Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

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, 8. Mai 2019

IHK

Abschlussprüfung Sommer 2019

1196

1 Ganzheitliche Aufgabe I
Fachqualifikationen

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung

5 Handlungsschritte
mit Belegsatz
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 **Nebenrechnungen/Hilfsaufzeichnungen** können Sie das im Aufgabensatz enthaltene Konzeptpapier verwenden. Dieses muss vor Bearbeitung der Aufgaben herausgetrennt werden. Bewertet werden jedoch nur Ihre Eintragungen im Aufgabensatz.

Nicht bearbeiteter Handlungsschritt ist Nr. ____

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.

	Punkte 1. Handlungsschritt	Punkte 2. Handlungsschritt	Punkte 3. Handlungsschritt	Punkte 4. Handlungsschritt	Punkte 5. Handlungsschritt
Spalte 1 - 14 s. o.	15 16	17 18	19 20	21 22	23 24

Gesamtpunktzahl 26 27 28

Prüfungszeit 25

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

Prüfungsort, Datum

Unterschrift

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2019 – Alle Rechte vorbehalten!

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

Sie sind Mitarbeiter/-in der rapidPack GmbH.
Die rapidPackGmbH stellt Verpackungsmaschinen her.
Sie wurde von der SoftDrink AG mit der Lieferung einer Verpackungsmaschine beauftragt.

Sie sollen vier der folgenden fünf Aufgaben in diesem Projekt erledigen:
1. Projektstrukturplan und UML-Sequenzdiagram entwerfen
2. Programm zur Auswertung einer Log-Datei anfertigen
3. Objektorientierte Software entwickeln
4. Datenmodell ergänzen
5. SQL-Abfragen formulieren

Beachten Sie die Angaben im Belegsatz 2.

Korrekturrand

1. Handlungsschritt (25 Punkte)

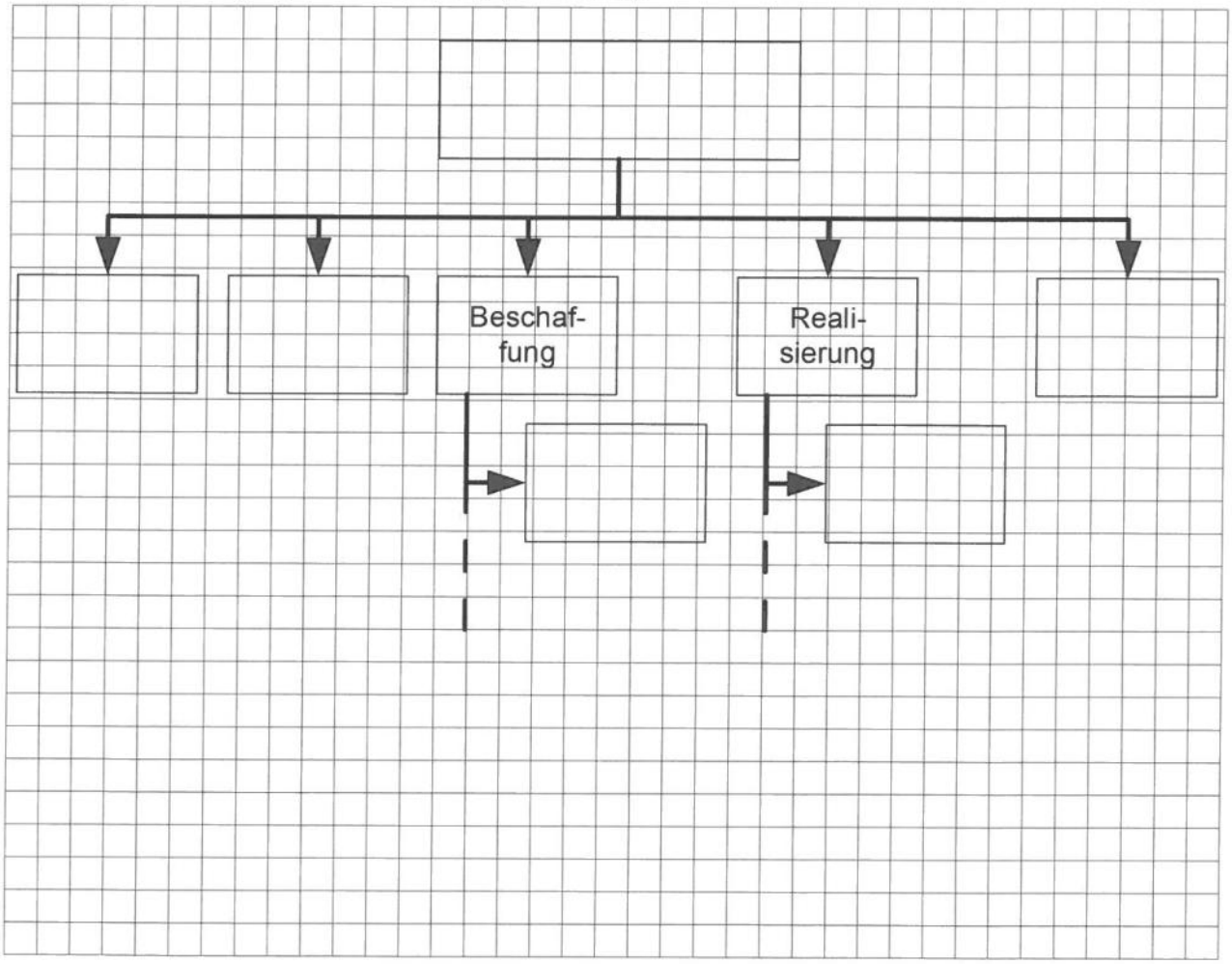
a) Für das Projekt „Maschinensteuerung" sind folgende Vorgänge geplant:

	Vorgang*	Beschreibung
1	Abnahme und Betrieb	System übergeben und betreiben
2	Angebotsvergleich	Angebote vergleichen
3	Bedarfsermittlung	Software-Bedarf ermitteln
4	Beschaffung	Software beschaffen
5	Bestellung	Software bestellen
6	Installation	Software installieren
7	Planung	IT-Systemanpassung planen
8	Probebetrieb	Software testen
9	Realisierung	IT-Systemanpassung durchführen
10	Schulung	Anwender schulen

* Sortierung in alphabetischer Reihenfolge

Ergänzen Sie den folgenden Projektstrukturplan. 5 Punkte

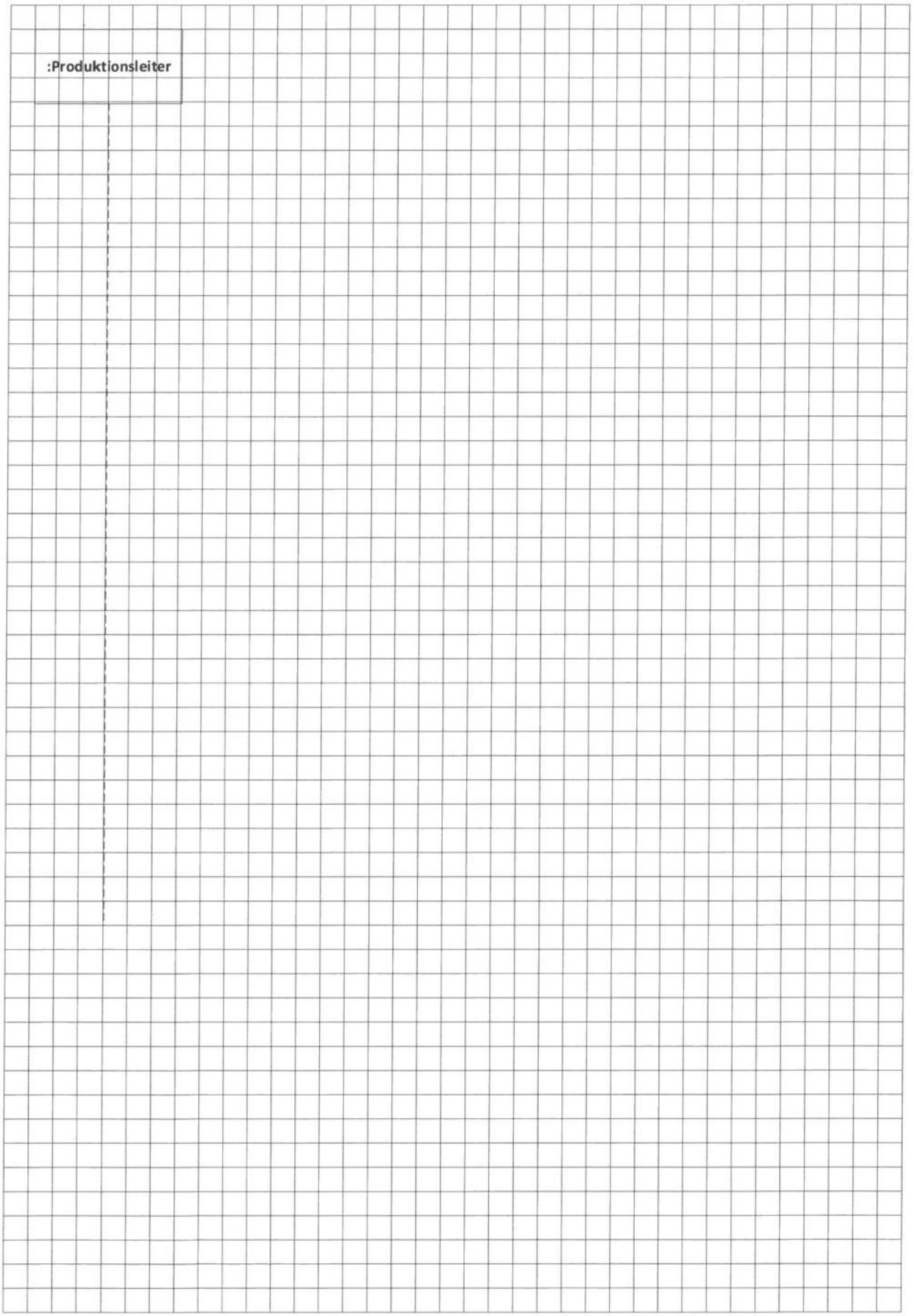
Strukturplan zum Projekt „Maschinensteuerung" (unvollständig)



b) Der Produktionsleiter der SoftDrink AG teilt dem Anwendungsentwickler der rapidPack GmbH mit, dass er für die Verpackungsmaschine ein neues Steuerprogramm entwickeln soll. Der Anwendungsentwickler erstellt das Steuerprogramm und weist den Maschinenführer an, das neue Steuerprogramm mittels eines Probelaufs zu testen. Der Maschinenführer führt den Probelauf durch und bestätigt dem Anwendungsentwickler die Funktionsfähigkeit des Steuerprogramms. Der Anwendungsentwickler wiederum meldet die Produktionsbereitschaft der Maschine dem Produktionsleiter zurück.

Stellen Sie die beschriebene Vorgehensweise in einem UML-Sequenzdiagramm dar. 20 Punkte

Korrekturrand



2. Handlungsschritt (25 Punkte)

Korrekturrand

Log-Datei mit Testdaten auswerten (Array)

Der Test einer Verpackungsmaschine liefert eine Log-Datei, deren Werte in einem Array vom Typ *Messung* vorliegen:

Hinweis: Strukturbeschreibung am Ende der Aufgabenstellung

Array *messung*

	datum	zeit	messArt	istWert	sollWert
0	2019-04-01	06:00:00	1	76	80
1	2019-04-01	06:00:15	2	197	200
...	2019-04-01	06:00:15	0	3,2	3,41
301	2019-04-02		0	76,5	78
302	2019-04-02		0	220	200
...					
426	2019-04-03		2	3,6	3,4
...					

Istwerte, deren prozentuale Abweichung vom Sollwert größer ist als die maximale Toleranz (in Prozent), werden als Fehler gewertet. Diese sollen für jeden Tag in einem Array *tagesProtokoll* vom Typ Integer erfasst werden. Die messArt-Nummer bestimmt die Position im Array.

Beispiel Array tagesProtokoll für den 2019-04-01

messArt	anzahlFehler
0	1
1	0
2	7
...	

Aufgabe:
Entwerfen Sie auf der Folgeseite (als Pseudocode, Struktogramm oder PAP) die Funktion *druckeReport* zur Auswertung des Arrays *messung*, die mithilfe der zur Verfügung stehenden Funktionen einen Fehlerreport(alle Tagesprotokolle) erstellt und ausdruckt. Der Funktion *druckeReport* werden beim Aufruf das Array *messung*, die Anzahl der Messarten *messArtAnzahl* und die maximale Toleranz in Prozent *maxToleranz* übergeben.

Hinweis: Die prozentuale Abweichung ergibt sich als Absolutwert der Abweichung des Istwerts vom Sollwert mal 100 geteilt durch den Sollwert.

Die Anzahl der Messarten kann variieren. In diesem Beispiel sind es drei Messarten, es können aber auch n Messarten sein.

Folgende Funktionen stehen Ihnen zur Verfügung:

```
FUNKTION setArray(    laenge: integer): arrayTyp integer
```
Erzeugt ein integer-Array mit der Länge *laenge*.

```
FUNKTION druckeTag(   datum: Datum,
                      tagesProtokoll: arrayTyp integer): void
```
Übergeben werden das Datum des Tages und das Tagesprotokoll für einen Tag. Die Position im Array ist gleich der messArt -Nummer. Druck des Fehlerreports für einen Tag in der Form:

Datum
Messart
Anzahl Abweichungen

```
FUNKTION laenge(array: arrayTyp X): integer
```
Liefert die Länge des angegebenen Arrays.

```
FUNKTION absolut(wert): double
```
Liefert den mathematisch absoluten Betrag des angegebenen Wertes.

Folgende Datenstrukturen stehen Ihnen zur Verfügung:

```
STRUKTUR Messung
  datum:        Datum,
  zeit:         Zeit,
  messArt:      integer,
  istWert:      double,
  sollWert:     double
ENDE STRUKTUR
```

```
FUNKTION druckeReport(    messung: arrayTyp Messung,
                          messArtAnzahl: integer,
                          maxToleranz: double): void
```

Korrekturrand

3. Handlungsschritt (25 Punkte)

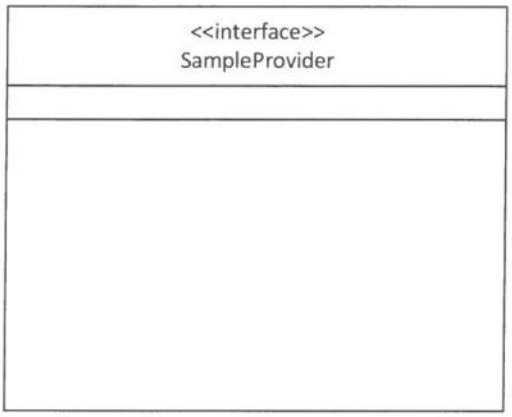
Korrekturrand

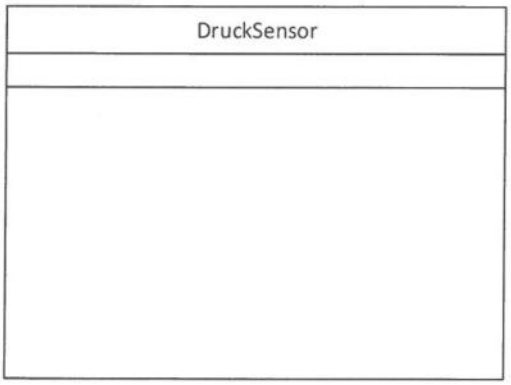
Für die Sensorik der Verpackungsmaschine soll eine objektorientierte Software entwickelt werden. Jeder *DruckSensor* und jeder *TemperaturSensor* nimmt bei einem Messvorgang mehrere Werte auf, die in einem Array für Dezimalzahlen gespeichert werden.

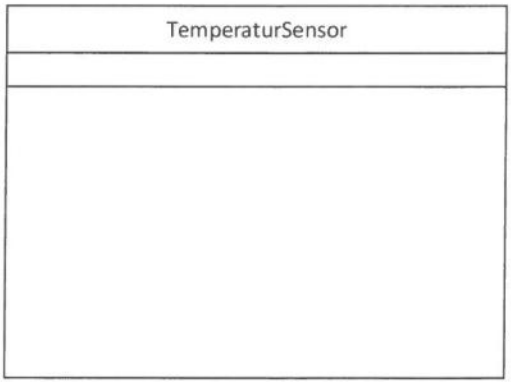
a) Die konkreten Realisierungen *DruckSensor* und *TemperaturSensor* sollen den Basistyp *SampleProvider* implementieren, der die folgenden *öffentlichen* Methodenköpfe vorgibt:

Methode	**Beschreibung**
getType	Hat keinen Übergabeparameter und gibt eine der Messart des Sensors entsprechende Zeichenkette („*Druck*" bzw. „*Temperatur*") zurück.
fetchSample	Weist den einzelnen Speicherplätzen des übergebenen Dezimalzahlenarrays die Messdaten zu und gibt nichts zurück.
sampleSize	Hat keinen Übergabeparameter und gibt die Länge des Messdatenarrays als ganze Zahl zurück

aa) Vervollständigen Sie das UML-Klassendiagramm gemäß Vorgabe. 6 Punkte







ab) Implementieren Sie in Pseudocode die Methode *getType* der Klasse *DruckSensor*. 2 Punkte

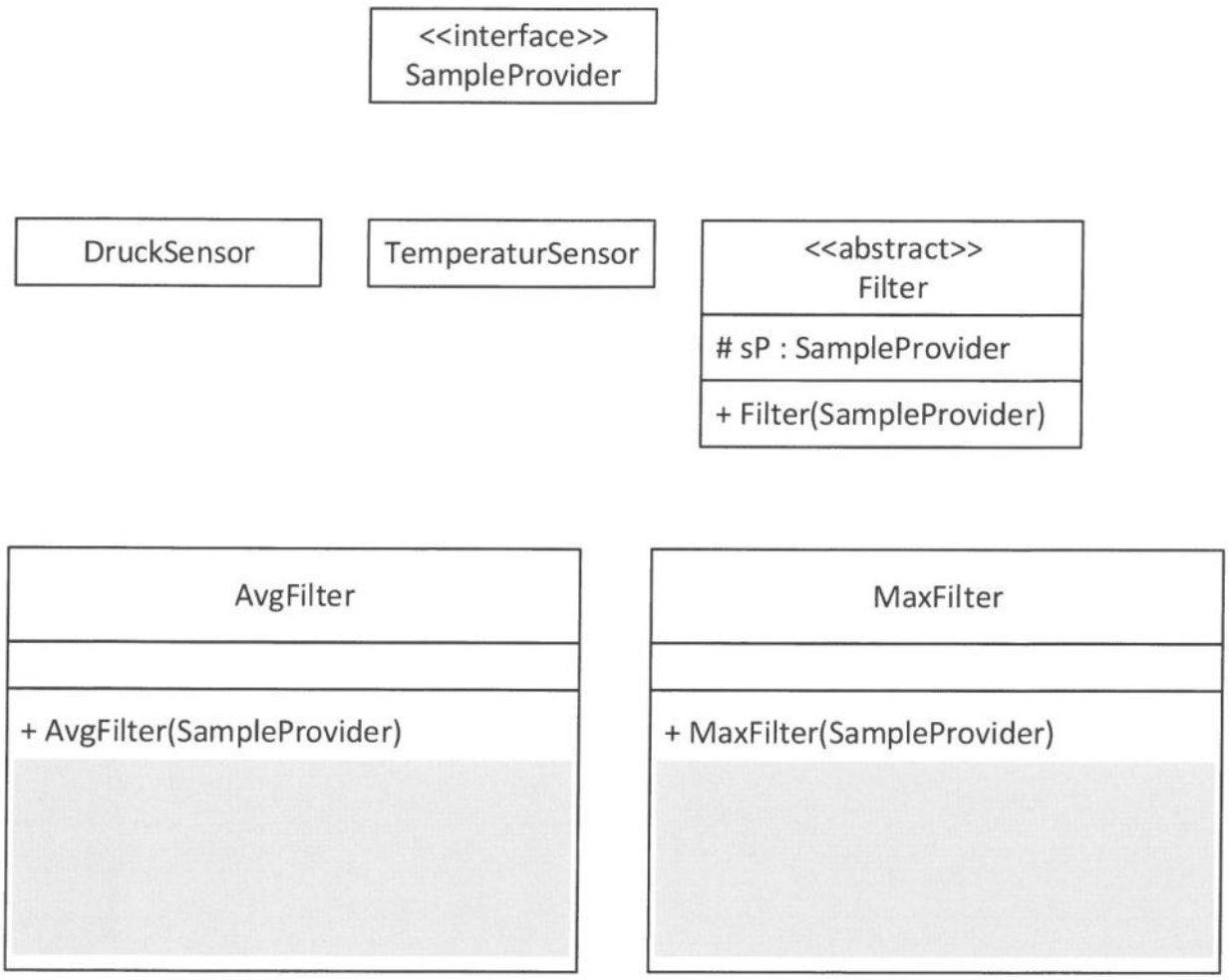
ac) Die Methode *fetchSample* gibt nichts zurück. Erläutern Sie, warum ein Client, der die Methode auf einem *DruckSensor*-Objekt aufruft, trotzdem auf die Messdaten zugreifen kann. 4 Punkte

Korrekturrand

b) Es soll möglich sein, sowohl die kompletten Arrays als auch Auswertungen *(Filter)* des Arrays, z. B. den Durchschnitts- oder Maximalwert direkt vom Sensor zu erhalten.

Um bereits ausgewertete Messvorgänge zu erhalten, werden die Sensorklassen von den Filterklassen (Wrapperklassen) umhüllt. Dazu müssen die konkreten Filterklassen *AvgFilter* und *MaxFilter* die abstrakte Klasse *Filter* erweitern, die sowohl *SampleProvider* implementiert als auch eine Referenz auf ein *SampleProvider*-Objekt hält (Adapter Entwufsmuster).

Tragen Sie in das UML-Klassendiagramm alle Beziehungen ein und ergänzen Sie die Methoden in den grau markierten Bereichen. 6 Punkte

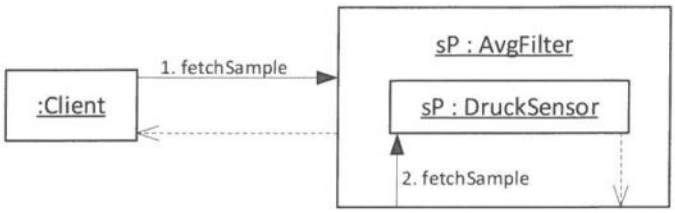


Fortsetzung 3. Handlungsschritt →

Fortsetzung 3. Handlungsschritt

Korrekturrand

c) Ein Client benötigt den Durchschnittswert einer Drucksensormessung. Dazu wird ein *DruckSensor*-Objekt erzeugt und mit einem *AvgFilter*-Objekt umhüllt. Bei Aufruf der Methode *sP.fetchSample* wird das übergebene Array mit Messdaten befüllt. Dann wird der Durchschnittwert berechnet und unter Arrayindex *[0]* abgelegt.



ca) Implementieren Sie in Pseudocode die clientseitige Erzeugung eines *DruckSensor*-Objekts und dessen Umhüllung mit einem *AvgFilter*-Objekt. 3 Punkte

cb) Implementieren Sie in Pseudocode die Konstruktoren der Klasse *Filter* und *AvgFilter*. 4 Punkte

4. Handlungsschritt (25 Punkte)

Die rapidPack GmbH stellt Maschinen aus verschiedenen Teilen her und will für ein neues Teilebestellsystem eine Datenbank entwickeln.

Ein grober, noch unvollständiger Entwurf der Datenbank liegt bereits vor.

a) Vervollständigen Sie das Datenmodell.
- Ergänzen Sie in den Tabellen *Teil, Bestellung, Lieferer* und *BestellPosition* die erforderlichen Attribute.
- Ergänzen Sie die leere Tabelle, um die folgende Anforderung zu erfüllen: Ein Teil kann von verschiedenen Lieferern zu unterschiedlichen Preisen bezogen werden. Für jede Bestellung kommen die Teile stets von einem Lieferanten. Vergeben Sie einen sinnvollen Tabellennamen und tragen Sie die erforderlichen Attribute ein.
- Kennzeichnen Sie die Primärschlüssel mit (PK) und die Fremdschlüssel mit (FK).
- Zeichnen Sie die Beziehungen zwischen den Tabellen mit den Kardinalitäten ein.

20 Punkte

TeilGruppe
Gruppe_ID
GruppeBezeichnung

BestellPosition
Bestell_ID

Bestellung
Bestell_ID
BestellDatum

Teil
Teil_ID
TeilBezeichnung
Bestand

Lieferer
LiefererName
LiefererAdresse

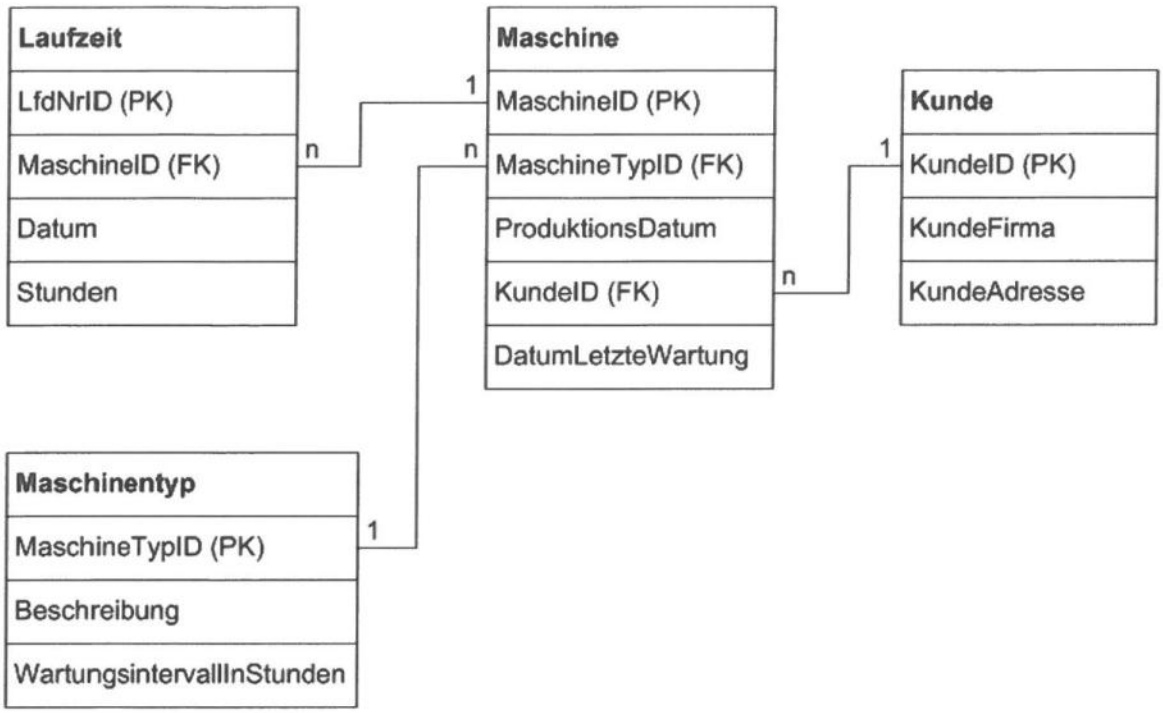
b) Erläutern Sie, warum die Tabelle *Lieferer* nicht der 3. Normalform entspricht. 5 Punkte

Korrekturrand

5. Handlungsschritt (25 Punkte)

Korrekturrand

Die rapidPack GmbH möchte ihre Maschinenwartung optimieren. Dazu hat sie bereits das folgende Wartungsmodul erstellt.



Zur Abfrage und Pflege sollen nachfolgende SQL-Anweisungen erstellt werden.

a) Erstellen Sie eine Liste aller Maschinentypen mit Anzahl der Maschinen des Typs absteigend sortiert nach AnzahlMaschinen.

5 Punkte

Beispielliste

MaschineTypID	Beschreibung	WartungsintervallInStunden	AnzahlMaschinen
1	Füll	900	3
2	Verpackung	1.800	1
3	Etikettierung	1.000	0

b) Erstellen Sie eine Liste aller Kunden und ihrer hinterlegten Maschinen, deren Laufzeit nach der letzten Wartung das Wartungsintervall in den nächsten 100 Stunden überschreiten werden. 8 Punkte

Korrekturrand

Beispielliste

KundeID	KundeFirma	KundeAdresse	MaschineID	Laufzeit
1	LikeLimo	Musteradresse	1	2.500

c) Erstellen Sie eine Liste aller Maschinentypen, der zugehörigen Kunden und Laufzeit der jeweiligen Maschine seit der letzten Wartung. 8 Punkte

Beispielliste

MaschineTypID	Beschreibung	KundeFirma	Laufzeit
1	Füll	LikeLimo	2.500
1	Füll	LikeLimo	NULL
1	Füll	Musterfirma	NULL
2	Verpackung	LikeLimo	NULL
3	Etikettierung	NULL	NULL

Fortsetzung 5. Handlungsschritt →

Fortsetzung 5. Handlungsschritt

Korrekturrand

d) Reduzieren Sie für alle Verpackungsmaschinen das Wartungsintervall um 10 %. 4 Punkte

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.

Abschlussprüfung Sommer 2019

IHK

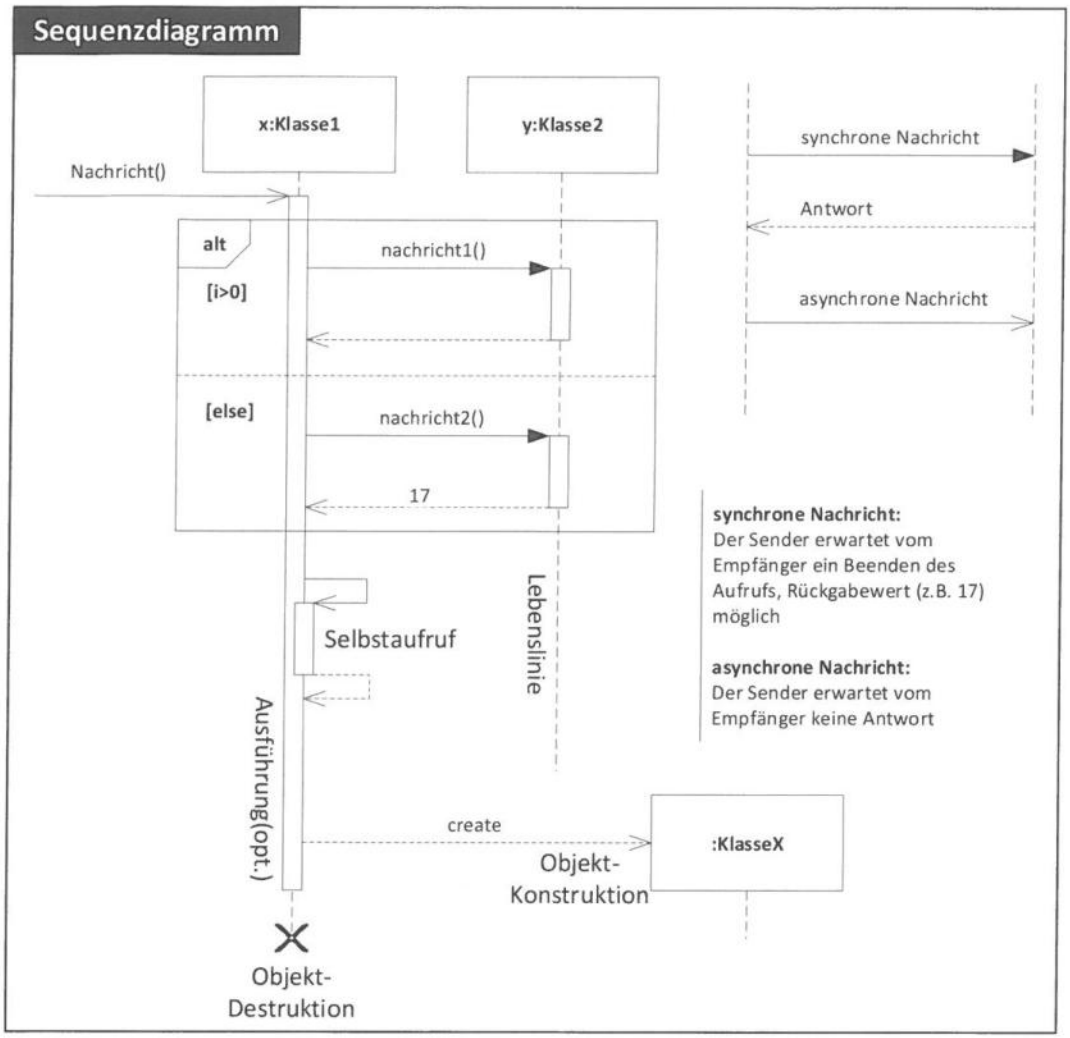
Belegsatz

Fachinformatiker Anwendungsentwicklung
Fachinformatikerin Anwendungsentwicklung
1196

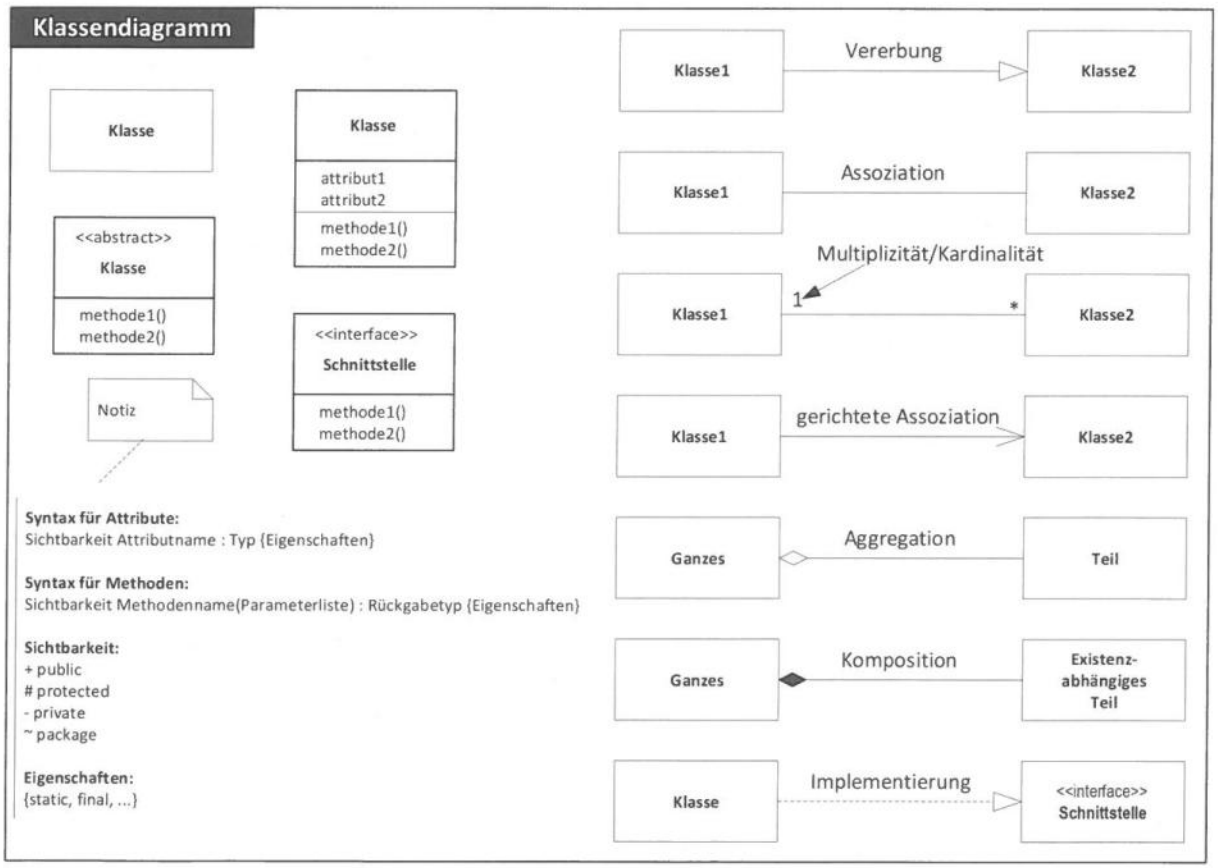
1 Ganzheitliche Aufgabe I
Fachqualifikationen

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2019 – Alle Rechte vorbehalten!

UML-Sequenzdiagramm



UML-Klassendiagramm



SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... **(SELECT** ... **FROM** ... **WHERE** ...**) AS** xyz **FROM** ... **WHERE** ...	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: *xyz)* ausgegeben.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery **)** **WHERE NOT EXISTS (** subquery **)**	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
GROUP BY Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [,Spaltenname2, ...] **ASC | DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
Syntax	**Beschreibung**
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname **VALUES** (Wert für Spalte 1 [, Wert für Spalte 2, ...])	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind

Fortsetzung →

SQL-Syntax (Auszug) – Fortsetzung

oder **SELECT ... FROM ... WHERE**	
Aggregatfunktionen	
AVG(Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname | Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname | Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname | Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(time,[DatumZeit**])**	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert)	Wandelt einen Wert in ein Datum um
DAY(Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR(Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: **DAY, MONTH, YEAR**	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Textattributen auf Gleichheit, Verwendung von Platzhaltern möglich.
NOT	Logische Negation
OR	Logisches ODER
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2018-03-29