

# 1

## Ganzheitliche Aufgabe I Fachqualifikationen

### Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.). Wird eine bestimmte Anzahl verlangt (z. B. „Nennen Sie fünf Merkmale ...“), so ist bei Aufzählung von fünf richtigen Merkmalen die volle vorgesehene Punktzahl zu geben, auch wenn im Lösungshinweis mehr als fünf Merkmale genannt sind. Bei Angabe von Teilpunkten in den Lösungshinweisen sind diese auch für richtig erbrachte Teilleistungen zu geben.

In den Fällen, in denen vom Prüfungsteilnehmer

- keiner der fünf Handlungsschritte ausdrücklich als „nicht bearbeitet“ gekennzeichnet wurde,
- der 5. Handlungsschritt bearbeitet wurde,
- einer der Handlungsschritte 1 bis 4 deutlich erkennbar nicht bearbeitet wurde,

ist der tatsächlich nicht bearbeitete Handlungsschritt von der Bewertung auszuschließen.

Ein weiterer Punktabzug für den bearbeiteten 5. Handlungsschritt soll in diesen Fällen allein wegen des Verstoßes gegen die Formvorschrift nicht erfolgen!

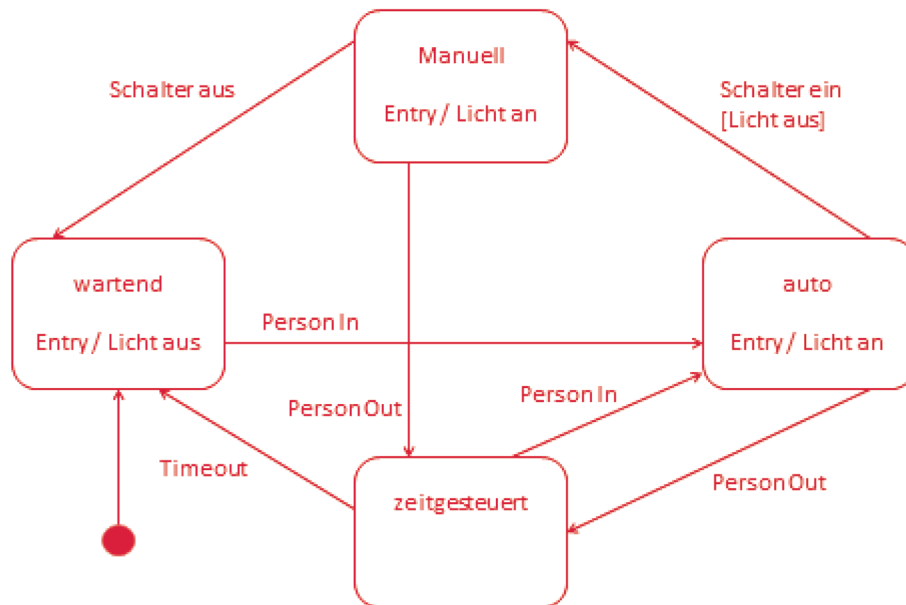
Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1	=	100 – 92 Punkte	Note 2	=	unter	92 – 81 Punkte	
Note 3	=	unter	81 – 67 Punkte	Note 4	=	unter	67 – 50 Punkte
Note 5	=	unter	50 – 30 Punkte	Note 6	=	unter	30 – 0 Punkte

## 1. Handlungsschritt (25 Punkte)

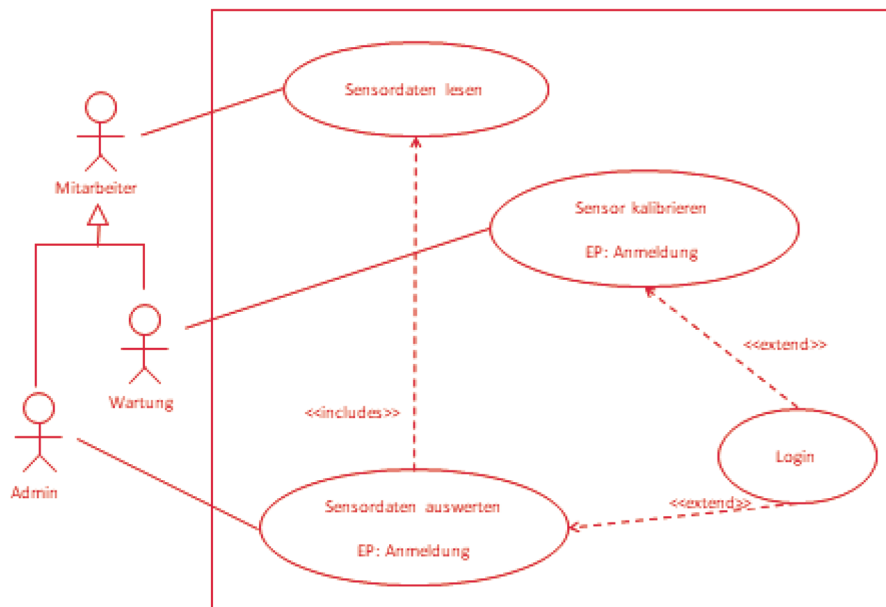
### a) Zustandsdiagramm, 15 Punkte

- |   |          |
|---|----------|
| Je Zustand 1 Punkt                                | 4 Punkte |
| Je Entry-Bedingung eines Zustandes 1 Punkt        | 3 Punkte |
| Je Transition 1 Punkt                             | 7 Punkte |
| Guard für Transition „auto“ -> „manuell“: 1 Punkt | 1 Punkt  |



### b) Use-Case-Diagramm 10 Punkte

- |  |            |
|--|------------|
| Je Anwendungsfall 0,5 Punkte                         | 2 Punkte   |
| Je Extension-Punkt 0,5 Punkte                        | 1 Punkt    |
| Je extend/include-Beziehung 0,5 Punkte               | 1,5 Punkte |
| Je Akteur 0,5 Punkte                                 | 1,5 Punkte |
| Richtige Vererbungsbeziehung                         |            |
| Mitarbeiter Basisakteur                              | 0,5 Punkte |
| Akteure Admin und Wartung von Mitarbeiter abgeleitet | 2 Punkte   |
| Je richtiger Zuordnung Akteur – Use Case 0,5 Punkte  | 1,5 Punkte |



## 2. Handlungsschritt (25 Punkte)

a) 5 Punkte

```
onNewValue(sensor_id: Integer, value: Double, timestamp: Long) {  
  
    Value v = new Value(sensor_id, value, timestamp);  
    tempList.setValue(v);  
  
}
```

b) 20 Punkte

```
maxPeriod(sensor_id: Integer, mindestwert: Double): Integer  
  
    period = 0  
    pos = 0  
  
    solange (pos < tempList.getSize(sensor_id) )  
        current_period = 0  
        solange (pos < tempList.getSize(sensor_id) &&  
            tempList.getValue(sensor_id, pos).value >=  
                mindestwert)  
            current_period++  
            pos++  
        ende solange  
        wenn (current_period > period)  
            period = current_period  
        ende wenn  
        pos++  
    ende solange  
  
    rückgabe period
```

### 3. Handlungsschritt (25 Punkte)

aa) 7 Punkte

1 Punkt Klassenname

Je 1 Punkt pro vollständige Instanzvariablen- bzw. Methodendeklaration

Tank
- bezeichner : String - fuellstand : Integer oder Double - fassungsvermoegen : Integer oder Double
+ Tank(String, Integer oder Double, Integer oder Double) + setFuellstand(Integer oder Double) : void + getFuellstand() : Integer oder Double

ab) 4 Punkte

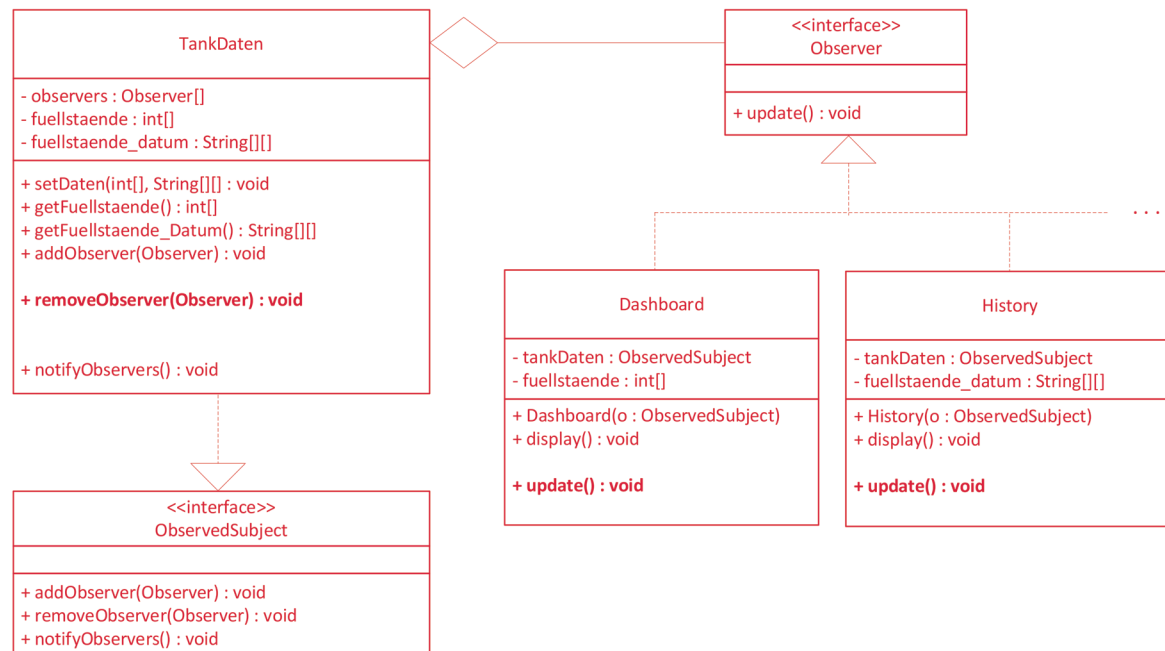
```
public oder + setFuellstand(Integer oder Double fuellstand) {this.fuellstand = fuellstand;}
```

```
public oder + getFuellstand() : {return this.fuellstand oder nur fuellstand;}
```

ba) 4 Punkte

1 Punkt je überschriebene Methode -> 3 Punkte

1 Punkt für Aggregation oder Komposition -> 1 Punkt



bb) 2 Punkte

Realisierung/Schnittstellenimplementierung:

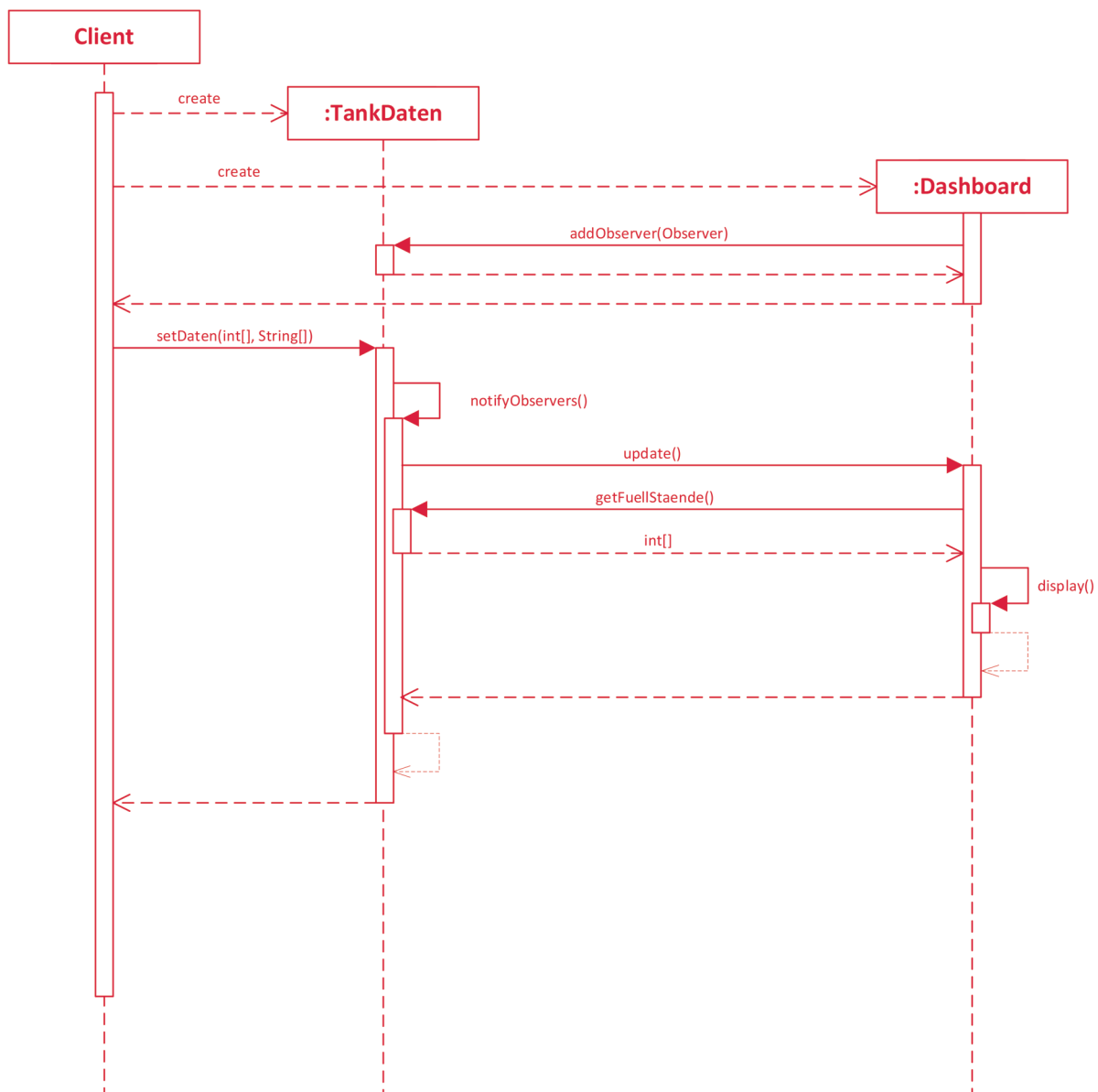
Ein Interface ist eine Vorlage-/Musterklasse, von der keine Objekte konstruiert werden können. Das ist aber von der abgeleiteten Klasse möglich.

Die Klasse *Tankdaten* muss alle abstrakten Methoden des implementierten Interface *ObservedSubject* implementieren.

ca) 6 Punkte

1 Punkte je Nachricht und zugehöriger -> 4 Punkte

0,5 Punkte je Methodenbalken -> 2 Punkte



cb) 2 Punkte

0,5 Punkte Methodenkopf, 1 Punkt ForEach-Schleife oder Zählschleife, 0,5 Punkte Methodenaufruf

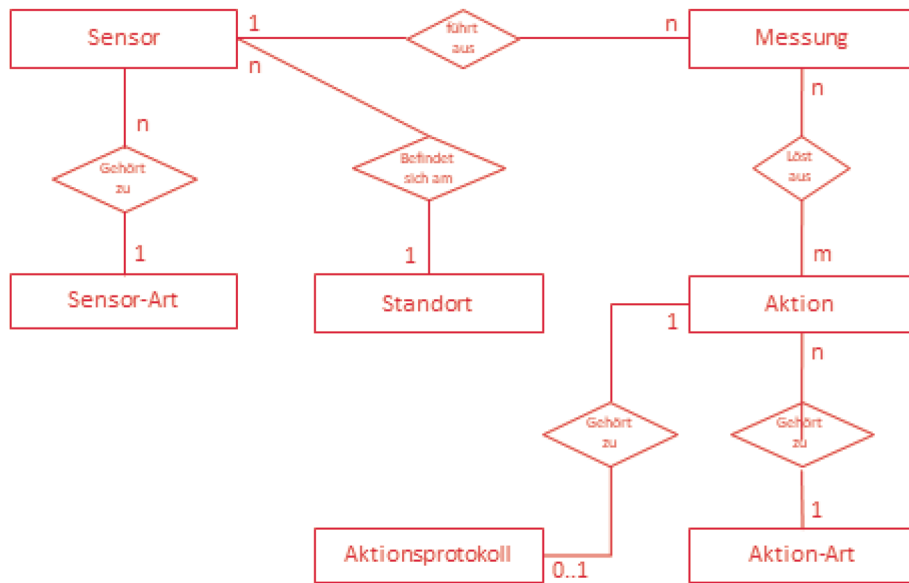
```
public oder + void notifyObservers() {for(Observer o : observers) o.update;}
```

#### 4. Handlungsschritt (25 Punkte)

a) 20 Punkte

Je richtig beschriebener Entitätstyp: 2 Punkte -> 14 Punkte

Für jede richtig beschriebene Beziehung: 1 Punkt -> 6 Punkte



b) 5 Punkte

Durch diese Vorgehensweise wird die Beschreibung der Aktion-Art redundant. Dadurch kann eine Änderungsanomalie entstehen, da eine Aktionsbeschreibung nun in mehreren Datensätzen vorkommen kann. Bei einer Änderung der Beschreibung müssten mehrere Datensätze aktualisiert werden.

## 5. Handlungsschritt (25 Punkte)

a) 5 Punkte

```
UPDATE Fehlzeit
SET      Fehlzeit.Bis_Datum = '01.07.2022',
         Fehlzeit.Grund = 'Dienstreise',
         Fehlzeit.Fehltage = 2
WHERE Fehlzeit.FZ_ID = 5;
```

b) 10 Punkte

```
SELECT
    Mitarbeiter.MA_ID, Mitarbeiter.Nachname,
    Mitarbeiter.Vorname, SUM(Fehlzeit.Fehltage)
FROM Mitarbeiter
LEFT JOIN Fehlzeit ON Mitarbeiter.MA_ID = Fehlzeit.MA_ID
WHERE
    Fehlzeit.Grund = 'Urlaub' AND
    Fehlzeit.Von_datum >= '01.01.2021' AND
    Fehlzeit.Bis_datum <= '31.12.2021'
GROUP BY Mitarbeiter.MA_ID, Mitarbeiter.Nachname, Mitarbeiter.Vorname;
```

ca) 2 Punkte

```
DROP TABLE Fehlzeit;
```

cb) 3 Punkte

```
CREATE TABLE Fehlzeitgrund(
    Grund_ID integer,
    Grund string,
    PRIMARY KEY(Grund_ID)
);
```

Formulierung mit CONSTRAINT auch möglich

cc) 5 Punkte

```
CREATE TABLE Fehlzeit(
    Fehlzeit.FZ_ID INTEGER
    Fehlzeit.MA_ID INTEGER,
    Fehlzeit.Von_Datum DATE,
    Fehlzeit.Bis_Datum DATE,
    Fehlzeit.Grund_ID INTEGER,
    Fehlzeit.Fehltage INTEGER,
    PRIMARY KEY(Fehlzeit.FZ_ID),
    FOREIGN KEY(Fehlzeit.Grund_ID) REFERENCES Fehlzeitgrund(Grund_ID),
    FOREIGN KEY(Fehlzeit.MA_ID) REFERENCES Mitarbeiter(MA_ID)
);
```

Formulierung mit CONSTRAINT auch möglich

