu einer Seite mit ihren aktuellen Auktionen führt. Verkäufer können neue Auktionen starten. Die Auktionen eines Verkäufers werden durchnummeriert. Diese Auktionsnummer ist nur je Verkäufer eindeutig. Jede Auktion hat ein Mindestgebot und eine Ablaufzeit.

Auf Auktionen können Gebote abgegeben werden. Die Gebote auf eine Auktion werden nach ihrem Eintreffen nummeriert. Diese Nummer ist nur innerhalb einer Auktion eindeutig. Zu einem Gebot werden noch die Zeit des Gebots sowie der gebotene Geldbetrag angegeben. Die Gebote werden von Käufern abgegeben. Jedes Gebot muss einem Käufer zugeordnet sein.

Jede Auktion besteht aus einer Menge von Artikeln, aber aus mindestens einem. Artikel haben eine Beschreibung und können über ihre Artikel-ID identifiziert werden. Zur Katalogisierung der Artikel gibt es Kategorien. Kategorien haben eine eindeutige ID und einen Namen. Jede Kategorie kann Subkategorien besitzen. Auch Subkategorien sind Kategorien (und können damit weitere Subkategorien besitzen). Jeder Artikel kann beliebig vielen Kategorien zugeordnet werden.

Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
- ein passendes ER-Diagramm,
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen,
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

2. Normalformen

Gegeben sei folgendes verallgemeinerte Relationenschema in 1. Normalform:

$$R:\{[A,B,C,D,E,F,G,H]\}$$

Für R soll die folgende Menge FD von funktionalen Abhängigkeiten gelten:

- $F \rightarrow E$
- $A \rightarrow BD$
- $AE \rightarrow D$
- $A \rightarrow EF$
- $AG \rightarrow H$

Fortsetzung nächste Seite!

Bearbeiten Sie mit diesen Informationen folgende Teilaufgaben. Vergessen Sie dabei nicht Ihr Vorgehen stichpunktartig zu dokumentieren und zu begründen.

- Bestimmen Sie alle Schlüsselkandidaten von R. Begründen Sie stichpunktartig, warum es außer den von Ihnen gefundenen Schlüsselkandidaten keine weiteren geben kann.
- Ist R in 2NF, 3NF?
- Berechnen Sie eine kanonische Überdeckung von FD. Es genügt, wenn Sie für jeden der vier Einzelschritte die Menge der funktionalen Abhängigkeiten als Zwischenergebnis angeben.
- Bestimmen Sie eine Zerlegung von R in 3NF. Wenden Sie hierfür den Synthesealgorithmus an.

3. SQL

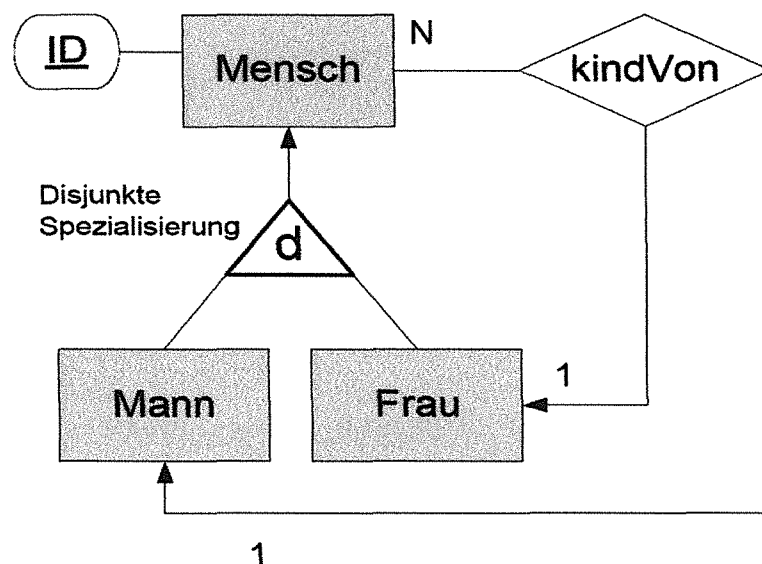
Gegeben seien folgende Relationen:

Mensch(ID, MutterID, VaterID)

Mann(ID)

Frau(ID)

Das zugehörige ER-Modell für dieses relationale Datenbankschema sieht folgendermaßen aus:



Fortsetzung nächste Seite!

Bearbeiten Sie folgende Teilaufgaben:

- a) Finden Sie die Töchter der Frau mit ID 42.
- b) Gibt es Männer, die ihre eigenen Großväter sind? Formulieren Sie eine geeignete SQL-Anfrage.
- c) Definieren Sie eine View *VaterKind* (*VaterID*; *KindID*), die allen Vätern (*VaterID*) ihre Kinder (*KindID*) zuordnet. Diese View darf keine NULL-Werte enthalten.
- d) Verwenden Sie die View aus c), um alle Väter zurückzugeben, absteigend geordnet nach der Anzahl ihrer Kinder.
- e) Hugo möchte mit folgender Anfrage auf Basis der View aus c) alle kinderlosen Männer erhalten:

```
SELECT VaterID
FROM VaterKind
GROUP BY VaterID
HAVING COUNT(KindID) = 0
```

- i. Was ist das Ergebnis von Hugos Anfrage und warum?
- ii. Formulieren Sie eine Anfrage, die tatsächlich alle kinderlosen Männer zurückliefert.

Hinweis: Denken Sie daran, dass SQL auch Mengenoperationen kennt.

Teilaufgabe 2

1. Qualitätsmanagement

Gegeben ist die folgende Methode zur Berechnung des größten gemeinsamen Teilers zweier Zahlen m und n:

```
1      public static int ggt(int m, int n)
2      {
3          if (m <= 0 || n <= 0)
4              return -1;
5          else {
6              while (m != n) {
7                  if (m < n) {
8                      int h = m;
9                      m = n;
10                     n = h;
11                 }
12                 int h = m - n;
13                 m = n;
14                 n = h;
15             }
16             return m;
17         }
18     }
```

Fortsetzung nächste Seite!

- a) Geben Sie einen zur Methode ggt passenden Kontrollflussgraphen an. Benennen Sie die Knoten mit der zur jeweiligen Anweisung gehörigen Zeilennummer.
- b) Erläutern Sie die Begriffe Knotenüberdeckung und Boundary-Interior-Pfadüberdeckung.
- c) Geben Sie eine minimale Testmenge an, die das Kriterium der Knotenüberdeckung erfüllt. Geben Sie für jeden Testfall den durchlaufenen Pfad als Sequenz von Knotennummern an.
- d) Verfahren Sie analog zu c) für das Kriterium der Kantenüberdeckung.
- e) Wie viele Testfälle werden für die Methode ggt benötigt, um die Boundary-Interior-Pfadüberdeckung zu erreichen? Begründen Sie Ihre Antwort.

2. Klassendiagramm

Erstellen Sie zu der folgenden Beschreibung ein UML-Klassendiagramm. Setzen Sie insbesondere Sichtbarkeiten von Elementen, abstrakten Klassen, Vererbungsbeziehungen, Multiplizitäten, Assoziationen, Rollennamen, Assoziationsnamen, Kompositionsbeziehungen, Leserichtungen und Eigenschaften ein, um die Beschreibung möglichst vollständig umzusetzen. Wenn Sichtbarkeiten nicht explizit erwähnt sind, gehen Sie von öffentlicher Sichtbarkeit aus. Wählen Sie geeignete Datentypen für Attribute.

Eine Organisationseinheit hat einen Namen und enthält beliebig viele Organisationsobjekte. Jedes Organisationsobjekt ist in höchstens einer Organisationseinheit enthalten. Ein Organisationsobjekt ist entweder eine Organisationseinheit oder eine Person. Eine Person hat genau einen Nachnamen, beliebig viele Vornamen, einen Geburtsort und ein Geburtsdatum. Ein Datum wird durch einen Datentyp modelliert, der Tag, Monat und Jahr gruppiert. Zu einer Person kann das Alter in Jahren mittels einer Operation bestimmt werden, die das aktuelle Datum als Parameter erhält.

Jede Person ist entweder ein Student oder ein Dozent. Ein Dozent hat ein Personalkennzeichen (eine beliebige Zeichenkette). Einem Studenten ist eine (ganzzahlige) Matrikelnummer zugeordnet, die nur gelesen werden darf. Ferner soll ein Zähler für die Matrikelnummer verwaltet werden, für den als Anfangswert 1 gewählt wird.

Eine Lehrveranstaltung hat einen Namen. Lehrveranstaltungen können voneinander abhängen (uneingeschränkte Multiplizitäten). Jeder Dozent hält mindestens eine Lehrveranstaltung, die ihrerseits von mindestens einem Dozenten angeboten wird. Jeder Student hört mindestens eine Lehrveranstaltung, die ihrerseits von mindestens einem Studenten gehört wird. Jede Lehrveranstaltung ist entweder eine Vorlesung oder eine Übung. Eine Übung kann höchstens eine Vorlesung unterstützen, und zu jeder Vorlesung gibt es höchstens eine Übung.

Fortsetzung nächste Seite!

3. Zustandsdiagramm

Erstellen Sie ein UML-Zustandsdiagramm, das die möglichen Zustände und Transitionen von Benutzern eines Stock-Photo-Systems (Vorrat-Foto-System) beschreibt. Vor- und Nachbedingungen sowie Aktionen dürfen Sie in Java-ähnlicher Syntax angeben.

Orientieren Sie sich an der folgenden Verhaltensbeschreibung:

- Nach der Registrierung kann ein Benutzer entscheiden, ob er zunächst aktiv (als Fotograf) oder passiv (nicht als Fotograf) teilnimmt.
- Ein (aktiver oder passiver) Benutzer kann sich jederzeit vom System abmelden. Abgemeldete Benutzer werden automatisch aus dem System gelöscht, wenn sie ihre Entscheidung nicht innerhalb von 100 Tagen revidieren.
- Ein passiver Benutzer kann jederzeit zu einem aktiven Benutzer werden und seine Karriere als Fotograf starten.
- Ein aktiver Benutzer wird überprüft, sobald eines seiner Fotos von einem anderen Benutzer als unangemessen gemeldet wird. Die Überprüfung erfolgt durch einen Administrator. Dieser kann den betroffenen Benutzer jederzeit sperren.
- Ein Administrator kann einen gesperrten Fotografen nach einiger Zeit wieder entsperren.
- Ein Administrator kann einen beliebigen aktiven (nicht gesperrten) Benutzer jederzeit zum Administrator ernennen.
- Ein Administrator kann sein Amt niederlegen und somit zum Aktiven Benutzer werden. Dies ist jedoch nur möglich, falls davor noch mindestens drei Administratoren im System registriert sind.
- Administratoren und gesperrte Benutzer, sowie Benutzer die sich in Prüfung befinden, können sich nicht vom System abmelden.

Hinweis: Überlegen Sie sich auch, welche Parameter Sie Ihren auf den Transitionen stehenden Ereignissen mitgeben.

Thema Nr. 2

Teilaufgabe 1

1. Modellierung

Das Fitnessstudio „Bavariogym“ plant zur Unterstützung des Trainings eine App für ihre Mitglieder. Betrachten Sie dazu folgendes Szenario:

- Zu einem Mitglied werden Vorname, Nachname, Eintrittsdatum und der Tarif gespeichert.
 - Im Fitnessstudio arbeiten speziell ausgebildete Fitnesstrainer. Deren Aufgabe ist es, Trainingspläne zu erstellen. Ein Trainer besitzt immer eine eindeutige Personal-Nummer. Gehen Sie davon aus, dass ein Trainer kein Mitglied ist.
 - Ein Trainingsplan ist nur in einem bestimmten Zeitraum gültig. Sehen Sie hierzu zwei Daten für die zeitlichen Grenzen vor. Um einen Trainingsplan zu identifizieren soll eine fortlaufende Nummer verwendet werden.
 - Jeder Trainingsplan schreibt das Training an unterschiedlichen Geräten vor. Dabei sollen jeweils die Anzahl der Wiederholungen pro Satz, die Anzahl der Sätze und die Gewichtseinstellung berücksichtigt werden.
 - Ein Mitglied kann nach mehreren Trainingsplänen gleichzeitig trainieren. Ebenso findet ein Trainingsplan bei mehreren verschiedenen Mitgliedern Anwendung.
 - Ein Gerät besitzt einen eindeutigen Namen, sowie eine Information über die Muskelgruppe, die trainiert wird.
- a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
- Die Entity-Typen, die Relationship-Typen und jeweils deren Attribute.
 - Die Funktionalitäten der Relationship-Typen.
- b) Wie würde sich ihr Modell aus Teilaufgabe a) ändern, wenn die Trainer den Mitgliedern individuelle, jedoch nicht exklusive Trainingspläne anfertigen? Zeichnen Sie ein aktualisiertes ER-Modell in Chen-Notation. Bestimmen Sie hierzu:
- Die Entity-Typen, die Relationship-Typen, jeweils **ohne** deren Attribute.
 - Die Funktionalitäten der Relationship-Typen.
- c) Wie würden Sie vorgehen, um ihr Modell aus Teilaufgabe a) so zu ändern, dass Sie für jede Person im Fitnessstudio den Vornamen, den Nachnamen, die Straße, die Hausnummer, die PLZ und den Ort speichern können. Beschränken Sie ihre Antwort auf maximal drei Sätze.

Fortsetzung nächste Seite!

2. SQL-Anfragen

Gegeben ist das folgende relationale Schema:

EQUIPE {[id, name]}

FAHRER {[id, name, equipe_id]}

ETAPE {[id, date, NbKm]}

TEILNAHME {[fahrer_id, etape_id, zeit]}

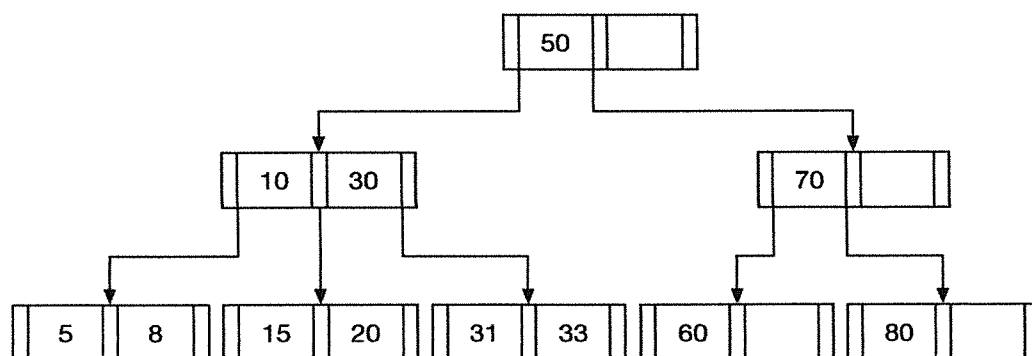
Zur Modellierung der Tour de France im letzten Jahr (2014) wird folgendes relationales Schema verwendet. EQUIPE enthält Informationen zur Mannschaft (der EQUIPE), zu denen jeder Fahrer eindeutig zugeordnet ist. FAHRER enthält die Informationen zum Fahrer mit dem Fremdschlüssel equipe_id. ETAPE enthält die Information zu den Etapen, die gefahren werden, das Attribut NbKm ist die Länge der Etape in km. TEILNAHME stellt die Verbindung von Etapen und Fahrern her, das Attribut zeit ist die individuelle Zeit eines Fahrers fahrer_id in der Etape etape_id. Fällt der Fahrer für die Etape id aus, so erscheint er in der Relation TEILNAHME für dieses etape_id nicht.

Formulieren Sie folgende Anfragen in SQL. Achten Sie falls nötig auf duplikatfreie Ausgaben:

- Wie lang war die Tour de France letztes Jahr?
- Geben Sie die Namen der Fahrer der Equipe 'RENNSTALL' aus.
- Geben Sie die Namen der Fahrer an, die an allen Etapen teilgenommen haben.
- Geben Sie die Gesamtlaufzeit aller Läufer der Equipe 'RENNSTALL' in der 1. Etape (ETAPE.id=1) aus.

3. Indexstrukturen

Gegeben ist der folgende B-Baum der Ordnung 3 (max. drei Kindknoten, max. zwei Schlüssel pro Knoten):



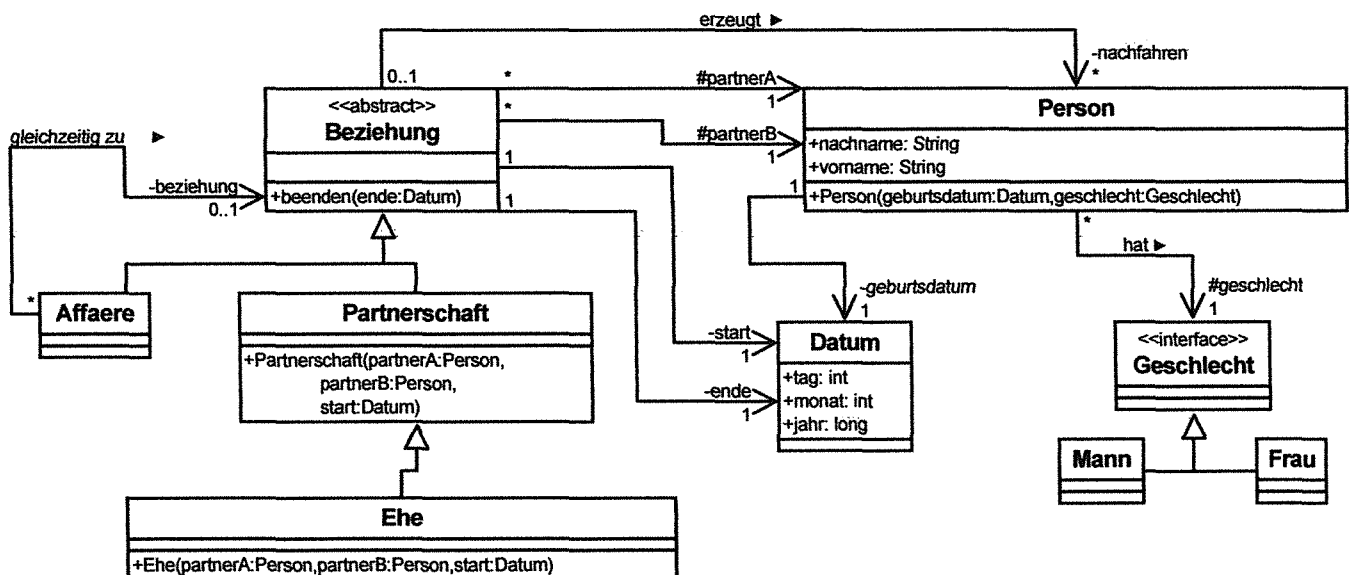
Fortsetzung nächste Seite!

- Fügen Sie die Werte 9 und 45 ein. Löschen Sie anschließend die Werte 30 und 70. Zeichnen Sie den Baum nach jeder Einfüge- bzw. Lösch-Operation.
- Geben Sie das Vorgehen für exakte Suche und Bereichssuche im B+-Baum an.
- Geben Sie einen Vorteil und einen Nachteil von B+-Bäume im Vergleich zu Hash-Tabellen an.

Teilaufgabe 2

Aufgabe 1: „OOD/OOP“

Gegeben sei folgendes UML-Klassendiagramm:

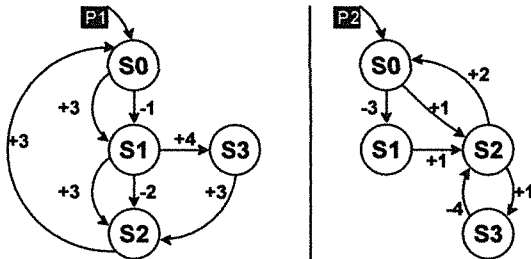


Geben Sie eine Implementierung dieses (vereinfachten) Klassendiagramms in einer gängigen und geeigneten objekt-orientierten Programmiersprache Ihrer Wahl an. Achten Sie dabei insbesondere auf die richtige Wiedergabe der Sichtbarkeiten und Multiplizitäten. Die Methoden- und Konstruktor-Rümpfe dürfen Sie leer lassen, sofern das Programm dann trotzdem noch übersetzbar ist (ggf. muss bei Konstruktoren ein gültiger **super**-Aufruf eingefügt werden).

Fortsetzung nächste Seite!

Aufgabe 2: „FSM und PN“

Betrachten Sie folgendes Modell eines einfachen Kommunikationsprotokolls:

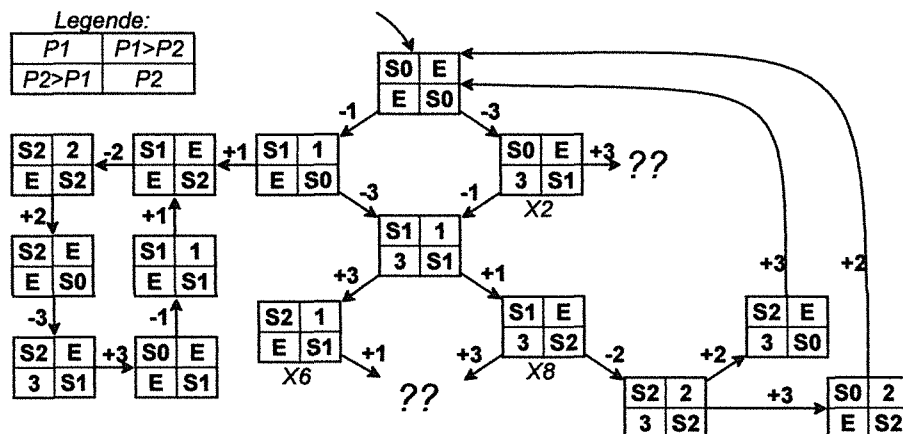


Die Prozesse P1 und P2 arbeiten asynchron und senden sich gegenseitig Nachrichten über einen Kanal mit genau einer Speicherstelle (Puffer) pro Übertragungsrichtung. Die Kantenbeschriftung ist wie folgt zu lesen:

- „-1“:= P1 sendet Nachricht „1“ an P2
- „+2“:= P2 empfängt Nachricht „2“ von P1

Von außen werden keine Nachrichten an die beiden Automaten gesendet.

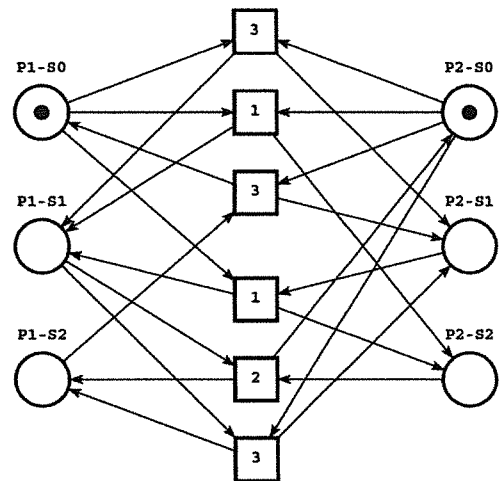
- a) Ergänzen Sie alle fehlenden Zustände im folgenden Erreichbarkeitsgraphen des sich ergebenden Produktautomaten. Übernehmen Sie dafür nur die benannten Zustände „X2“, „X6“ bzw. „X8“ in Ihr Arbeitsblatt und vervollständigen Sie die davon ausgehenden Teilgraphen.



- b) Beantworten Sie folgende Fragestellungen und begründen Sie jeweils kurz Ihre Antwort:
- Ist der Startzustand jederzeit direkt oder indirekt wiederherstellbar?
 - Können beide Prozesse ausgehend vom Startzustand alle ihre individuellen Zustände einnehmen?
 - Enthält das System Verklemmungssituationen (Deadlocks)?
 - Wenn Deadlocks auftreten können, beschreiben Sie kurz die Situation, die jeweils vorliegt.

Fortsetzung nächste Seite!

- c) Nach Analyse des Produktautomaten modelliert jemand das nebenstehende Petri-Netz und behauptet, es zeigt genau das gleiche Verhalten der beiden Prozesse, wenn man die Platzmarkierung als Zustandszeiger und das Schalten einer Transition als Übertragung der gleichnamigen Nachricht interpretiert. Nehmen Sie kurz aber nachvollziehbar Stellung zu dieser Behauptung.



Aufgabe 3 „Formale Verifikation“

```
long doubleFac(long n) {
    /* P */ long df = 1;
    for (long x = n; x > 1; x -= 2) {
        df *= x;
    } /* Q */
    return df;
}
```

$P := n \geq 0$

$$Q := df = n!! := \begin{cases} 2^k \cdot k! & \text{für gerade } n, \text{ dabei sei } k := \frac{n}{2} \\ \frac{(2k)!}{2^k \cdot k!} & \text{für ungerade } n, \text{ dabei sei } k := \frac{n+1}{2} \end{cases}$$

Zur Vereinfachung nehmen Sie im Folgenden an, dass die verwendeten Datentypen unbeschränkt sind und daher keine Überläufe auftreten können!

- a) Welche der folgenden Bedingungen ist eine zum Beweisen der Korrektheit der Methode mittels *wp*-Kalkül (Floyd-Hoare-Kalkül) sinnvolle Schleifeninvariante?

- $df = n!! - x!! \wedge x \geq 1$
- $df = (n - x)!! \wedge x \geq 1$
- $df \cdot x!! = n!! \wedge x \geq 0$
- $(df + x)!! = n!! \wedge x \geq 0$

Fortsetzung nächste Seite!

- b) Zeigen Sie *formal* mittels *wp*-Kalkül, dass die von Ihnen gewählte Bedingung unmittelbar vor Beginn der Schleife gilt, wenn zu Beginn der Methode die Anfangsbedingung P gilt.
- c) Zeigen Sie *formal* mittels *wp*-Kalkül, dass die von Ihnen gewählte Bedingung tatsächlich eine Invariante der Schleife ist.
- d) Zeigen Sie *formal* mittels *wp*-Kalkül, dass am Ende der Methode die Nachbedingung Q erfüllt wird.
- e) Beweisen Sie, dass die Methode immer terminiert. Geben Sie dazu eine Terminierungsfunktion an und begründen Sie kurz Ihre Wahl.