Abschlussprüfung Sommer 2022
der Berufsschulen (gewerbl. Bereich)
Ministerium für Kultus, Jugend und Sport
Baden-Württemberg

Abschlussprüfung Sommer 2022
der Industrie- und Handelskammern
(schriftlicher Teil)
Baden-Württemberg

Fachinformatiker/-in Fachrichtung Anwendungsentwicklung

FA 234

Planen eines Softwareproduktes

Bearbeitungszeit:
90 Minuten

Verlangt:	**Alle Aufgaben**

Hilfsmittel: Nicht programmierter Taschenrechner

Bewertung: Die Bewertung der einzelnen Aufgaben ist durch Punkte näher vorgegeben.

Zu beachten: Die Prüfungsunterlagen sind vor Arbeitsbeginn auf Vollständigkeit zu überprüfen.

Der Aufgabensatz zu Planen eines Softwareproduktes besteht aus:

- den Aufgaben 1 bis 3

Bei Unstimmigkeiten ist sofort die Aufsicht zu informieren.

Klare und übersichtliche Darstellung der Rechengänge mit Formeln und Einheiten wird entscheidend mitbewertet.

Projekt: IAV-DriveApp (Import, Analyse and View of Drive-Data)

Projektbeschreibung:

Das Unternehmen Packmeister GmbH ist ein Hersteller von hochentwickelten automatisierten Verpackungsmaschinen. Die Packmeister GmbH hat sich dabei auf die Verpackung von Hygieneartikeln wie zum Beispiel Windeln, Papiertaschentüchern oder Toilettenpapier spezialisiert.
Die Artikel werden von den Maschinen mit sehr hoher Geschwindigkeit in Einzelhandelsverpackungen verpackt.
In den Maschinen kommen unterschiedliche elektrische Antriebsmotoren zum Einsatz. Beispielsweise für die automatische Zuführung der einzelnen Artikel zur Maschine.
Um möglichst viele Artikel pro Stunde verarbeiten zu können, müssen die Motoren bei hoher Geschwindigkeit sehr präzise arbeiten.
Dazu gilt es zum einen, die Antriebssteuerungen vor der Auslieferung der Maschine an den Kunden zu optimieren (Aufgabe 1) und zum anderen, die Antriebe während des Betriebs laufend zu überwachen (Aufgabe 2).
Für beide Zwecke werden Betriebswerte wie die aktuelle Antriebsposition, die Stromaufnahme und die Geschwindigkeit durch die Antriebssteuerungen in Echtzeit aufgezeichnet (Tracing).

Aufgabe 1 **50**

Ausgangssituation
Die durch das Tracing gewonnenen Betriebswerte sollen durch eine neu zu entwickelnde Visualisierungs- und Analyse-Software gelesen und als Kennlinien dargestellt werden. Die Visualisierung ist hilfreich, um Optimierungsmöglichkeiten zu erkennen.
Bis die Optimierung abgeschlossen ist, müssen pro Antrieb ca. 4-6 Aufzeichnungen durchgeführt und mit Hilfe der Kennlinienvisualisierung analysiert werden. Jede Aufzeichnung dauert maximal 3 Sekunden. Dabei werden in sehr kurzen Abständen (z. B. 50 ms) die Position, die Geschwindigkeit und die Stromaufnahme gemessen und in eine Datei geschrieben.
Mit den daraus gewonnenen Messwerten werden die Antriebs-Steuerungen so eingestellt, dass die Packmaschinen die Artikel mit höchstmöglicher Geschwindigkeit präzise in ihre Verpackungen bringen.

Alle Trace-Dateien werden auf dem Datei-Server gespeichert. Je nach Kundenwunsch werden Antriebssteuerungen von unterschiedlichen Steuerungsherstellern eingesetzt. Das Dateiformat ist somit uneinheitlich.

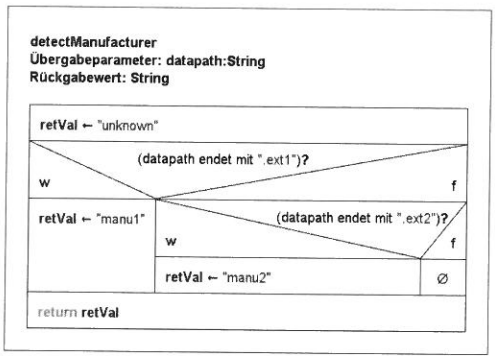
```
Datum/Uhrzeit: 13.06.2021 11:43
Antriebsbezeichnung: Zuführung
Zeitpunkt in ms,Position in mm,Geschwindigkeit in m/s,Stromaufnahme in Ampere
0000.0,000.00,0.00,1.50
0100.0,000.00,0.02,1.71
0200.0,005.00,0.10,3.60
0300.0,020.10,0.20,2.07
0400.0,049.81,0.22,0.82
```

Abbildung 1: Beispiel-Trace-Datei Hersteller 1

```
<Date>13.06.2021</Date>
<Time>11:43</Time>
<Operator>Max Muster</Operator>
<Drivename>R8700W18</Drivename>
<Stepsize unit="ms">100.0</<Stepsize>
<DataPoints>
    <Point velo="0.00" current="1.500">000.0</Point>
    <Point velo="0.02" current="1.710">000.0</Point>
    <Point velo="0.10" current="3.600">0.005</Point>
    <Point velo="0.20" current="2.070">0.020</Point>
    <Point velo="0.20" current="0.820">0.049</Point>
       ...
</DataPoints>
```

Abbildung 2: Beispiel-Trace-Datei Hersteller 2

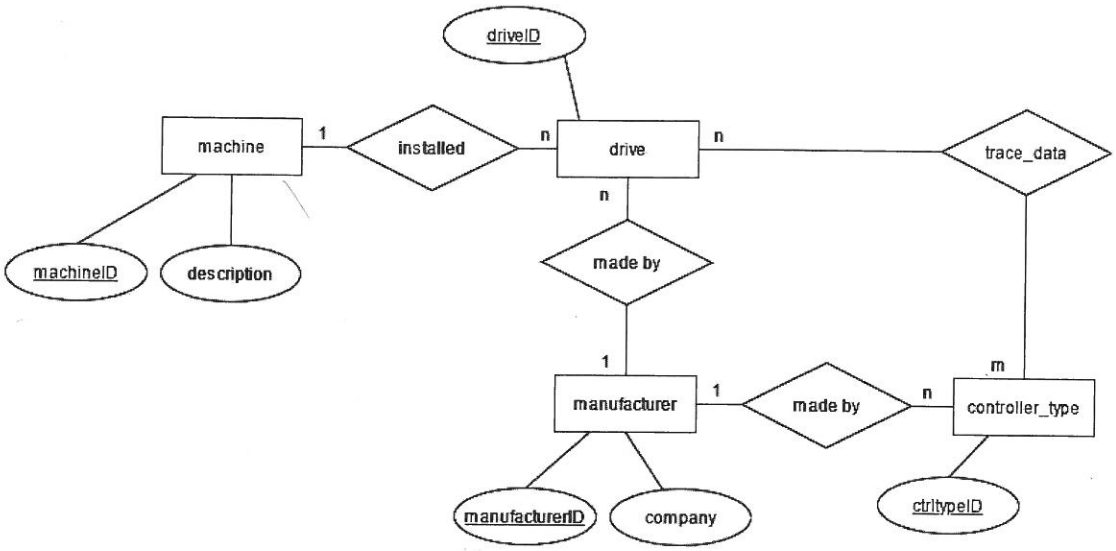
		Aufgaben Punkte
1.1	Nennen Sie die Datenformate, welche die beiden Steuerungshersteller jeweils für ihre Trace-Daten nutzen und wodurch die jeweiligen Datenformate zu erkennen sind.	4
1.2	Sie sind als Mitglied des Projektteams **IAV-DriveApp** mit der Entwicklung des Trace-Daten-Imports betraut worden. Sie sollen deshalb ein bestehendes UML-Klassendiagramm (Anlage 1) um die Importfunktionalität erweitern und darauf achten, dass das Design für weitere Import-Algorithmen von weiteren Steuerungsherstellern möglichst flexibel bleibt.	
1.2.1	Begründen Sie, welches Designpattern (Entwurfsmuster) in der Anlage 3 durch die Klasse **Import** umgesetzt wird!	5
1.2.2	Ergänzen Sie das UML-Klassendiagramm (Anlage 1) mit der Importfunktionalität für zwei verschiedene Datenformate (Manufacturer1 und Manufacturer2) mit einheitlicher Schnittstelle namens **ImportService**, welche in der Anlage 2 beschrieben wird. Für jedes Datenformat ist eine eigene Klasse vorzusehen, welche die Schnittstelle **ImportService** implementieren muss. Stellen Sie auch die notwendigen Beziehungen dieser Schnittstelle her. Benötigte Parameter und deren Datentypen sowie die Rückgabewerte mit entsprechenden Datentypen legen Sie selbst sinnvoll fest. Achten Sie dabei auf die bereits vorhandenen Klassen des Diagramms.	20
1.3	Implementieren Sie die Methode **detectManufacturer** der Klasse **Import** mit der an Ihrer Schule unterrichteten Programmiersprache nach Vorlage des abgebildeten Struktogramms.	7



1.4	Implementieren Sie die Methode **ImportData** der Klasse **ImportController** mit der an Ihrer Schule unterrichteten Programmiersprache (Anlage 1 und Anlage 2). Beachten Sie unter anderem, dass das Attribut **ImportService** bereits durch einen vorherigen Aufruf der Methode **Read** initialisiert wurde.	6
1.5	Die nächste Version der Software soll einen weiteren Hersteller von Antriebssteuerungen unterstützen, damit dessen Trace-Daten ebenfalls verarbeitet werden können.	
1.5.1	Erklären Sie, welchen Vorteil die Verwendung der Klasse **Import** in diesem Fall mit sich bringt.	3
1.5.2	Erklären Sie, welche Anpassungen im Quellcode der Klasse **Import** vorzunehmen sind, damit weitere Hersteller von der Software unterstützt werden können.	5

Aufgabe 2 23

Ausgangssituation
Um neue Antriebe in Kombination mit Steuerungstypen unterschiedlicher Hersteller im Dauerbetrieb zu testen, soll eine Datenbank erstellt werden, die die Sensordaten über einen längeren Zeitraum speichert. Die Softwareabteilung hat ein ERD der zu erstellenden Datenbank entworfen:



2.1 Geben Sie das relationale Modell in der Relationenschreibweise an und kennzeichnen Sie die Schlüsselattribute in eindeutiger Weise! 7
Die Attribute der Verbindungsentität entnehmen Sie den Dateien aus den Abbildungen 1 und 2 der Aufgabe 1.

2.2 Die Verbindungsentität muss mittels SQL erstellt werden. Verwenden Sie als Vorlage Abbildung 1 und 2 mit den Trace-Daten der Hersteller 1 bzw. Hersteller 2 und geben Sie den entsprechenden SQL-Befehl an. 5

Hinweis:
Wenn Sie die Attribute unter Aufgabe 2.1 nicht herausfinden konnten, beschränken Sie sich auf die Schlüsselattribute.

2.3 Erstellen Sie eine SQL-Abfrage, die ermittelt, wie viele Motoren der Motorenhersteller „OverDrive“ mit der „manufacturerID“ ‘12‘ in dem Maschinentyp „Folienking“ mit der „machineID“ ‘3‘ verbaut sind. 6
Anmerkung: Lassen Sie zusätzlich zur Anzahl der verbauten Motoren von der Tabelle „machine“ die Attribute „description“ und „machineID“ sowie von der Tabelle „manufacturer“ die „company“ ausgeben.

2.4 Die in der Datenbank gespeicherten Sensordaten sollen über einen REST-Service allen Niederlassungen zur Verfügung gestellt werden.

2.4.1 Benennen Sie das Protokoll, das die REST-Architektur zur Kommunikation zwischen Client und Server benutzt. 2

2.4.2 Benennen Sie das Datenformat des Datentransfers, das bei REST meist verwendet wird. 3

Aufgaben Punkte

Aufgabe 3 17

Ausgangssituation
Die Firma Packmeister GmbH möchte ihren Kunden, im Rahmen von Industrie 4.0, den Service des „predictive Maintenance" anbieten.

3.1 Nennen Sie jeweils zwei Vorteile für die Packmeister GmbH und deren Kunden durch die Einführung des Service des „predictive Maintenance". 4

3.2 Sie sind im Projektteam, das einen Geschäftsprozess zu „predictive Maintenance" einführen soll. Der Ablauf des Geschäftsprozesses wurde bereits vorgegeben. 13
Füllen Sie die Vorlage des eEPK (Anlage 4), anhand der folgenden Beschreibung aus.

Die Service-Abteilung überwacht automatisiert die Sensordaten. Sobald ein Schwellenwert überschritten wird, kontrolliert ein Mitarbeiter die entsprechenden Daten der Kunden-Datenbank. Die relevanten Daten sind in der Tabelle „tb_Sensordaten" gespeichert. Anhand des Datenblattes des betroffenen Motors prüft ein Mitarbeiter, ob ein Wartungsfall vorliegt.

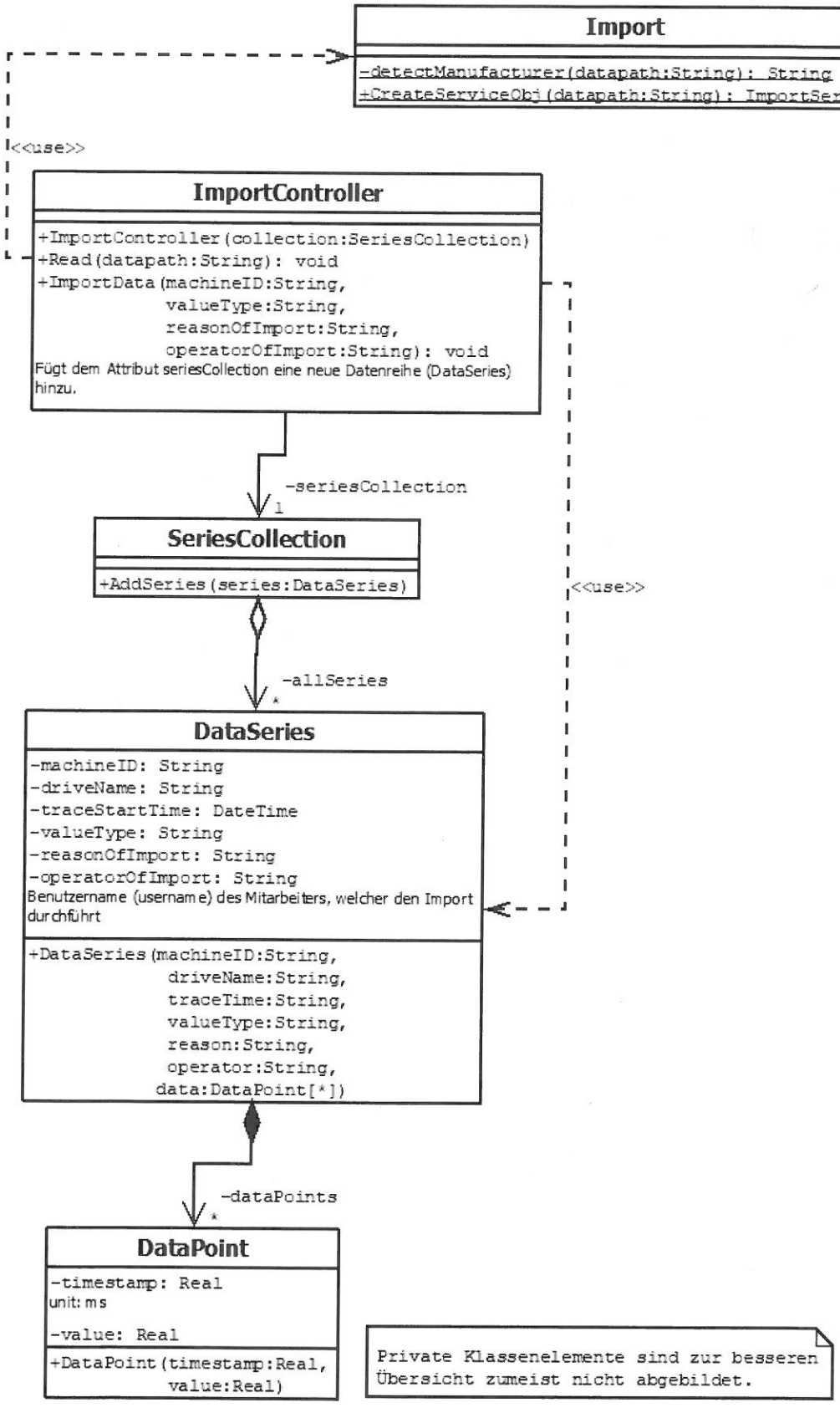
Liegt kein Wartungsfall vor, gilt der Prozess als beendet und der Kunde wird darüber in Kenntnis gesetzt.
Liegt ein Wartungsfall vor, muss die Dringlichkeit eingeordnet werden. Drei Stufen werden unterschieden: Hohe, mittlere oder niedrige Dringlichkeit. Ein Mitarbeiter nimmt diese Einordnung anhand einer Wartungsvorlage und des Datenblatts der jeweiligen Maschine unter Abgleich der Daten aus der Tabelle „tb_Sensordaten" vor.
Abhängig von der Dringlichkeit entscheidet der Mitarbeiter, wann eine Wartung von welchem Techniker durchgeführt werden soll und trägt dies in den Wartungskalender ein. Die benötigten Informationen bezieht er aus einer Tabelle „tb_Techniker" in der die Qualifikationen der einzelnen Techniker enthalten sind und dem Datenblatt der Maschine. Gleichzeitig bestellt dieser Mitarbeiter Ersatzteile, falls welche benötigt werden.

Wenn Terminierung und Bestellung durchgeführt wurden erfolgt eine Rückmeldung an den Kunden und - falls nötig - an den entsprechenden Techniker. Sobald die beteiligten Personen über den Wartungstermin informiert wurden, ist dieser Prozess abgeschlossen.

Bitte geben Sie dieses Blatt mit Ihren Lösungen ab.

Name, Vorname: _____ **Klasse:** _____

Anlage 1 UML-Klassendiagramm



Anlage 2 Klassen (siehe auch Anlage 1)

ImportController

importService	Privates Attribut
SeriesCollection	Liste von Objekten der Klasse DataSeries
Read	Initialisiert unter Verwendung der Klasse Import das Attribut importService und liest durch Aufruf der Methode Read der Schnittstelle ImportService den Dateiinhalt ein.
ImportData	Fügt dem Attribut seriesCollection eine neue Datenreihe (DataSeries) hinzu. Das User-Interface nutzt die Methode, um unter Angabe zusätzlicher Metadaten (Maschinen-Ident-Nummer, Grund des Datenimports, ...) eine neue Datenreihe zu importieren.

Import

detectManufacturer	Der Parameter datapath enthält den vollständigen Dateinamen der Trace-Daten-Datei (inkl. Pfad und Extension). Abhängig von der Dateierweiterung gibt die Methode eine Formatkennung zurück. Mögliche Rückgabewerte sind {unknown, manu1, manu2}. Hinweis: Die Unterscheidung des Datenformats nur anhand der Extension ist eine Vereinfachung im Rahmen der Prüfung.
CreateServiceObj	Instanziiert Objekte von Klassen, die das Interface ImportService implementieren, abhängig von der Dateierweiterung des übergebenen „datapath“.

Schnittstelle

ImportService

GetDataPath	Lesen des Dateinamens (vollständig inkl. Pfad).	
Read	Liest alle Zeilen der Datei in den Hauptspeicher. Rückgabe und Parameter nicht erforderlich.	
GetTraceStartTime	Gibt den Zeitpunkt und Datum der Aufzeichnung zurück.	
GetDriveName	Liefert den Namen des Antriebs (z. B. Bestückung oder Pressen)	
GetAvailableValueTypes	Die Aufzeichnungen enthalten gewöhnlich Datenreihen für die Position, die Geschwindigkeit und die Stromaufnahme. Alles in Abhängigkeit von der Zeit. Manche Hersteller zeichnen zusätzlich noch die Motorspannung oder das Drehmoment auf. Welche Werte die gewählte Datei tatsächlich enthält, wird durch die Methode GetAvailableValueTypes als Array zurückgegeben. Z. B. {Position, Velocity, Current} oder {Position, Velocity, Torque}	
GetData	Gibt ein Array mit den aufgezeichneten Datenpunkten (Klasse DataPoint) zu genau einem übergebenen ValueType zurück. Jeder Datenpunkt besteht aus einem Zeitwert (timestamp) und einem Messwert (value). Siehe Klassendiagramm.	

Anlage 3 Designpatterns

Singleton Pattern
Eine Singletonklasse ist so implementiert, dass es maximal nur eine Instanz der Klasse geben kann.

Decorator Pattern
Eine Instanz eines Decorators fügt zu einer vorhandenen Instanz neue Funktionalitäten hinzu.

Factory Pattern
Eine Factory-Klasse bietet die Funktionalität, Instanzen von anderen Klassen zu generieren. Dabei muss der Klassenname nicht angegeben werden.

Observer Pattern
Bei Änderungen an einer Instanz werden mehrere Beobachter Instanzen von diesen Änderungen benachrichtigt. Die Beobachter müssen sich dazu am zu beobachtenden Objekt „registrieren".

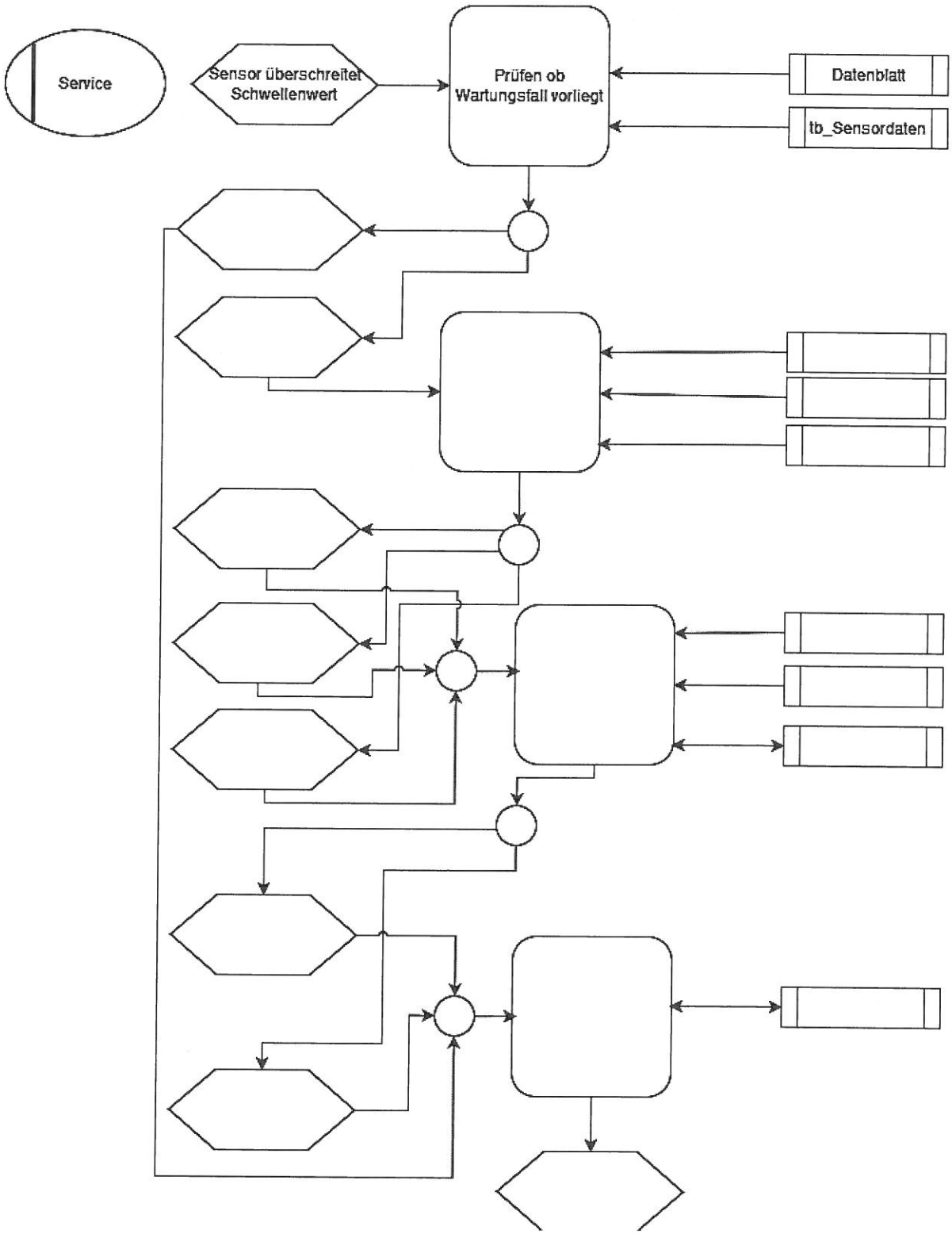
Glossar

Betriebswerte oder Prozesswerte	Daten, die während eines technischen Vorgangs ermittelt werden. Hier in diesem Projekt handelt es sich um die Stromaufnahme, die Geschwindigkeit und die Position von beweglichen Maschinenelementen, welche durch elektrische Motoren angetrieben werden.
Tracing	Das Aufzeichnen oder Mitschreiben von Daten, die während eines Vorgangs laufend anfallen. Hier: Das Schreiben von Prozessdaten in eine Datei während der Bewegung.
DataSeries	Eine Datenreihe. Hier speziell eine Zeitreihe. Also Daten, die in zeitlichem Bezug stehen. Hier z. B. die Stromaufnahme eines Antriebs in Abhängigkeit von der Zeit.
DataPoint	Datenpunkt. Eine Momentaufnahme eines Werts oder mehrerer Werte.
Timestamp	Zeitstempel Zu Messwerten oder Ereignissen wird meist der Zeitpunkt des Messens bzw. der Zeitpunkt des Eintretens des Ereignisses erfasst.
SeriesCollection	Eine Liste mehrerer Datenreihen.
Operator	Der Bediener einer Maschine oder einer Software. Er überwacht und steuert.
Reason Reason of import	Grund Der Grund oder die Absicht, warum der Datenimport durchgeführt wurde.
Drive	Antrieb oder auch Motor
Controller	Steuerung

Anlage 4

Bitte geben Sie dieses Blatt mit Ihren Lösungen ab.

Name, Vorname: _____ Klasse: _____



Fachinformatiker/-in Fachrichtung Anwendungsentwicklung FA 234
Planen eines Softwareproduktes

Lösungsvorschläge: **Lösungsvorschläge sind in der Regel Vorschläge der einreichenden Schulen; sie sind im Wortlaut nicht bindend. Anderslautende, aber zutreffende Antworten sind ebenfalls als richtig zu werten.**

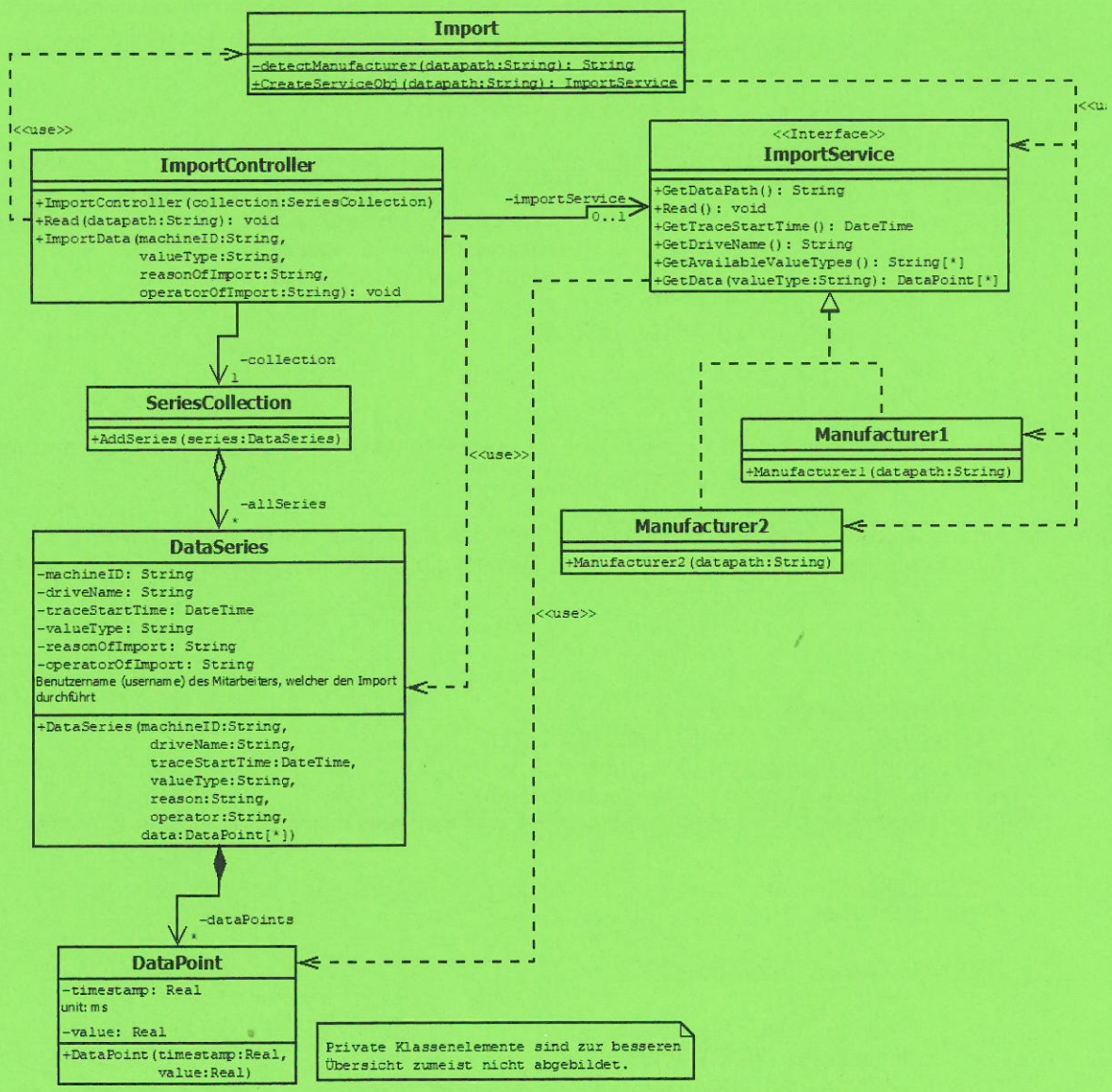
Nur für die Hand des Prüfers! Punkte

Aufgabe 1 — 50

1.1 Hersteller 1 nutzt ein CSV-Format (Daten durch Kommata getrennt) und Hersteller 2 nutzt ein XML-Format (Elemente werden durch Tags ausgezeichnet). — 4

1.2.1 Factory-Pattern, weil eine in Instanz einer Klasse generiert wird ohne, dass der Klassenname genannt werden muss. — 5

1.2.2 — 20



Datentypbezeichnungen nach UML 2.5

1.3 Lösung in C#: 7

```
private static string detectManufacturer(string datapath)
{
    string retVal = „unknown“;

    if (datapath.EndsWith(„.ext1“, true, null))
        retVal = „manu1“;
    else if (datapath.EndsWith(„.ext2“, true, null))
        retVal = „manu2“;

    return retVal;
}
```

Auf den 2. und 3. Parameter beim Aufruf der Methode EndsWith kann verzichtet werden. Andere Lösungen wie z.B. die Adressierung der letzten vier Zeichen über den Index sind ebenso richtig.
`z.B. datapath[datapath.Length-4] == 'e'`

1.4 Lösung in C#: 6

```
public void ImportData(string machineID, string valueType, string reasonOfImport,
                       string operatorOfImport)
{
    DataSeries ds = new DataSeries(machineID, importService.GetDriveName(),
                                   importService.GetTraceStartTime(), valueType,
                                   reasonOfImport, operatorOfImport,
                                   importService.GetData(valueType);

    collection.AddSeries(ds);
}
```

1.5.1 Der bereits vorhandene Programmcode muss lediglich ergänzt werden um die neu hinzugekommenen Dateiformate zu erkennen. 3

1.5.2 Um das neue Format zu erkennen ist die Methode detectManufacturer zu erweitern. 5
Zum Erzeugen von Instanzen des neuen Typs ist die Methode CreateServiceObj zu erweitern.

Aufgabe 2 **23**

2.1 machine(machineID, description, …) 7
drive(driveID, …, #machineID, #manufacturerID)
manufacturer(manufacturerID, company, …)
controller_type(ctrltypeID, …, #manufacturerID)
trace_data(#driveID, #ctrltypeID, timestamp, position, geschwindigkeit, stromaufnahme)

Primärschlüssel
#Fremdschlüssel

2.2 5

```
CREATE TABLE `trace_data` (
    `driveID` INT(11),
    `ctrltypeID` INT(11),
    `timestamp` TIMESTAMP,
    `position (mm)` DOUBLE,
    `geschwindigkeit (m/s)` DOUBLE,
    `stromaufnahme (Ampere)` DOUBLE,
    PRIMARY KEY (`driveID`, `ctrltypeID`);
```

2.3 **SELECT** ma.machineID, ma.description, m.company, **COUNT**(d.driveID) **AS** Anzahl_verbauter_Motoren 6
FROM manufacturer m **INNER JOIN** drive d **ON** m.manufacturerID = d.manufacturerID **INNER JOIN** machine ma **ON** ma.machineID = d.machineID
WHERE ma.machineID = 3 **AND** m.manufacturerID = 12;

2.4.1 HTTP(S) 2

2.4.2 JSON (Java Script Object Notation) 3

Aufgabe 3 **17**

3.1 Packmeister GmbH: 4
- Kundenbindung
- Kundenzufriedenheit

Kunde:
- Kostenersparnis (durch Verringerung der Ausfallzeiten)
- Höhere Planbarkeit der Produktionskapazität (durch weniger kurzfristige Unterbrechungen)

3.2 13