
Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
---------------------------	-----------------------	-----------------------------

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Herbst
2012**

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

Sie sollen ein System zur Verwaltung von Einkaufszentren entwerfen. Gehen Sie dabei von folgendem Szenario aus:

- Einkaufszentren haben einen eindeutigen Namen und eine Adresse. Sie bestehen aus mehreren Etagen. Jede Etage hat eine Nummer (z. B. „2. Etage“). Diese ist jedoch nur innerhalb eines Einkaufszentrums eindeutig. In jeder Etage gibt es Räume. Räume werden durch eine Raumnummer identifiziert, die auf einer Etage eindeutig ist (z. B. „Raum 7 der 2. Etage“). Räume haben eine Fläche. Außerdem können Räume neben anderen Räumen liegen. Räume, die nebeneinander liegen, befinden sich in demselben Einkaufszentrum in derselben Etage.
- Ein Einkaufszentrum hat Läden. Der Name eines Ladens ist innerhalb eines Einkaufszentrums eindeutig. Für jeden Laden wird die Verkaufsfläche gespeichert. Jeder Laden belegt ein oder mehrere Räume des Einkaufszentrums, in dem er sich befindet. Es gibt keine Läden, die sich einen Raum teilen.
- Es gibt Warengruppen. Diese haben einen eindeutigen Namen (z. B. „Handtaschen“ oder „Schuhe“). Läden können einer beliebigen Anzahl Warengruppen zugeordnet werden.
- Für das Personal, welches sich in Verkäufer und Reinigungskräfte aufteilt, werden ihre eindeutige Personalnummer, ihr Name und ihre Adresse gespeichert. Verkäufer können Auszeichnungen erhalten (z.B. „Verkäufer des Monats November 2010“). Für jede Auszeichnung werden Name und Datum angegeben (z. B. „Verkäufer des Monats“ und „November 2010“). Es kann vorkommen, dass sich mehrere Verkäufer eine Auszeichnung teilen müssen.
- Läden beschäftigen Personal, d. h. Verkäufer und Reinigungskräfte. Einkaufszentren beschäftigen nur Reinigungskräfte, aber keine Verkäufer.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell.

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, und
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

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.

Fortsetzung nächste Seite!

2. Anfragen

Gegeben seien die folgenden Relationen:

Menschen : {[MenschId, Name, Alter]}

Berufe : {[BerufId, Name]}

HatBeruf : {[MenschId, BerufId]}

Ein Mensch kann mehrere Berufe haben.

FanVon : {[IdolId, FanId]}

Ein Idol kann mehrere Fans haben.

Idole und Fans sind Menschen.

- a) Formulieren Sie eine SQL-Anfrage, die *MenschId* und *Name* des Rockmusikers (bzw. der Rockmusiker) mit den meisten Fans zurückgibt.
Hinweis: Sie dürfen eigene Views definieren.
- b) Geben Sie einen Ausdruck in relationaler Algebra an, der *MenschId* und *Name* der Fußballspieler zurückliefert, die weniger als 20 Jahre alt sind und mindestens einen Fan haben, der älter ist als 60 Jahre.
- c) Formulieren Sie einen Ausdruck im relationalen Tupelkalkül, der *BerufId* und *Name* derjenigen Berufe zurückgibt, bei denen *jeder*, der sie ausübt, älter ist als 40 Jahre.
Hinweis: Der Ausdruck darf auch unsicher sein.
- d) Geben Sie einen Ausdruck im relationalen Domänenkalkül an, der alle Paare von *MenschIds* von Menschen zurückliefert, die gegenseitig voneinander Fans sind. Der *Name* des Menschen, dessen *MenschId* zuerst genannt wird, soll dabei (als String) kleiner oder gleich dem Namen des Menschen sein, dessen *MenschId* als zweites genannt wird.
Hinweis: Der Ausdruck darf auch unsicher sein.
- e) Formulieren Sie eine Datalog-Anfrage, die *MenschId* und *Name* von allen Menschen zurückgibt, die Fans von einem Boxer und von einem Ballettänzer sind.

Fortsetzung nächste Seite!

3. Normalformen und Anomalien

Gegeben sei die Relation R mit Schema $R: \{[A, B, C, D, E]\}$.

Für die Relation R gelten die folgenden Funktionalen Abhängigkeiten:

- $A \rightarrow B$
- $A, C \rightarrow D$
- $B \rightarrow E$

- a) Bestimmen Sie alle Schlüsselkandidaten von R .
- b) Prüfen Sie, ob R in 1NF, 2NF, 3NF bzw. BCNF ist. Begründen Sie Ihre Antwort.
- c) Zeigen Sie an einer geeigneten Instanz, dass in der Relation R eine Update-Anomalie auftreten kann.
Hinweis: Beachten Sie, dass Ihre Instanz alle Funktionalen Abhängigkeiten aus der Aufgabenstellung erfüllt.
- d) Betrachten Sie nacheinander die Normalformen 1NF, 2NF, 3NF und BCNF. Gibt es eine Relation, die sich in der betrachteten Normalform befindet und in der eine Update-Anomalie auftreten kann (ohne Begründung)?

Fortsetzung nächste Seite!

Teilaufgabe 2:

Sie sollen für eine Online-Apotheke eine Software zur Verwaltung des Medikamentenbestandes implementieren. Zu jedem Medikament gehört eine Nummer, ein Markenname, ein Hersteller, ein Lieferant, ein Verfallsdatum, ein Einzelpreis und sein Lagerbestand (Anzahl der Einheiten). Dasselbe Medikament kann mit unterschiedlichen Verfallsdaten vorhanden sein. Dazu kommt die Zusatzinformation, etwa ob und in welchem Umfang es vorbestellt wurde.

Es soll möglich sein, den Lagerbestand aufzufüllen und zu reduzieren (zum Beispiel beim Verkauf eines Medikaments). Es soll möglich sein, alle Medikamente, deren Verfallsdatum überschritten ist, zu entfernen. Außerdem soll es möglich sein, Aufträge an Lieferanten zu erstellen.

Die beteiligten Personen können Kunden, Apotheker oder Administratoren sein; der Einfachheit halber können Apotheker und Administratoren nicht Kunden sein.

Beteiligte Personen können sich beim System an- und wieder abmelden.

Kunden können Medikamente kaufen und bezahlen. Außerdem können Kunden den Bestand durchsuchen und erhalten dann für jedes geführte Medikament die Mitteilung, ob es verfügbar ist, oder nicht. In letzterem Falle können sie es vorbestellen. Kunden können aber nicht Stückzahlen oder Verfallsdaten einsehen.

Apotheker können Medikamente verkaufen, den Bestand durchsuchen, verändern und Aufträge an Lieferanten tätigen. Außerdem können sie Kunden neu aufnehmen oder löschen.

Administratoren haben alle Befugnisse von Apothekern und können außerdem Apotheker und Administratoren aufnehmen oder löschen.

- a) Entwickeln Sie aus der gegebenen Anwendungsbeschreibung ein UML-Klassendiagramm. Alle Angaben aus dem Text sollen sich in Ihrem Modell wiederfinden; weitere sollten nur dann hinzugenommen werden, wenn es aufgrund der von Ihnen gewählten Modellierung erforderlich erscheint.
Verwenden Sie Vererbung und Interfaces, wo es sinnvoll möglich ist und achten Sie auf eine angemessene Darstellung von Assoziationen und Aggregationen, wo erforderlich. Es geht hier nur um das UML-Klassendiagramm; insbesondere ist eine Modellierung als relationale Datenbank *nicht* verlangt und auch nicht im Sinne der Aufgabenstellung.
- b) Modellieren Sie den gesamten Anwendungsfall des Kaufens eines Medikaments (Anmelden, Bestand anzeigen, Aussuchen, Kaufen, Abmelden) zunächst als UML-Anwendungsfall und dann als UML-Aktivitätsdiagramm.
Achten Sie auch auf die Möglichkeit des Fehlschlagens einer Teilaktivität.

Fortsetzung nächste Seite!

Aufgabe 2:

Das Entwurfsmuster *Beobachter* (Observer) kommt zum Einsatz, wenn auf Änderungen einer Datenstruktur an verschiedenen Stellen im Programm reagiert werden soll.

- a) Beschreiben Sie eine mögliche konkrete Anwendung dieses Musters.
- b) Geben Sie ein allgemeines UML-Klassendiagramm für dieses Entwurfsmuster an.
- c) Skizzieren Sie, wie die beteiligten Klassen und Interfaces in Java oder einer ähnlichen Sprache implementiert werden können.
- d) Nennen Sie zwei weitere Entwurfsmuster und beschreiben Sie ihre Einsatzmöglichkeiten in jeweils ca. drei Sätzen.

Thema Nr. 2

Teilaufgabe 1:

- 1 Allgemeinwissen
 - 1.1 Erläutern Sie kurz die folgenden Begriffe im Kontext von föderierten Datenbanksystemen.
 - a) Komponentenschema
 - b) Lokales Schema
 - c) Föderiertes Schema
 - d) Externes Schema
 - e) Exportschema
 - 2 Relationale Algebra
 - 2.1 Nennen Sie die fünf Grundoperationen der relationalen Algebra.
 - 2.2 Warum wird in gewöhnlichen Datenbanken die relationale Algebra nicht strikt eingehalten? Geben Sie ein Beispiel an und erklären Sie wie man die relationale Eigenschaft erzwingen kann.
 - 3 ER-Modellierung und Relationenmodell
 - 3.1 Nennen und erläutern Sie kurz vier Beispiele für Integritätsbedingungen.
 - 3.2 Sie sollen die Verwaltung von TÜV-Stützpunkten modellieren. Erstellen Sie 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.
- In der Formulierung der Beziehungen sind die Kardinalitätseinschränkungen genau zu beachten. Im Allgemeinen gilt: alles was nicht explizit eingeschränkt ist, muss im Modell auch uneingeschränkt bleiben. (Keine Einschränkungen in eine Beschreibung hineininterpretieren). Alles was explizit eingeschränkt ist, darf im Modell nicht uneingeschränkt bleiben. (Alle formulierten Einschränkungen sollen erkennbar sein).
1. Es gibt TÜV-Stationen. Diese sind an einer Adresse ansässig und besitzen einen Namen. Eine TÜV-Station ist offiziell unter einer eindeutigen ID registriert.
 2. Jede TÜV-Station ist in Abteilungen organisiert. Eine Abteilung hat einen Namen und eine Spezialisierung (z.B. "Bremsen", "Flüssigkeiten", etc.). Eine Abteilung soll durch einen künstlichen Schlüssel (ID) eindeutig identifiziert werden.
 3. Es gibt Angestellte, die über ihre Sozialversicherungsnummer (SVN) eindeutig identifiziert sind. Angestellte haben einen Vor- und einen Nachnamen. Außerdem sind sie irgendwann geboren und haben ein Geschlecht. Darüber hinaus ist ihnen ein Gehalt sehr wichtig.
 4. Prüfsingenieure besitzen eine eindeutige Lizenznummer. In der Regel haben sie ein Spezialisierungsfach. Darüber hinaus können sie eine Zusatzqualifikation (Sicherheitsingenieur etc.) besitzen.

Fortsetzung nächste Seite!

5. Prüflingenieur sind Angestellte: Ein Angestellter kann ein Prüflingenieur sein; ein Prüflingenieur ist stets ein Angestellter.
 6. Eine Abteilung beschäftigt beliebig viele Angestellte. Ein Angestellter kann in einer oder mehreren Abteilungen arbeiten.
 7. Jeder Prüflingenieur ist in genau einer TÜV-Station tätig. Eine Station kann mehrere Prüflingenieure beschäftigen.
 8. Keine TÜV-Station ohne Fahrzeuge: Sie werden durch die Fahrgestellnummer identifiziert und es werden weitere Daten wie Hersteller, Typ und Baujahr gespeichert.
 9. Ein Prüflingenieur kann mehrere Fahrzeuge untersuchen.
 10. Eine TÜV-Station muss mindestens eine Abteilung haben. Jede Abteilung gehört zu genau einer TÜV-Station.
 11. "Gebühren": Prüfungsleistungen werden in Rechnung gestellt. Zu jeder Leistung speichern wir die Beschreibung der Leistung und den jeweiligen Preis. Identifiziert werden sie durch Leistungsnummer-Codes (nach DIN). Rechnungen haben eine eindeutige Rechnungsnummer. Jede Rechnung umfasst mindestens eine Leistung.
 12. Angestellte leiten andere Angestellte
 13. Einem Chef können andere Angestellte untergeben sein.
 14. Jeder Angestellte ist Untergebener von maximal einem Angestellten.
- 3.3 Erstellen Sie zu dem E/R-Diagramm aus Aufg. 3.2 ein möglichst vollständiges Relationenschema – bspw. Berücksichtigung totaler Partizipationen. Vermeiden Sie unnötiges Ausprägen von Relationen bei allen Beziehungen.

Beispiel für die Notation:

Relationenname (Primärschlüssel, Attribut1, Attribut2, ...,
 Fremdschlüssel [AndereRelation], (FKAttribA, FKAttribB) [DritteRelation])
 Attribut2 NOT NULL

4 Normalformen

- 4.1 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.
- 4.2 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.

AbtNr	Name	Strasse	Ort	PLZ	Vorwahl
1	Entwicklung	Abbestr.	München	80999	089
2	Forschung	Allacher Str.	München	80997	089
3	Entwicklung	Zieglhof	Regensburg	93055	0941
4	Fertigung	Bucher Hauptstr.	Nürnberg	90427	0911
5	Logistik	Osserweg	Passau	94034	0851
6	Vertrieb	Alfred-Nobel-Str.	Würzburg	97080	0931

Fortsetzung nächste Seite!

5 Schlüsselzugriffe

- 5.1 Wozu ist die Möglichkeit des Satzzugriffes über Schlüssel wünschenswert?
- 5.2 Grenzen Sie die Begriffe Primärorganisation und Sekundärorganisation voneinander ab.
- 5.3 Gegeben ein leerer B-Baum (maximale Anzahl der Einträge pro Knoten ist $2k$, $k=1$). Die Zahlen 13, 17, 21, 11, 5, 15 und 12 sind in dieser vorgegebenen Reihenfolge in den B-Baum einzufügen. Löschen Sie dann den Schlüssel 15. Zeichnen Sie die Zustände des B-Baums nach jeder Strukturänderung. Falls sich durch eine Operation die Struktur des Baumes nicht ändert, muss er nicht neu gezeichnet werden, sondern der einzufügende Wert kann einfach in den Baum eingetragen werden. Wenn eine Operation mehrere Strukturänderungen erfordert, so ist der Baum nach jeder Änderung neu zu zeichnen.
- 5.4 Nennen Sie zwei Unterschiede zwischen B-Baum und B*-Baum sowie je einen Vor- und Nachteil beider Varianten.

Fortsetzung nächste Seite!

Teilaufgabe 2:**1 Objektorientierung**

Ein Getränkelieferservice verwaltet die Bestellungen verschiedener Kunden. Die folgenden Teilaufgaben sind in einer objektorientierten Programmiersprache zu lösen (die verwendete Sprache ist vorab anzugeben.).

1. Implementieren Sie eine Klasse **Kasten** zur Beschreibung eines Getränkekastens mit den folgenden Eigenschaften. Entscheiden Sie dabei jeweils ob eine Realisierung als Objekt- oder Klassenfeld sinnvoll ist.

- Es existiert ein einheitliches Kastenpfand in Höhe von 1,50 Euro.
- Für alle Flaschen in einem Kasten gelte ein einheitliches Flaschenpfand, das jedoch von Kasten zu Kasten verschieden sein kann.
- Während das Flaschenpfand für alle Flaschen eines Kastens gleich ist, sind die Einzelpreise der Flaschen je nach Inhalt unterschiedlich. Die Einzelpreise (ohne Flaschenpfand) der im Kasten enthaltenen Flaschen sollen in einem 2-dimensionalen Array abgelegt werden.

Geben Sie für die Klasse **Kasten** einen geeigneten Konstruktor an.

Ergänzen Sie in der Klasse **Kasten** eine Objektmethode zur Berechnung des Gesamtpreises des Getränkekastens inklusive Kasten- und Flaschenpfand. (12 Punkte)

2. Schreiben Sie eine Klasse **Bestellung**. Jeder Bestellung soll eine eindeutige Bestellnummer zugeordnet werden, die über den Konstruktoraufwurf erstellt wird. Außerdem soll zu jeder Bestellung der Name des Kunden gespeichert werden, sowie eine einfach verkettete Liste der bestellten Getränkekästen. Die Klasse **Bestellung** soll weiterhin eine Methode beinhalten, die den Gesamtpreis der Bestellung ermittelt. (10 Punkte)
3. Schreiben Sie ein kleines Testprogramm, das eine Bestellung erstellt, die zwei Getränkekästen umfasst. Der erste Kasten soll ein 1 x 1 Getränkekasten mit einer Flasche zu 0,75 Euro sein, der zweite Kasten soll - wie in Abbildung 1 dargestellt - ein 3 x 3 Getränkekasten mit 3 Flaschen zu 0,7 Euro auf der Diagonalen und 3 weiteren Flaschen zu je 1 Euro sein. Das Flaschenpfand beider Kästen beträgt 0,15 Euro pro Flasche, das Kastenpfand 1,50 Euro. Anschließend soll der Preis der Bestellung berechnet und auf der Standardausgabe ausgegeben werden. (8 Punkte)

1,0	1,0	0,7
1,0	0,7	0
0,7	0	0

Abbildung 1: Belegung des zweiten Getränkekastens aus Teilaufgabe (3)

Fortsetzung nächste Seite!

2 Projektmanagement

Abbildung 2 stellt ein CPM-Netzwerk dar. Die Ereignisse sind fortlaufend nummeriert (Nummer im Inneren der Kreise) und tragen keine Namen. Gestrichelte Linien stellen Pseudo-Aktivitäten mit einer Dauer von 0 dar.

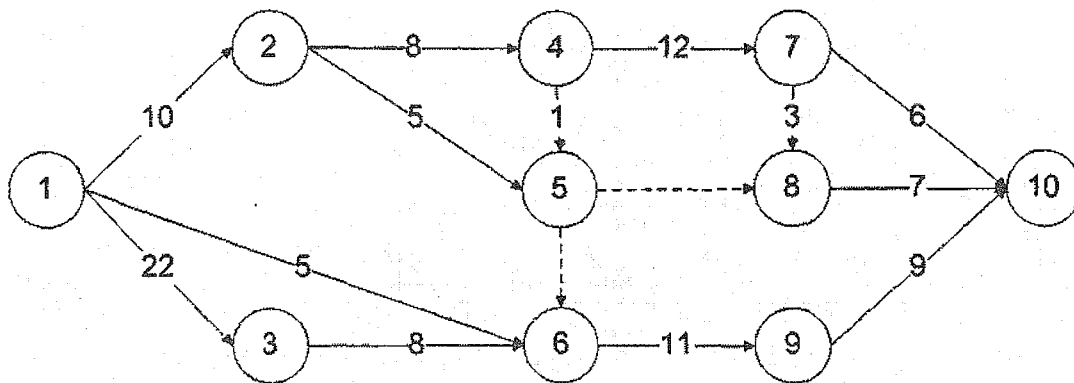


Abbildung 2: CPM-Netzwerk.

1. Berechnen Sie die früheste Zeit für jedes Ereignis, wobei angenommen wird, dass das Projekt zum Zeitpunkt 0 startet! (5 Punkte)
2. Setzen Sie anschließend beim letzten Ereignis die späteste Zeit gleich der frühesten Zeit und berechnen Sie die spätesten Zeiten! (5 Punkte)
3. Berechnen Sie nun für jedes Ereignis die Pufferzeiten! (5 Punkte)
4. Bestimmen Sie den kritischen Pfad! (2 Punkte)
5. Konvertieren Sie das Gantt-Diagramm aus Abbildung 3 in ein CPM-Netzwerk! (10 Punkte)

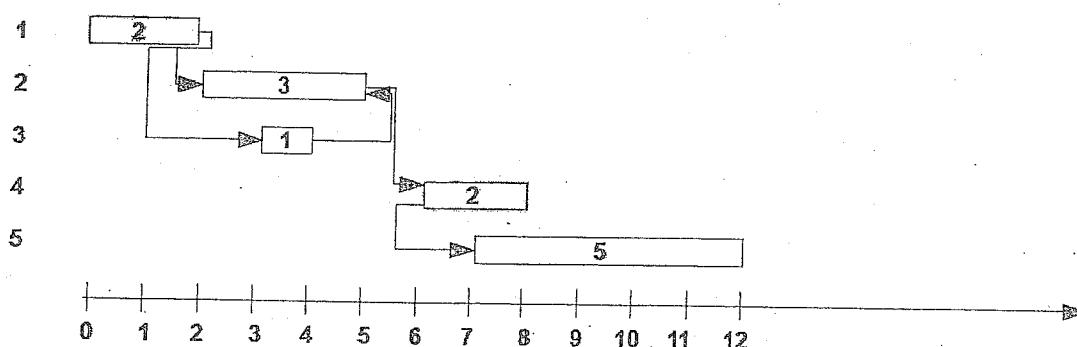


Abbildung 3: Gantt-Diagramm

3 UML-Klassendiagramme

Die UEFA möchte zur kommenden Europa League und Champions League Saison ein neues System zur Verwaltung der entsprechenden Partien einsetzen. Erstellen Sie zu der folgenden Beschreibung ein Klassendiagramm, das Attribute und Assoziationen mit Kardinalitäten sowie die Methoden enthält. Setzen Sie dabei das Konzept der Vererbung sinnvoll ein.

- Ein Fussballspiel wird durch Zeit und Ort beschrieben.
- Ein Spiel findet immer zwischen zwei Teams statt.
- Teams werden durch den Namen beschrieben.
- An einem Spiel sind unterschiedliche Personen beteiligt, die ebenfalls durch einen Namen unterschieden werden:
 - o Schiedsrichter, die sich in Hauptschiedsrichter, Ersatzschiedsrichter und Linienrichter untergliedern
 - o Spieler
 - o Trainer
- Ein Team besteht aus einem Kader von 23 Spielern, wobei die Startelf nur aus 11 Spielern besteht. Pro Team gibt es genau einen Kapitän, der als Spielführer bezeichnet wird.
- Das Team wird von genau einem Trainer betreut.
- Ein Spieler kann eine bestimmte Anzahl an Toren erzielen, die ausgegeben werden kann.
- Jede Person kann rennen.
- Darüber hinaus kann ein Schiedsrichter einen Regelverstoß anzeigen.
- Der Ersatzschiedsrichter kann noch die Tafel heben.
- Der Hauptschiedsrichter kann pfeifen, eine Karte zeigen und nur er kann das Spiel unterbrechen.
- Der Linienrichter kann mit der Fahne winken.
- Spieler können kicken.
- Es gibt eine Spezialform eines Spielers - den Torwart - dieser kann zusätzlich noch fangen.
- Der Trainer kann schreien und einen Spieler für einen anderen Spieler auswechseln.
- Das System erfasst, welcher Spieler für einen anderen Spieler eingewechselt wurde.
- Jeder Schiedsrichter kann Personen ermahnen, aber nur der Hauptschiedsrichter kann Personen auch verwarnen oder des Feldes verweisen. Die ermahnten bzw. verwarnen und des Feldes verwiesenen Personen werden erfasst.

(30 Punkte)