

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

Die WärL Chemie GmbH expandiert und plant in diesem Zusammenhang ein neues Gebäude. Für dieses Projekt übernimmt die IT-Abteilung der WärL Chemie GmbH selbst die Realisierung der Gebäudesteuerung und Wartungsarbeiten.

Sie sollen vier der folgenden fünf Aufgaben in diesem Projekt erledigen:

1. Zustandsdiagramm und Anwendungsfalldiagramm für Controller und Lichtsteuerung erstellen
2. OOP-Methoden für die Auswertung von Temperaturmessungen implementieren
3. UML-Modellierung für ein Smartphone-Dashboard zur Anzeige von Tankfüllständen anfertigen
4. ER-Modell zur Speicherung von Sensordaten anlegen
5. SQL-Abfragen für eine Zeiterfassungsdatenbank formulieren

1. Handlungsschritt (25 Punkte)

a) Für jeden Flur der Gebäude der WärL Chemie GmbH soll eine automatisierte Lichtsteuerung eingerichtet werden.

Der Controller zur Lichtsteuerung kann sich in einem der Zustände „wartend“, „auto“, „manuell“ oder „zeitgesteuert“ befinden.

- Der Controller befindet sich initial im Zustand „wartend“ bei ausgeschaltetem Licht.
- Wenn Sensoren Personen im Flur erkennen, wechselt der Controller in den Zustand „auto“ und das Licht wird mit dem Eintritt in diesen Zustand eingeschaltet.
- Wenn keine Personen mehr im Flur erkannt werden und sich der Controller im Zustand „auto“ befindet, findet ein Wechsel in den Zustand „zeitgesteuert“ statt.
- Falls innerhalb einer festgelegten Zeitspanne erneut Personen erkannt werden, erfolgt ein Wechsel vom Zustand „zeitgesteuert“ in den Zustand „auto“.
- Falls innerhalb der Zeitspanne keine Personen den Flur betreten, erfolgt ein Wechsel in den Zustand „wartend“ und das Licht wird bei Eintritt in diesen Zustand ausgeschaltet.
- Da beim Übergang vom Zustand „wartend“ in den Zustand „auto“ das Licht etwas verzögert eingeschaltet wird, kann eine Person (im Zustand „auto“) das Licht manuell einschalten, solange es noch aus ist. Dann erfolgt ein Zustandswechsel in den Zustand „manuell“ und das Licht wird mit Eintritt in diesen Zustand eingeschaltet.
- Im Zustand „manuell“ kann das Licht von einer Person ausgeschaltet werden. Der Controller wechselt dabei in den Zustand „wartend“. Falls von den Sensoren keine Personen mehr erfasst werden, erfolgt ein Übergang vom Zustand „manuell“ in den Zustand „zeitgesteuert“.

Erstellen Sie für die beschriebene Situation ein Zustandsdiagramm für den Controller der Lichtsteuerung.

15 Punkte

b) Im Rahmen der neuen Aufgaben für das Gebäudemanagement fallen für die Mitarbeiter folgende Tätigkeiten an:

- Für Wartungsmitarbeiter fallen entsprechende Wartungen und Kalibrierungen für die Sensoren an. Für die Wartung ist ein Login erforderlich, falls dieser noch nicht erfolgt ist.
- Administratoren können Sensordaten auswerten. Dazu müssen in jedem Fall Sensordaten gelesen werden. Falls dies noch nicht erfolgt ist, ist für die Auswertung der Daten ein Login erforderlich.
- Jeder Mitarbeiter kann die Sensordaten auslesen.

Erstellen Sie für die beschriebene Situation ein Anwendungsfalldiagramm.

10 Punkte

2. Handlungsschritt (25 Punkte)

In einem Gebäudeteil gibt es Sensoren, die Temperaturwerte in unregelmäßigen Zeitabständen messen. Zur Auswertung der Messwerte sollen u. a. zwei Methoden implementiert werden. Folgende Klassen sind bereits vorhanden:

Value
- sensor_id : Integer
- value: Double
- time: Long
+ Konstruktor(sensor_id: Integer, value: Double, time: Long)
+ getId() : Integer
+ getValue() : Double
+ getTime(): Long

TempList	
+setValue(value: Value)	Speichert ein Value-Objekt chronologisch in einer Liste. Die Objekte werden für jeden Sensor getrennt gespeichert.
+getValue(sensor_id, pos: Integer) : Value	Liefert für den Sensor mit der übergebenen Sensor-Id das Value-Objekt an der Position pos.
+getSize(sensor_id: Integer): Integer	Liefert die Anzahl der gespeicherten Value-Objekte für den Sensor mit der übergebenen Sensor-Id.

- a) Sobald an einem Sensor eine neue Messung vorliegt, wird automatisch die Methode
onNewValue(sensor_id: Integer, value: Double, time: Long)
aufgerufen.

Die Methode onNewValue soll mit folgender Funktionalität implementiert werden:

- Erstellen eines Value-Objektes mit den übergebenen Parametern (siehe Klassendiagramm für Value)
- Speichern des Value-Objektes mit der Methode setValue des Objektes tempList (das Objekt vom Typ TempList ist bereits vorhanden und kann verwendet werden, siehe Klassendiagramm TempList).

Implementieren Sie die Methode onNewValue in Pseudocode.

onNewValue(sensor_id: Integer, value: Double, timestamp: Long)

5 Punkte

b) Um Temperaturdaten dieses Gebäudes statistisch auswerten zu können, soll eine Methode
maxPeriod(sensor_id: Integer, mindestwert: Double): Integer
implementiert werden, die aus allen in tempList gespeicherten Temperaturwerten die höchste Anzahl von hintereinander gespeicherten Werten des Sensors ermittelt, welche den vorgegebenen Mindestwert einhalten.

Beispiel:

Es liegen die Temperaturwerte 20, 22, 23, 21, 19, 18, 20, **22, 23, 23, 24, 22**, 21 vor.

Die höchste Anzahl von hintereinanderliegenden Werten, welche den Mindestwert 22 einhalten, ist fünf.

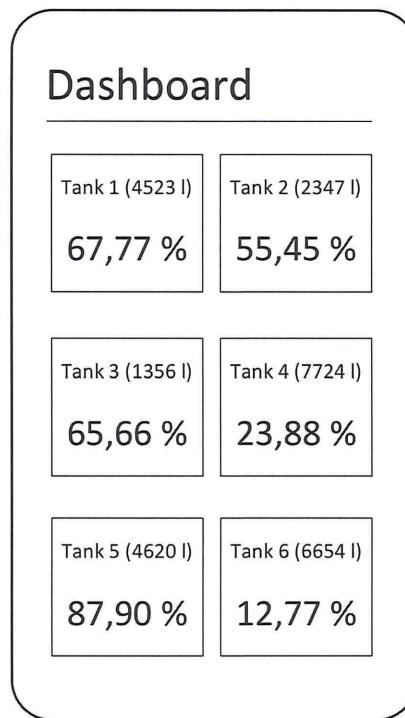
Implementieren Sie die Methode maxPeriod in Pseudocode.

20 Punkte

`maxPeriod(sensor_id: Integer, mindestwert: Double): Integer`

3. Handlungsschritt (25 Punkte)

Eine Smartphone-Anwendung mit unterschiedlichen Anzeigemöglichkeiten für Tankfüllstände soll entwickelt werden.



In einer ersten Implementierung soll eine „Dashboard“-Anzeige, welche die Füllstände aller Tanks anzeigt und später eine „History“-Anzeige, die den zeitlichen Verlauf des Füllstands eines Tanks darstellt, erstellt werden.

a) Zunächst soll eine Klasse *Tank* für Tankobjekte modelliert werden.

aa) Die Klasse *Tank* soll Folgendes beinhalten:

- Die nur klassenintern sichtbaren Instanzvariablen *bezeichner*, *fuellstand*, *fassungsvermoegen*
- Einen öffentlichen Konstruktor zur Initialisierung der Instanzvariablen.
- Beispielhaft für den *fuellstand* je eine öffentliche Set- und Get-Methode.

Hinweis: Geben Sie jeweils sinnvolle Datentypen an.

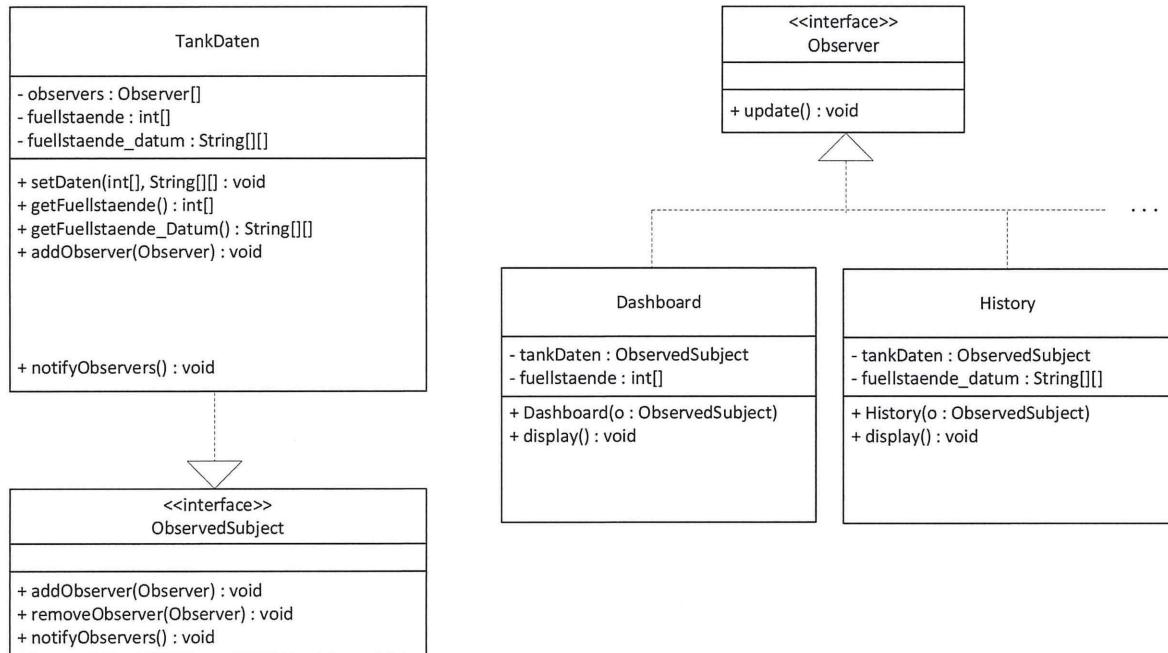
Erstellen Sie das UML-Klassendiagramm für die Klasse *Tank*.

7 Punkte

ab) Implementieren Sie in Pseudocode die Set- und Get-Methode für die Instanzvariable *fuellstand* der Klasse *Tank*. 4 Punkte

Fortsetzung 3. Handlungsschritt

- b) Die Füllstände der Tanks werden stündlich aktualisiert. Alle Anzeigen sollen entsprechend angepasst werden. Ein noch unvollständiger Entwurf nach dem Observer-Muster zur Umsetzung dieser Anforderung liegt bereits vor.



- ba) Ergänzen Sie im obigen Klassendiagramm in den Klassen *TankDaten*, *Dashboard* und *History* die jeweils fehlende Implementierung und die Beziehung zwischen der Klasse *TankDaten* und dem Interface *Observer*. 4 Punkte
- bb) Erläutern Sie die Art der Beziehung zwischen den Klassen *TankDaten* und *ObservedSubject*. 2 Punkte
-
-
-

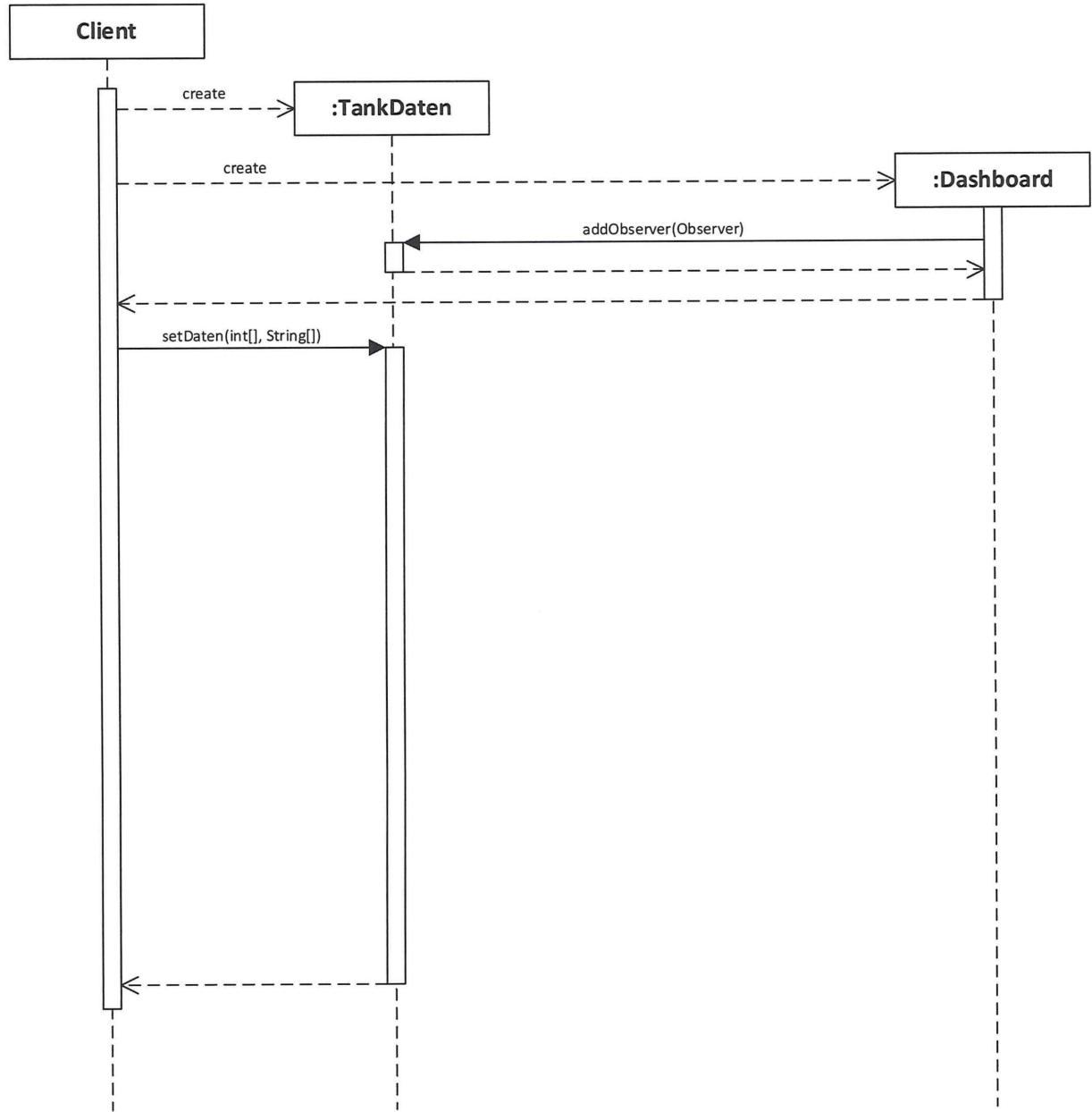
c) Zu Dokumentationszwecken soll nachfolgend beschriebener Programmablauf exemplarisch dargestellt werden.

ca) Ergänzen Sie das folgende UML-Sequenzdiagramm.

6 Punkte

- Der Client erzeugt ein *TankDaten*- und ein *Dashboard*-Objekt.
- Im Konstruktor des *Dashboard*-Objekts wird die Methode *addObserver* aufgerufen.
- Der Client ruft die Methode *setDaten* auf.
- In der Methode *setDaten* wird *notifyObservers* gestartet.
- Die Methode *notifyObservers* führt *update* aus.
- Die Methode *update* holt sich über den Aufruf der Methode *getFuellstaende* das Array mit den aktuellen Füllständen der Tanks und startet die Methode *display* zur Anzeige der Daten.
- Der Kontrollfluss geht von *display* über *update*, *notifyObservers* und *setData* zurück zum Client.

Hinweis: Die ersten drei Spiegelstriche sind bereits umgesetzt.



cb) Die Methode *notifyObservers* benachrichtigt über den Aufruf von *update* alle registrierten Observer.

Implementieren Sie *notifyObservers* in Pseudocode.

2 Punkte

4. Handlungsschritt (25 Punkte)

- a) Die WärL Chemie GmbH soll zur Speicherung der Sensordaten ein ER-Modell erstellen.

 - Jeder Sensor hat eine eigene Sensor-Art.
 - Mehrere Sensoren können von der gleichen Sensor-Art sein.
 - Ein Sensor hat einen festen/bestimmten Standort.
 - An einem Standort können verschiedene Sensoren sein.
 - Von einem Sensor können mehrere Messungen durchgeführt werden.
 - Eine Messung kann verschiedene Aktionen auslösen.
 - Eine Aktion kann von unterschiedlichen Messungen ausgelöst werden.
 - Zu jeder ausgelösten Aktion kann genau ein Eintrag im Aktionsprotokoll gehören.
 - Es gibt mehrere Aktions-Arten, die durch eine Aktion ausgelöst werden können.

Erstellen Sie ein ER-Modell. Attribute müssen nicht erfasst werden.

20 Punkte

- b) Die Tabelle Aktion-Art soll nicht mehr verwendet werden. Daher soll in der Tabelle Aktion das Attribut Art-Id entfernt werden und die Beschreibung der Aktion-Art hinzugefügt werden.

Beschreiben Sie das Problem, welches bei dieser Vorgehensweise auftreten kann.

5 Punkte

5. Handlungsschritt (25 Punkte)

Sie sollen SQL-Abfragen für folgende Zeiterfassungsdatenbank erstellen.

Mitarbeiter

MA_ID	Nachname	Vorname	Geb-Datum	Tagesarbeitszeit
811	Müller	Jens	14.04.1995	8
812	Scholz	Birgit	23.08.1977	4
815	Schmidt	Ulrich	02.11.1970	8
817	Storck	Hans	14.11.2003	6
841	Ullmann	Franz	21.12.1972	8
902	Sorge	Susanne	02.03.1965	8

Fehlzeit

FZ_ID	MA_ID	Von_Datum	Bis_Datum	Grund	Fehltage
1	811	18.10.2021	22.10.2021	Krank	5
2	902	18.10.2021	05.11.2021	Krank	16
3	811	30.12.2021	31.12.2021	Urlaub	2
4	811	03.01.2022	05.01.2022	Urlaub	3
5	815	30.06.2022	30.06.2022	Urlaub	1
6	815	03.07.2022	08.07.2022	Urlaub	6

Hinweis: Jahresübergreifender Urlaub generiert zwei Datensätze (siehe FZ_ID 3 und 4).

- a) Für den Mitarbeiter Ulrich Schmidt wurde in der Tabelle *Fehlzeit* ein Datensatz falsch erfasst:

Statt einer zweitägigen „Dienstreise“ vom 30.06. bis 01.07.2022 wurde versehentlich ein eintägiger „Urlaub“ für den 30.06.2021 eingetragen (siehe Tabelle *Fehlzeit*).

Erstellen Sie eine SQL-Anweisung, mit der die Korrektur durchgeführt werden kann.

5 Punkte

- b) Erstellen Sie eine SQL-Abfrage, welche die Urlaubstage aller Mitarbeiter im Jahr 2021 ermittelt.

10 Punkte

Beispielausgabe:

811	Müller	Jens	15
812	Scholz	Birgit	10
815	Schmidt	Ulrich	0
817	Storck	Hans	0
841	Ullmann	Franz	21

Fortsetzung 5. Handlungsschritt

- c) Die bestehende Datenbank soll wie im Folgenden beschrieben verändert werden.

Erstellen Sie dazu jeweils die SQL-Anweisung.

- ca) Die Tabelle *Fehlzeit* soll gelöscht werden.

2 Punkte

- cb) Es soll eine Tabelle *Fehlzeitgrund* mit folgenden Feldern erstellt werden.

3 Punkte

Fehlzeitgrund

Grund_ID	Grund
1	Urlaub
2	Krank
3	Dienstreise

Hinweis: Es müssen keine Datensätze eingefügt werden.

- cc) Die Tabelle *Fehlzeit* soll in der dargestellten Form neu erstellt werden. In die Tabelle *Fehlzeit* sollen in der Spalte *Grund_ID* nur solche Werte eingetragen werden können, die in der Tabelle *Fehlzeitgrund* als Primärschlüssel vorkommen. 5 Punkte

Fehlzeit

FZ_ID	MA_ID	Von_Datum	Bis_Datum	Grund_ID	Fehltage
1	811	18.10.2021	22.10.2021	1	5
2	902	18.10.2008	05.05.2021	2	16