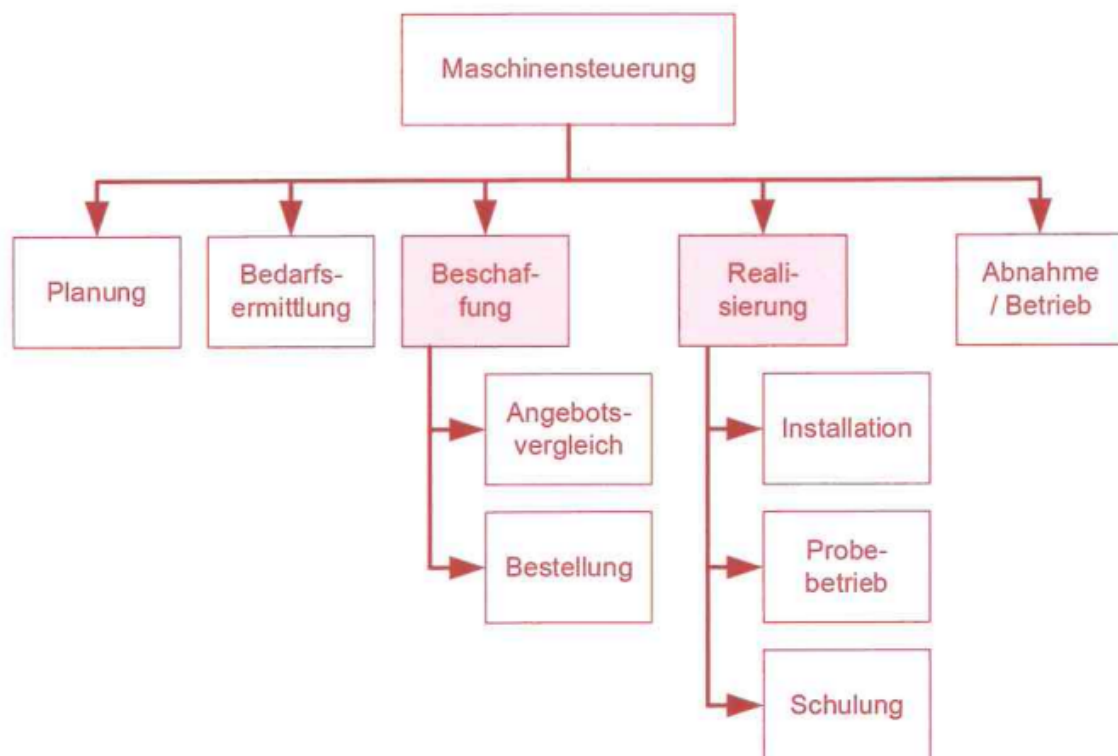


a) 5 Punkte

1 Punkt für Projektname Maschinensteuerung auf oberster Ebene

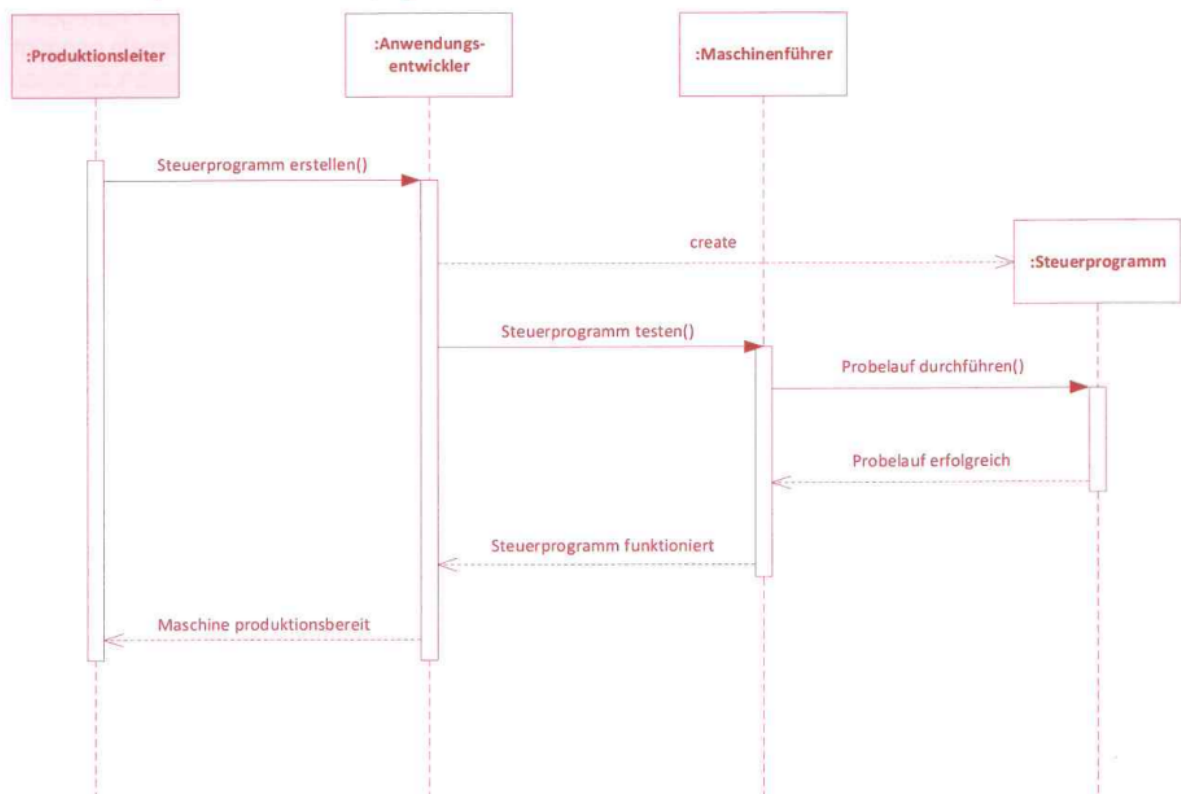
je 0,5 Punkte für sinnvoll angeordnete Arbeitspakete

grau hinterlegte Arbeitspakete waren vorgegeben



Andere Lösungen sind möglich.

- b) 20 Punkte
- 6 Punkte, je 3 Punkte für die anonyme Objekte Anwendungsentwickler und Maschinenführer inklusive Lebenslinie
 - 4 Punkte für Methodenbalken
 - 6 Punkte für Nachricht- und Antwortpfeile inklusive Beschriftung
 - 4 Punkte für Objektkonstruktion Steuerprogramm



2. Handlungsschritt (25 Punkte)

Bewertung

4 Punkte für die Schleife

12 Punkte, 3 x 4 Punkte für die Alternativen

2 Punkte für Umsetzung Datumwechsel

1 Punkt für Berechnung Fehleranzahl

3 Punkte für Berechnung prozentuale Abweichung

3 Punkte für Befüllen/Verwenden des TagesProtokoll-Arrays (u. a. messArt als Index)

Hinweis für Prüfer/-in

Diese Punkteverteilung gilt für diese Lösung.

Für andere Lösungen müssen die Punkte sinngemäß verteilt werden.

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

    tP := setArray(messArtAnzahl)
    abweichung: double
    abweichungProzent: double
    datum: Datum
    datum := messung[0].datum

    VON i := 0 SOLANGE i <= laenge(messung)
        WENN i != laenge(messung) UND messung[i].datum == datum
            abweichung := absolut(messung[i].sollWert -
messung[i].istWert)
            abweichungProzent := abweichung * 100 / messung[i].sollWert
            WENN abweichungProzent > maxToleranz
                tP[messung[i].messArt] += 1

        ENDE WENN
    SONST

        druckeTag(datum, tP)
        tP := setArray(messArtAnzahl)
        WENN i != laenge(messung)
            datum := messung[i].datum
        ENDE WENN
    ENDE WENN
ENDE VON i
ENDE FUNKTION druckeReport
```

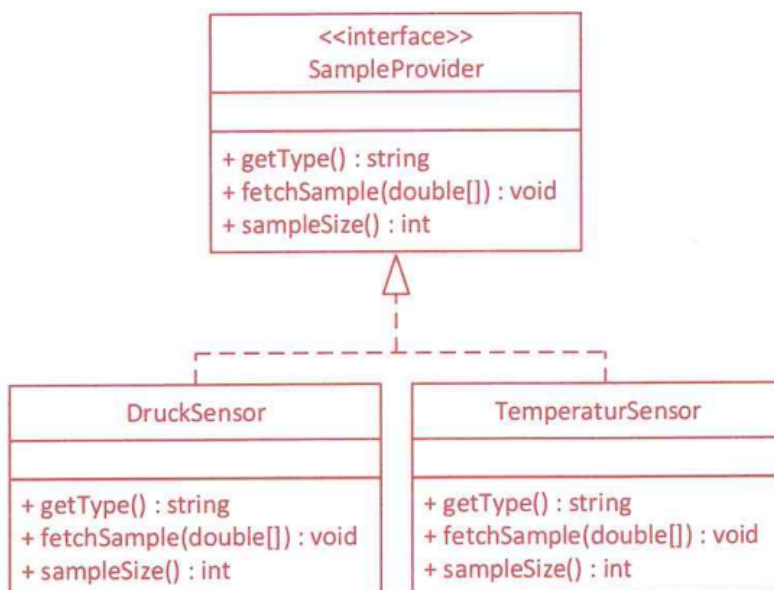
3. Handlungsschritt (25 Punkte)

aa) 6 Punkte

je 1 Punkt für *getType* und *sampleSize*, 1,5 Punkte für *fetchSample*

1,5 Punkte für Realisierungsbeziehung (1 Punkt für nichtausgefüllte Pfeilspitze, 0,5 Punkte für gestrichelte Linie)

je 0,5 Punkte für gesamten Methodenabschnitt in *DruckSensor* und *TemperaturSensor*



ab) 2 Punkte

1 Punkt Rückgabe oder return

1 Punkt "Druck"

+ getType() : string

Rückgabe: "Druck"

ac) 4 Punkte

Als Übergabeparameter wird die Referenz auf ein Dezimalzahlenarray übergeben (call by reference), somit sind die Werte nach Abarbeitung von *fetchSample* auch clientseitig zugreifbar.

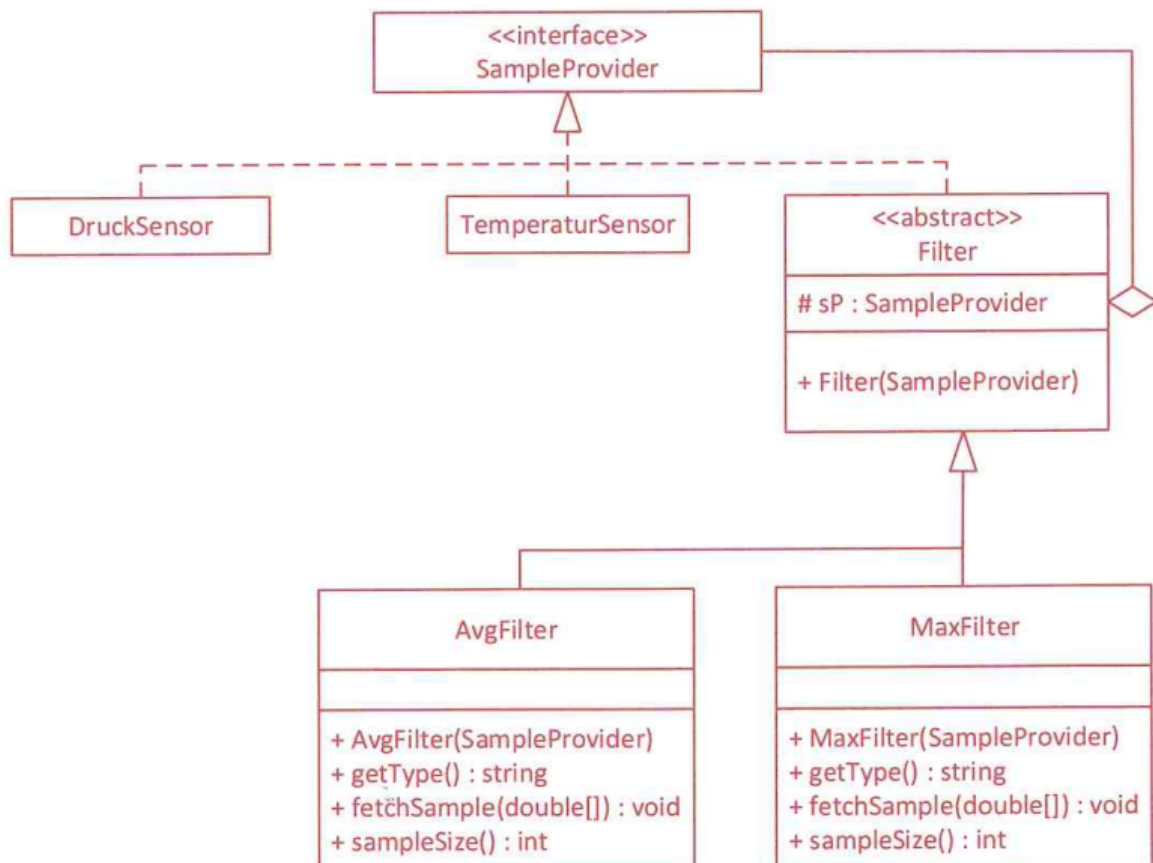
b) 6 Punkte

1 Punkt Implementierung zwischen *SampleProvider* und *Filter* (Pfeil und gestrichelte Linie)

1,5 Punkte Erweiterung (Vererbung) zwischen *Filter* und *AvgFilter*, *MaxFilter*

1,5 Punkte Aggregation zwischen *Filter* und *SampleProvider* (Raute und Linie)

je 1 Punkte für die Wiederholung der Methoden in *AvgFilter* und *MaxFilter*



ca) 3 Punkte

```
sP : SampleProvider := new DruckSensor()
```

```
sP := new AvgFilter(sP)
```

```
oder als Einzeiler sp : SampleProvider = new AvgFilter(new DruckSensor())
```


cb) 4 Punkte

je 1 Punkt pro Konstruktorkopf

je 1 Punkt pro Anweisung

```
+ Filter(sP : SampleProvider)  
this.sP := sP
```

```
+ AvgFilter(sP : SampleProvider)  
super(sP)
```

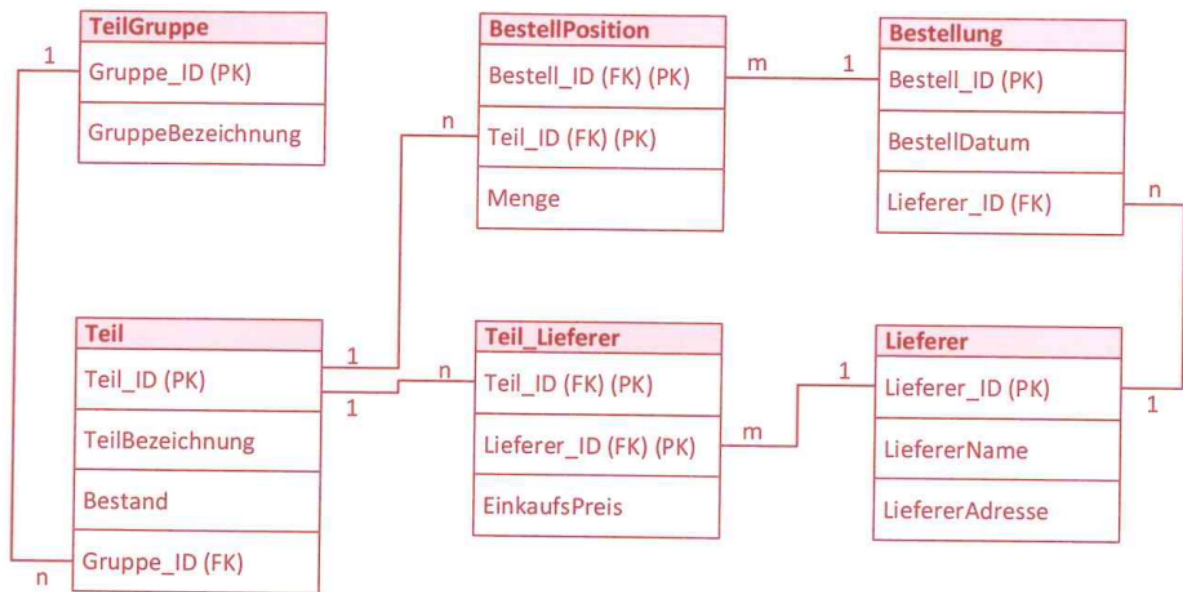
a) 20 Punkte

5 Punkte, je fehlendes Attribut 1 Punkt

4 Punkte, Tabellename 1 Punkt, je Attribut 1 Punkt

5 Punkte, je Schlüssel 0,5 Punkte

6 Punkte, je Beziehung mit Kardinalitäten 1 Punkt



b) 5 Punkte

Die 3. Normalform wird nicht erreicht, weil die 1. und 2. Normalform verletzt sind.

Die Adressdaten des Lieferers sind in einer Spalte zusammengefasst und nicht in eigenen Spalten *StrNr* bzw. Tabellen *PLZ* und *Ort* gespeichert.

5. Handlungsschritt (25 Punkte)

a) 5 Punkte

```
SELECT MT.MaschineTypeID,  
        MT.Beschreibung,  
        MT.WartungsintervallInStunden,  
        (SELECT COUNT(M.MaschineID)  
         FROM Maschine AS M  
         WHERE M.MaschineTypeID = MT.MaschineTypeID) AS AnzahlMaschinen  
FROM Maschinentyp as MT  
ORDER BY AnzahlMaschinen DESC;
```

b) 8 Punkte

```
SELECT K.KundeID ,K.KundeFirma, K.KundeAdresse ,M.MaschineID,  
      (SELECT SUM(L.Stunden) FROM Laufzeit AS L  
       WHERE L.MaschineID = M.MaschineID AND L.Datum > M.DatumLetzteWartung) AS Laufzeit  
FROM Kunde AS K  
      INNER JOIN Maschine AS M ON K.KundeID = M.KundeID  
      INNER JOIN MaschineTyp AS MT ON M.MaschineTypeID = MT.MaschineTypeID  
      WHERE (SELECT SUM(L.Stunden) FROM Laufzeit AS L  
            WHERE L.MaschineID = M.MaschineID  
            AND L.Datum > M.DatumLetzteWartung) + 100 > MT.WartungsintervallInStunden;
```

c) 8 Punkte

```
SELECT MT.MaschineTypID, MT.Beschreibung, K.KundeFirma,  
      (SELECT SUM(L.Stunden) FROM Laufzeit AS L  
       WHERE L.MaschineID = M.MaschineID AND L.Datum > M.DatumLetzteWartung) AS Laufzeit  
FROM Maschinentyp AS MT  
      LEFT JOIN Maschine AS M ON M.MaschineTypID = MT.MaschineTypID  
      LEFT JOIN Kunde AS K ON K.KundeID = M.KundeID;
```

d) 4 Punkte

```
UPDATE MaschineTyp SET WartungsintervallInStunden = WartungsintervallInStunden * 0.9;  
WHERE Beschreibung LIKE "Verpackung";
```