

**Diese Kopfleiste bitte unbedingt ausfüllen!**

Familienname, Vorname (bitte durch eine Leerspalte trennen)



IHK

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
5 5	1 1 9 6		
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14

Termin: Mittwoch, 26. April 2023

# Abschlussprüfung Sommer 2023

1196

1

## Ganzheitliche Aufgabe I Fachqualifikationen

# Fachinformatiker Fachinformatikerin Anwendungsentwicklung (AO 1997)

## 5 Handlungsschritte mit Belegsatzen

90 Minuten Prüfungszeit

## 100 Punkte

## Bearbeitungshinweise

1. Der vorliegende Aufgabensatz besteht aus insgesamt 5 Handlungsschritten zu je 25 Punkten.

**In der Prüfung zu bearbeiten sind 4 Handlungsschritte**, die vom Prüfungsteilnehmer frei gewählt werden können.

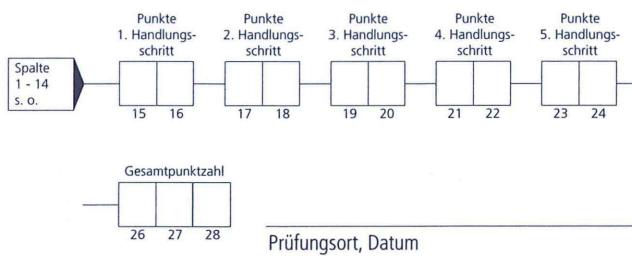
Der nicht bearbeitete Handlungsschritt ist durch Streichung des Aufgabentextes im Aufgabensatz und unten mit dem Vermerk „Nicht bearbeiteter Handlungsschritt: Nr. ...“ an Stelle einer Lösungsniederschrift deutlich zu kennzeichnen. Erfolgt eine solche Kennzeichnung nicht oder nicht eindeutig, gilt der 5. Handlungsschritt als nicht bearbeitet.

2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-Nr. in die oben stehenden Felder ein.
  3. Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
  4. Halten Sie sich bei der Bearbeitung der Aufgaben genau an die **Vorgaben der Aufgabenstellung** zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier Angaben bewertet.
  5. Tragen Sie die frei zu formulierenden **Antworten dieser offenen Aufgabenstellungen** in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
  6. Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
  7. Verwenden Sie nur einen Kugelschreiber und schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
  8. Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
  9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
  10. Für **Hilfsaufzeichnungen** können Sie das in der Tasche beigelegte Konzeptpapier verwenden. Bewertet werden jedoch grundsätzlich nur Ihre Eintragungen in diesem Aufgabensatz.

**Wird vom Korrektor ausgefüllt!**

## Bewertung

**Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen. Für den abgewählten Handlungsschritt ist anstatt der Punktzahl die Buchstabenkombination „AA“ in die Kästchen einzutragen.**



Prüfungszeit  25  
Die entsprechende Ziffer (1, 2 oder 3) finden Sie in der Abfrage nach der Prüfungszeit im Anschluss an die letzte Aufgabe.

## Unterschrift

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen. Hinweis: Im Interesse einer besseren Lesbarkeit wird in der Aufgabenstellung und in den Angaben zur Aufgabenstellung nur die männliche Form (generisches Maskulinum) verwendet. Die verkürzte Sprachform beinhaltet keine Wertung und die gewählten männlichen Formulierungen gelten uneingeschränkt auch für die weiteren Geschlechter.

Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2023 – Alle Rechte vorbehalten

**Die Handlungsschritte 1 bis 5 beziehen sich auf die folgende Ausgangssituation:**

Die IT-Antos AG wird von der Kletterhallenkette McBoulder GmbH beauftragt, ein IT-Projekt umzusetzen.

Sie sollen vier der folgenden fünf Aufgaben innerhalb dieses Projekts erledigen:

1. Aktivitätsdiagramm für Veranstaltungsplanung erstellen
2. Prozedur für maximale Auslastung entwickeln
3. OOP-Software für Rechnung modellieren
4. Normalisierung einer Tabelle durchführen
5. SQL-Anweisungen für Mitgliederverwaltung formulieren

**1. Handlungsschritt (25 Punkte)**

Sie werden von Ihrem Ausbilder beauftragt, den Prozess zur Organisation von Wettkampfveranstaltungen in einem Aktivitätsdiagramm zu dokumentieren. Um den Prozess zu verstehen, sprechen Sie mit mehreren Prozessbeteiligten.

Sie beginnen mit einem Mitarbeiter der Veranstaltungsplanung.

Dieser teilt Ihnen mit, dass er zunächst eine Kletterhalle anfragen muss und diese bei Verfügbarkeit bucht. Ist die Halle nicht frei, wird eine weitere Halle angefragt.

Sobald die Halle reserviert ist, werden die Schrauber mit dem Schrauben der Routen beauftragt, das Anmeldeformular online gestellt und der Link dazu an die Teilnehmer verschickt. Sobald die Routen abgenommen und die Teilnehmer zur Qualifikation eingeladen sind, beginnt die Qualifikation.

Wenn alle Teilnehmer ihre Qualifikationsergebnisse gemeldet haben, werden die Ergebnisse ausgewertet, die qualifizierten Teilnehmer zum Finale eingeladen und die Finalveranstaltung durchgeführt.

Informationen eines Teilnehmers:

Wenn der Teilnehmer den Link erhält, meldet er sich für den Wettkampf an. Nach dem Qualifikationsstart klettert er die Route und meldet seine Ergebnisse dem Veranstalter.

Die Schrauber der Boulder-Routen geben Ihnen folgende Hinweise:

Die Schrauber nehmen den Auftrag des Veranstalters an, schrauben die Routen und lassen diese vom Veranstalter abnehmen.

Erstellen Sie auf der gegenüberliegenden Seite für diesen Ablauf ein entsprechendes Aktivitätsdiagramm.

Routenschrauber	Veranstalter	Teilnehmer	Korrekturrand

## 2. Handlungsschritt (25 Punkte)

Korrekturrand

Das Eintreten (*checkedIN true*) und das Verlassen (*checkedIN false*) der Kletterhalle wird am Eingangs- bzw. Ausgangsdrehkreuz durch das Vorhalten eines RFID-Chips an ein Lesegerät erfasst.

Jeder Vorgang erzeugt einen Eintrag in einer Log-Datei, z.B.:

**date;time;guestID;checkedIN**

**01.04.2023;06:13;123;true**

**01.04.2023;06:17;234;true**

**01.04.2023;06:25;345;true**

**01.04.2023;07:03;456;true**

**01.04.2023;07:10;234;false**

Für die Auswertung wurden alle Einträge bereits in *Check*-Objekte umgewandelt, die in einer Liste *CheckList* vorliegen. Die *CheckList*-Objekte enthalten alle *Check*-Objekte jeweils eines ganzen Kalendertages.

Check	CheckList	KeyValueList
<ul style="list-style-type: none"><li>- date : String</li><li>- time : String</li><li>- guestID: Integer</li><li>- checkedIN : Boolean</li></ul> <ul style="list-style-type: none"><li>+ Check(String, String, Integer; Boolean)</li><li>+ isCheckedIN() : Boolean</li><li>+ getHour() : Integer</li></ul>	<ul style="list-style-type: none"><li>+ get(index : Integer) : Check</li><li>+ size() : Integer</li></ul>	<ul style="list-style-type: none"><li>+ put(Integer, Integer) : void</li></ul>

Schreiben Sie eine Methode, welche die maximale Auslastung (Höchstzahl der gleichzeitig anwesenden Gäste) je Stunde ermittelt. Die Wertepaare (Attribut: Stunde, Wert: maximale Personenzahl) sollen in eine Liste vom Typ *KeyValueList* geschrieben werden. Die Methode bekommt ein *CheckList*-Objekt übergeben und soll ein *KeyValueList*-Objekt zurückgeben.

**Hinweise:** *Check*-Objekte können mittels *get* und *index* aus *CheckList* geholt, Wertepaare mit *put* in die *KeyValueList* geschrieben werden. Die Methode *getHour* gibt die Stunde der Membervariable *time* als ganzzahligen Wert zurück. Die *size*-Methode liefert die Anzahl der Einträge in einer *CheckList*.

Die Darstellung des Quelltextes soll mit einem Struktogramm oder allgemein verständlichem Programm- bzw. Pseudocode erfolgen.

+ getMaxOccupancyPerHour(CheckList) : KeyValueList

Korrekturrand

### 3. Handlungsschritt (25 Punkte)

Korrekturrand

Für die McBoulder GmbH soll eine objektorientierte Fakturierungssoftware entwickelt werden.

- a) Analysieren Sie die folgende Rechnung, indem Sie ein Klassendiagramm mit entsprechenden Klassen, Attributen, Methoden und Beziehungen zwischen den Klassen erstellen. 16 Punkte

## Hinweise:

- Es sollen keine Setter-Methoden modelliert werden. Die Objekte aller Klassen sollen mithilfe des jeweiligen Konstruktors erzeugt werden.
  - Der Kundenname, die Kundenadresse, das Rechnungsdatum und die Artikelbezeichnung sollen als String-Objekte angelegt werden.
  - Die Positionen der Rechnung sollen in einem Array gespeichert werden.
  - Rechnungsobjekte sollen eine Methode *addPosition* zum Hinzufügen einer Position zum Positionenarray und eine Methode *berechneBetrag* haben. Die Methode *berechneBetrag* soll den Rechnungsbetrag zurückgeben.
  - Alle veränderlichen Daten sollen innerhalb eines Rechnungsobjektes zur Verfügung stehen.

# McBoulder

Jochen Meister  
Am Kobel 15  
12345 Hausen

Rechnung: 1234

Kunde: 567

Wir bedanken uns für das entgegengebrachte Vertrauen und stellen Ihnen folgende Lieferung in Rechnung:

Bezeichnung	Anzahl	Preis	Gesamt
Jahresbeitrag	1	316,99 EUR	316,99 EUR
Schuhe Flex	1	158,00 EUR	158,00 EUR
Chalk 500 g	3	10,00 EUR	30,00 EUR
Chalk-Bag	1	18,95 EUR	18,95 EUR
<b>Betrag</b>			523,94 EUR

<b>Bezeichnung</b>	<b>Anzahl</b>	<b>Preis</b>	<b>Gesamt</b>
Jahresbeitrag	1	316,99 EUR	316,99 EUR
Schuhe Flex	1	158,00 EUR	158,00 EUR
Chalk 500 g	3	10,00 EUR	30,00 EUR
Chalk-Bag	1	18,95 EUR	18,95 EUR
<b>Betrag</b>			<b>523,94 EUR</b>

- b) Für den allgemeinen Dokumentdruck soll es unterschiedliche Formatklassen (z. B. mit oder ohne Kopfzeile) geben. Eine erste Implementierung sieht eine Methode *erstelleDokument* in der Klasse Rechnung mit folgendem Verhalten vor:  
+ erstelleDokument(formatTyp : Integer) : Dokument

Korrekturrand

BEGINN

```
Dokument dokument = null;
WENN formatTyp == 1
    DANN dokument = new MitKopfzeile()
SONST WENN formatTyp == 2
    DANN dokument = new OhneKopfzeile()
ENDE WENN
dokument.addEmpfaenger(kunde);
dokument.addInhalt(positionen);
Rückgabe dokument;
```

ENDE

- ba) Beschreiben Sie ein mögliches Problem einer solchen Umsetzung.

2 Punkte

---

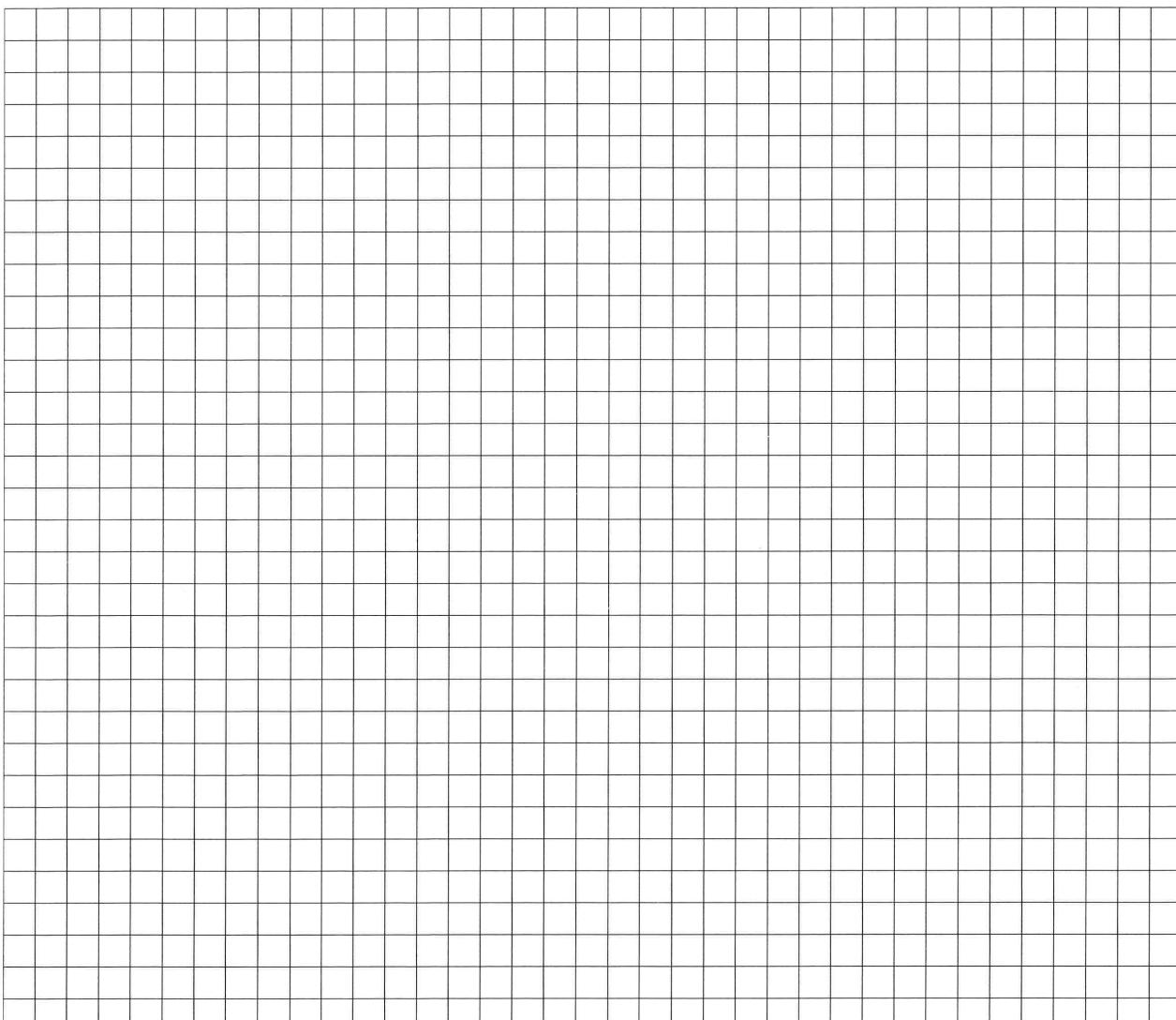
---

---

- bb) Die Dokumentklassen sollen von der Klasse Rechnung vollständig entkoppelt sein.

Entwerfen Sie ein Muster als Klassendiagramm, welches diese Vorgabe umsetzt.

7 Punkte



#### **4. Handlungsschritt (25 Punkte)**

Korrekturrand

Seit kurzer Zeit bietet die McBoulder GmbH in ihren Kletterhallen auch Physiotherapiebehandlungen an.

Die zugehörigen Daten liegen in einer Tabelle vor, welche die Kriterien der 1. Normalform erfüllt:

Leistung Nr (PK)	Leistung Name	Leistung Preis	Behandlung Nr (PK)	Behandlung Datum	Kunde Nr	Kunde Name	Therapeut Nr	Therapeut Name	Halle Nr	Halle Standort
1	Massage	42,50 €	1	23.04.2023	123	Berg	11	Barbie	4	München-Mitte
2	Fango	35,20 €	1	23.04.2023	123	Berg	15	Schütz	4	München-Mitte
3	Gymnastik	33,80 €	2	20.02.2023	345	Klette	11	Helper	9	Nürnberg
2	Fango	35,20 €	2	20.02.2023	345	Klette	11	Helper	9	Nürnberg
1	Massage	42,50 €	3	30.04.2023	123	Berg	22	Barbie	6	München-Ost

Hinweis: Die Attribute *LeistungNr* und *BehandlungNr* bilden zusammen den Primärschlüssel (PK) der Tabelle.

Durch die Befragung von Prozessbeteiligten wurden folgende funktionalen Abhängigkeiten ermittelt:

- LeistungNr → LeistungName, LeistungPreis
- LeistungNr, BehandlungNr → TherapeutNr, TherapeutName
- BehandlungNr → BehandlungDatum, KundeNr, KundeName, HalleNr, HalleStandort
- KundeNr → KundeName
- TherapeutNr → TherapeutName
- HalleNr → HalleStandort

a) Erläutern Sie, warum die gegebene Tabelle schon in der 1. Normalform vorliegt. 2 Punkte

---



---



---



---



---

b) Nennen Sie die Kriterien, welche zur Erreichung der 2. Normalform erfüllt sein müssen. 2 Punkte

---



---



---



---



---

c) Nennen Sie die Kriterien, welche zur Erreichung der 3. Normalform erfüllt sein müssen. 1 Punkt

---



---



---



---



---

d) Entwickeln Sie auf der gegenüberliegenden Seite aus der vorliegenden Tabelle in der 1. Normalform zunächst ein relationales Datenbankmodell in der 2. Normalform und daraus dann ein relationales Datenbankmodell, das die 3. Normalform erfüllt. 20 Punkte

Hinweis: In den Datenmodellen können die Attributwerte weggelassen werden. Zeichnen Sie jeweils alle Beziehungen mit ihren Kardinalitäten ein und kennzeichnen Sie Primärschlüssel mit dem Zusatz (PK) und Fremdschlüssel mit dem Zusatz (FK).

2. Normalform:

A large grid of squares, approximately 20 columns by 20 rows, designed for writing the normal form of a game matrix.

Korrekturrand

3. Normalform:

A large grid of squares, approximately 20 columns by 20 rows, designed for writing the normal form of a game matrix.

## **5. Handlungsschritt (25 Punkte)**

Korrekturrand

Folgende Tabellen stellen einen Auszug aus der Mitgliederdatenbank dar.

**Tabelle: MitgliedArt**

MgA_Id	MgA_Art	Mg_Gebuehr_monatlich
1	gold	45,00
2	silber	35,00
3	bronze	25,00
4	Ehrenmitgliedschaft	0,00
5	Studenten	12,50
...		

**Tabelle: Mitglied**

Mg_Id	Mg_Nachname	Mg_Vorname	Mg_PLZ	Mg_Ort	Mg_Strasse	Mg_Hausnummer	Mg_GebDat	Mg_MgAId	Mg_Geschlecht
1	Peterson	Peter	55678	Köln	Kölner Ringstr.	122	2000-05-01	5	m
2	Ringer	Helga	56767	Köln	Gartenweg	1	1988-02-02	3	w
3	Merten	Manny	56657	Köln	Hauptstr.	1333	1977-04-09	1	m
4	Schulte	Heinrich	56657	Köln	Hauptstr.	178	1973-05-14	2	m
5	Morgen	Sophia	57787	Köln	Perlst.	44	2004-12-16	3	w
8	Koller	Karsten	58765	Köln	Zum Stein	5	1992-05-15	1	m
...									

**Tabelle: Interessent**

Int_Id	Int_Nachname	Int_Vorname	Int_PLZ	Int_Ort	Int_Strasse	Int_Hausnummer	Int_GebDat	Int_Geschlecht
3	Paul	Franziska	57787	Köln	Werthain	33	2005-11-23	w
4	Freier	Friedhelm	56767	Köln	Zum Zaunkönig	444	1999-03-03	m

- a) Erstellen Sie eine SQL-Anweisung, mit der Sie die jährlichen Einnahmen aller männlichen Mitglieder erhalten, gruppiert nach der MitgliedArt und absteigend sortiert nach der Jahresgebühr. 8 Punkte

Beispiel-Ergebnistabelle:

MitgliedArt	AnzahlMitglieder	Jahresgebühr
gold	2	1080,00
silber	1	420,00
Studenten	1	150,00
bronze	0	0,00
Ehrenmitgliedschaft	0	0,00

b) Erstellen Sie eine SQL-Anweisung, mit der Sie das Durchschnittsalter je Geschlecht erhalten.

5 Punkte

Korrekturrand

Beispiel-Ergebnistabelle:

Geschlecht	DurchschnittAlter
m	36
w	26

c) Frau Morgen mit der Mg\_Id = 5 zieht um und wohnt jetzt in der Aachener Str. 12 in 56676 Heinsberg.

Erstellen Sie eine SQL-Anweisung, welche die Daten entsprechend ändert.

5 Punkte

d) Herr Merten mit der Mg\_Id = 3 kündigt seine Mitgliedschaft.

Erstellen Sie eine SQL-Anweisung, welche die Daten von Herrn Merten löscht.

2 Punkte

## **Fortsetzung 5. Handlungsschritt**

## Korrekturrand

- e) Zu einem Event sollen alle Mitglieder und Interessenten angeschrieben werden.

Erstellen Sie eine SQL-Anweisung, mit der Sie über eine Abfrage alle Mitglieder und Interessenten sortiert nach PLZ absteigend und Nachname aufsteigend erhalten. 5 Punkte

## Beispiel-Ergebnistabelle:

Nachname	Vorname	PLZ	Ort	Strasse	Hausnummer	Geschlecht
Koller	Karsten	58765	Köln	Zum Stein	5	m
Paul	Franziska	57787	Köln	Werthain	33	w
Freier	Friedhelm	56767	Köln	Zum Zaunkönig	444	m
Ringer	Helga	56767	Köln	Gartenweg	1	w
Morgen	Sophia	56676	Heinsberg	Aachener Str.	12	w
Schulte	Heinrich	56657	Köln	Hauptstr.	178	m
Peterson	Peter	55678	Köln	Kölner Ringstr.	122	m

**PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!**

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

- 1 Sie hätte kürzer sein können.
  - 2 Sie war angemessen.
  - 3 Sie hätte länger sein müssen.



## Belegsatz

Fachinformatiker Anwendungsentwicklung (AO 1997)

Fachinformatikerin Anwendungsentwicklung (AO 1997)

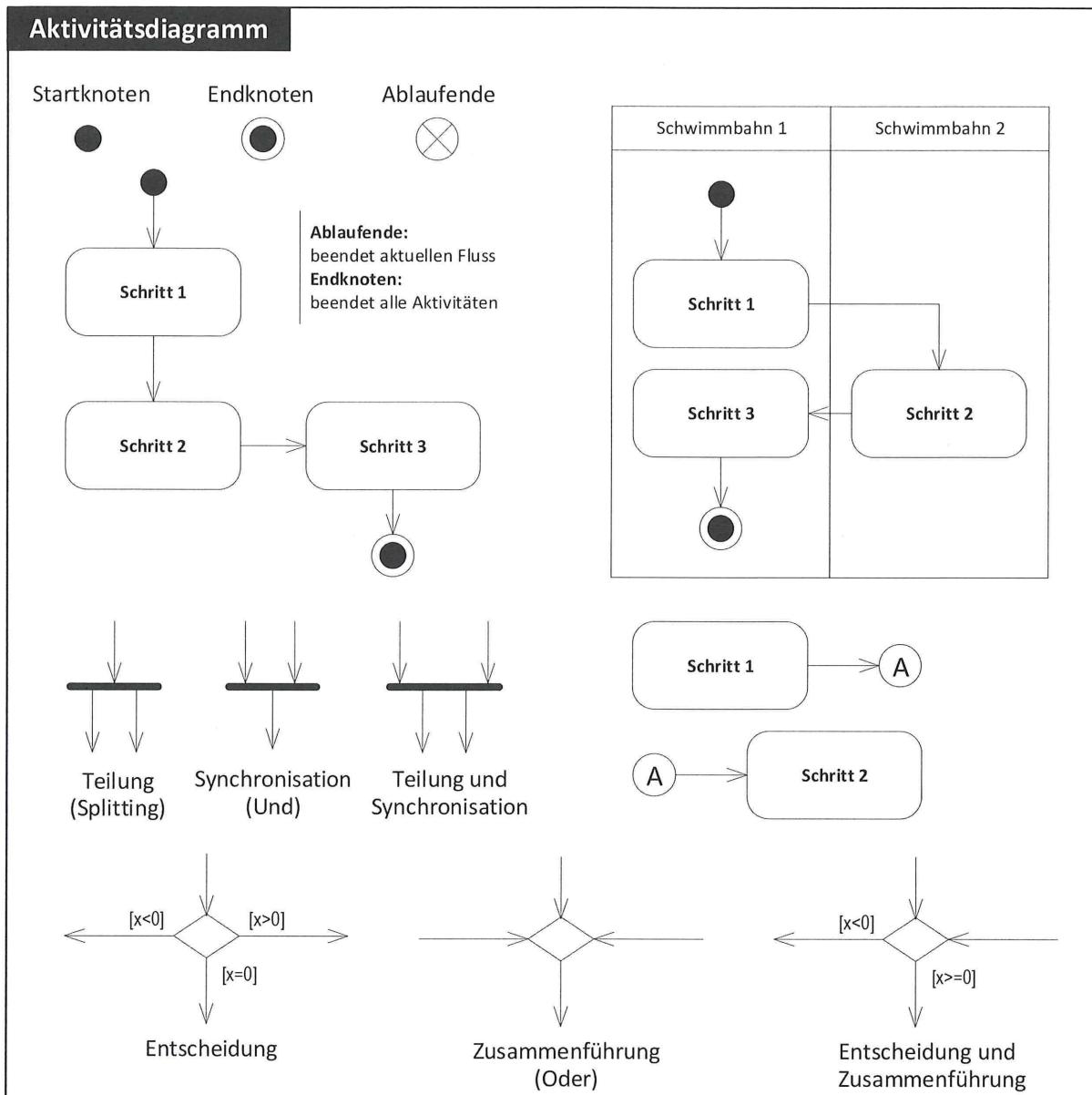
1196

## 1 Ganzheitliche Aufgabe I Fachqualifikationen

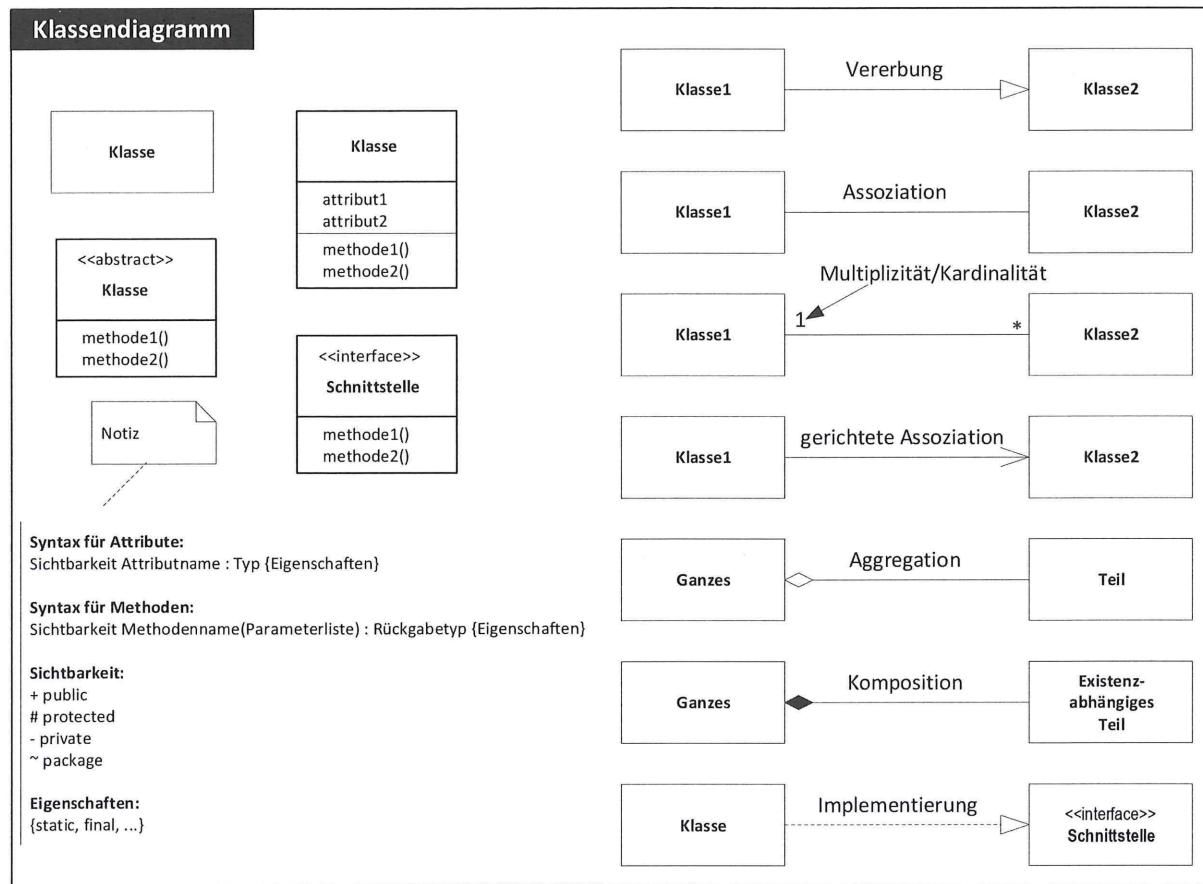
### Inhalt

UML-Aktivitätsdiagramm	2
UML-Klassendiagramm	3
SQL-Syntax (Auszug)	4/5

## UML-Aktivitätsdiagramm



## UML-Klassendiagramm



## SQL-Syntax (Auszug)

Syntax	Beschreibung
<b>Tabelle</b>	
<b>CREATE TABLE</b> Tabellenname( Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
<b>ALTER TABLE</b> Tabellenname <b>ADD COLUMN</b> Spaltenname Datentyp <b>DROP COLUMN</b> Spaltenname Datentyp  <b>ADD FOREIGN KEY</b> (Spaltenname) <b>REFERENCES</b> Tabellenname( Primärschlüsselspaltenname )	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte  Definiert eine Spalte als Fremdschlüssel
<b>CHARACTER</b>	Textdatentyp
<b>DECIMAL</b>	Numerischer Datentyp (Festkommazahl)
<b>DOUBLE</b>	Numerischer Datentyp (Doppelte Präzision)
<b>INTEGER</b>	Numerischer Datentyp (Ganzzahl)
<b>DATE</b>	Datum (Format DD.MM.YYYY)
<b>PRIMARY KEY</b> (Spaltenname)	Erstellung eines Primärschlüssels
<b>FOREIGN KEY</b> (Spaltenname) <b>REFERENCES</b> Tabellenname( Primärschlüsselspaltenname )	Erstellung einer Fremdschlüssel-Beziehung
<b>DROP TABLE</b> Tabellenname	Löscht eine Tabelle
<b>Befehle, Klauseln, Attribute</b>	
<b>SELECT</b> *   Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
<b>FROM</b>	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
<b>SELECT ...</b> <b>(SELECT ...</b> <b>FROM ...</b> <b>WHERE ...)</b> AS xyz	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: xyz) ausgegeben.
<b>FROM ...</b> <b>WHERE ...</b>	
<b>SELECT DISTINCT</b>	Eliminiert Redundanzen, die in einer Ausgabetafel auftreten können, Werte werden jeweils nur einmal angezeigt.
<b>INNER JOIN</b>	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
<b>LEFT JOIN / LEFT OUTER JOIN</b>	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
<b>RIGHT JOIN / RIGHT OUTER JOIN</b>	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
<b>FULL JOIN</b>	Liefert aus beiden Tabellen jeweils alle Datensätze
<b>WHERE</b>	Bedingung, nach der Datensätze ausgewählt werden sollen
<b>WHERE EXISTS ( subquery )</b> <b>WHERE NOT EXISTS ( subquery )</b>	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
<b>GROUP BY</b> Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
<b>ORDER BY</b> Spaltenname1 [,Spaltenname2, ...] <b>ASC   DESC</b>	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder

	ASC: aufsteigend; DESC: absteigend
<b>UNION / UNION ALL</b>	Vereinigung der Ergebnismengen zweier Abfragen / mit Dubletten
<b>Syntax</b>	<b>Beschreibung</b>
<i>Datenmanipulation</i>	
<b>DELETE FROM</b> Tabellenname	Löschen von Datensätzen in der genannten Tabelle
<b>UPDATE</b> Tabellenname <b>SET</b>	Aktualisiert Daten in Feldern einer Tabelle
<b>INSERT INTO</b> Tabellenname <b>VALUES</b> (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder <b>SELECT ... FROM ... WHERE</b>	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
<i>Aggregatfunktionen</i>	
<b>AVG</b> (Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
<b>COUNT</b> (Spaltenname   * )	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
<b>SUM</b> (Spaltenname   Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
<b>MIN</b> (Spaltenname   Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
<b>MAX</b> (Spaltenname   Formel)	Ermittelt den größten aller Werte im angegebenen Feld
<i>Funktionen</i>	
<b>LEFT</b> (Zeichenkette, Anzahlzeichen)	Liefert Anzahlzeichen der Zeichenkette von links.
<b>RIGHT</b> (Zeichenkette, Anzahlzeichen)	Liefert Anzahlzeichen der Zeichenkette von rechts.
<b>CURRENT</b>	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
<b>CONVERT</b> (time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
<b>DATE</b> (Wert)	Wandelt einen Wert in ein Datum um
<b>DAY</b> (Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
<b>MONTH</b> (Datum)	Liefert den Monat aus dem angegebenen Datum
<b>TODAY</b>	Liefert das aktuelle Datum
<b>WEEKDAY</b> (Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
<b>YEAR</b> (Datum)	Liefert das Jahr aus dem angegebenen Datum
<b>DATEADD</b> (Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
<b>DATEDIFF</b> (Datumsteil, Anfangsdatum, Enddatum) Datumsteile: <b>DAY</b> , <b>MONTH</b> , <b>YEAR</b>	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
<i>Operatoren</i>	
<b>AND</b>	Logisches UND
<b>LIKE</b>	Überprüfung von Textattributen auf Gleichheit, Verwendung von Platzhaltern möglich.
<b>NOT</b>	Logische Negation
<b>OR</b>	Logisches ODER
<b>IS</b>	Überprüfung auf NULL
<b>=</b>	Test auf Gleichheit
<b>&gt;, &gt;=, &lt;, &lt;=, &lt; &gt;</b>	Test auf Ungleichheit
<b>*</b>	Multiplikation
<b>/</b>	Division
<b>+</b>	Addition, positives Vorzeichen
<b>-</b>	Subtraktion, negatives Vorzeichen

Stand 2018-03-29