

1. Handlungsschritt (25 Punkte)

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

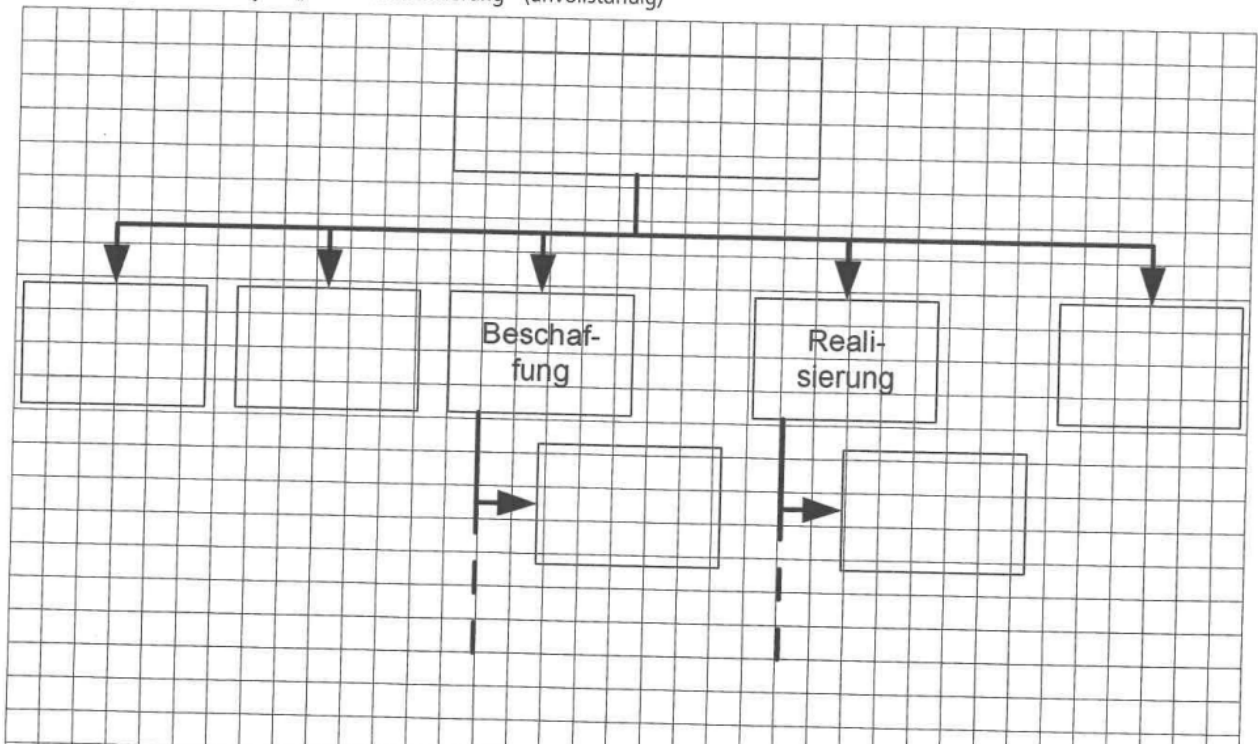
	Vorgang*	Beschreibung
1	Abnahme und Betrieb	System übergeben und betreiben
2	Angebotsvergleich	Angebote vergleichen
3	Bedarfsermittlung	Software-Bedarf ermitteln
4	Beschaffung	Software beschaffen
5	Bestellung	Software bestellen
6	Installation	Software installieren
7	Planung	IT-Systemanpassung planen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

* Sortierung in alphabetischer Reihenfolge

Ergänzen Sie den folgenden Projektstrukturplan.

5 Punkte

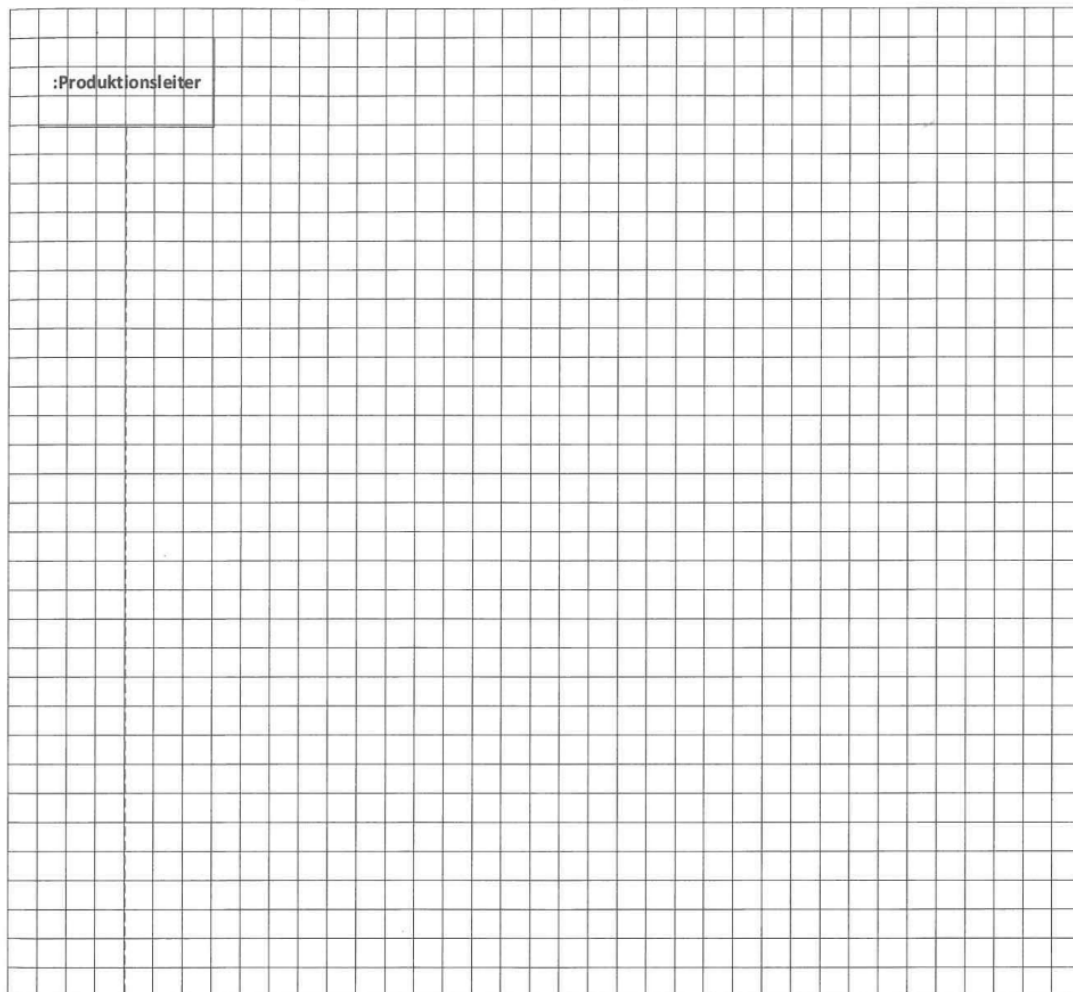
Strukturplan zum Projekt „Maschinensteuerung“ (unvollständig)



- 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)

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 ausdrückt. 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

Korrekt

3. Handlungsschritt (25 Punkte)

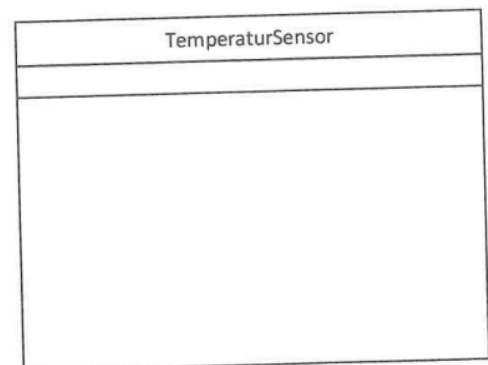
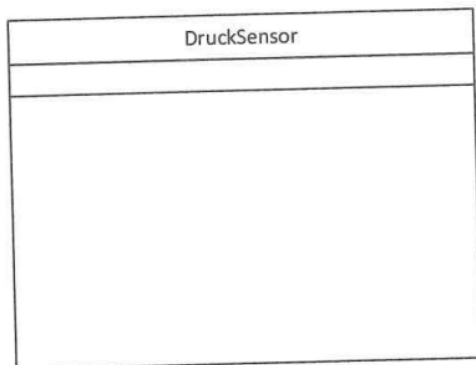
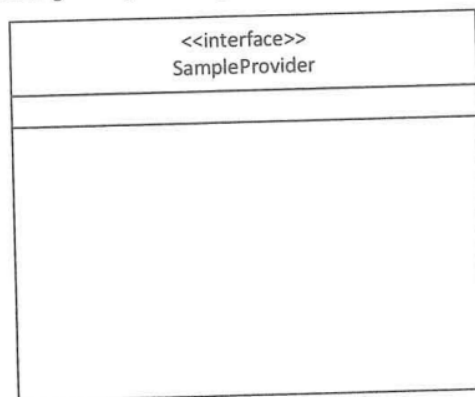
Für die Sensorik der Verpackungsmaschine soll eine objektorientierte Software entwickelt werden. Jeder *DruckSensor* und jeder *TemperaturSensor* nimmt bei einem Messvorgang mehrere Werte auf, die in einem Array für Dezimalzahlen gespeichert werden.

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

Methode	Beschreibung
<i>getType</i>	Hat keinen Übergabeparameter und gibt eine der Messart des Sensors entsprechende Zeichenkette („ <i>Druck</i> “ bzw. „ <i>Temperatur</i> “) zurück.
<i>fetchSample</i>	Weist den einzelnen Speicherplätzen des übergebenen Dezimalzahlenarrays die Messdaten zu und gibt nichts zurück.
<i>sampleSize</i>	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

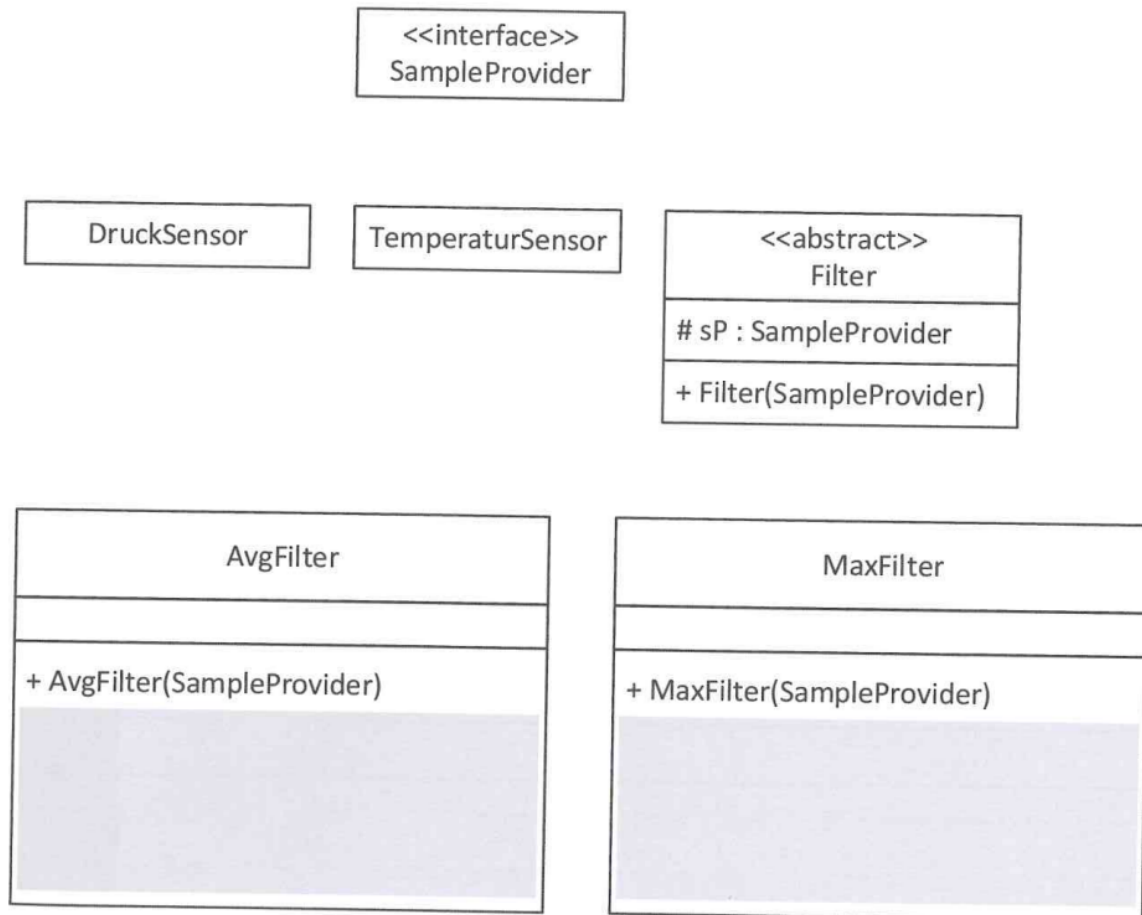
Korrektu

- 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 Entwurfsmuster).

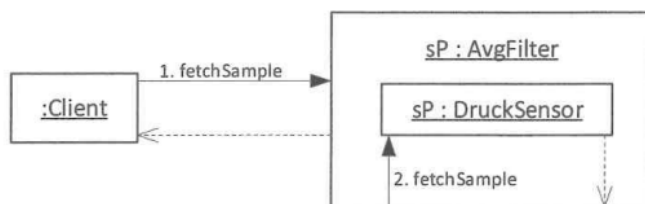
Tragen Sie in das UML-Klassendiagramm alle Beziehungen ein und ergänzen Sie die Methoden in den grau markierten Bereichen.

6 Punkte



Fortsetzung 3. Handlungsschritt

- 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 Durchschnittswert 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 Lieferanten 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

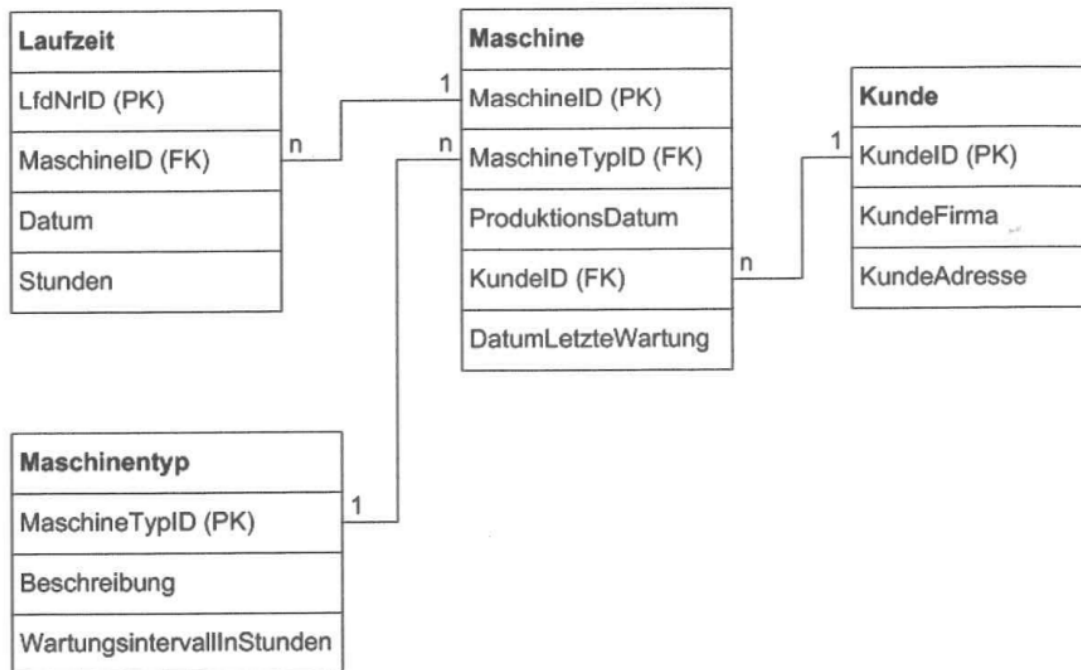
b) Erläutern Sie, warum die Tabelle *Lieferer* nicht der 3. Normalform entspricht.

5 Punkte

Korrekturrand

5. Handlungsschritt (25 Punkte)

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

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

- c) 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

Beispielliste

KundeID	KundeFirma	KundeAdresse	MaschineID	Laufzeit
1	LikeLimo	Musteradresse	1	2.500

Korr

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

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

4 Punkte
