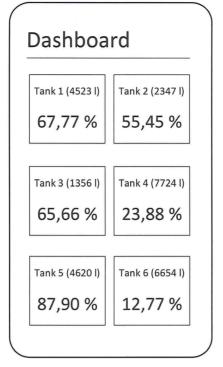
Eine Smartphone-Anwendung mit unterschiedlichen Anzeigemöglichkeiten für Tankfüllstände soll entwickelt werden.



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.

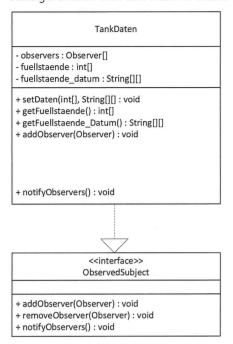
Erstellen Sie das UML-Klassendiagramm für die Klasse <i>Tank</i> .												
Implementieren Sie in Pseudocode die Set- und Get-Methode für die Instanzvariable <i>fuellstand</i> de	er Klasse <i>Tank</i> . 4 Punkte											

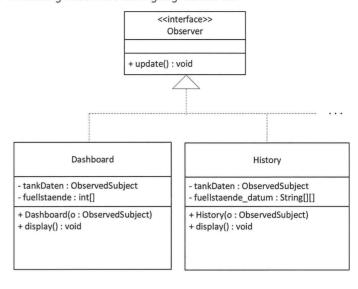
ab)	In	Implementieren Sie in Pseudocode die Set- und Get-Methode für die Instanzvariable fuellstand der Klasse Tank.														4	Pun	kte									

Fortsetzung 3. Handlungsschritt

Korrekturrand

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

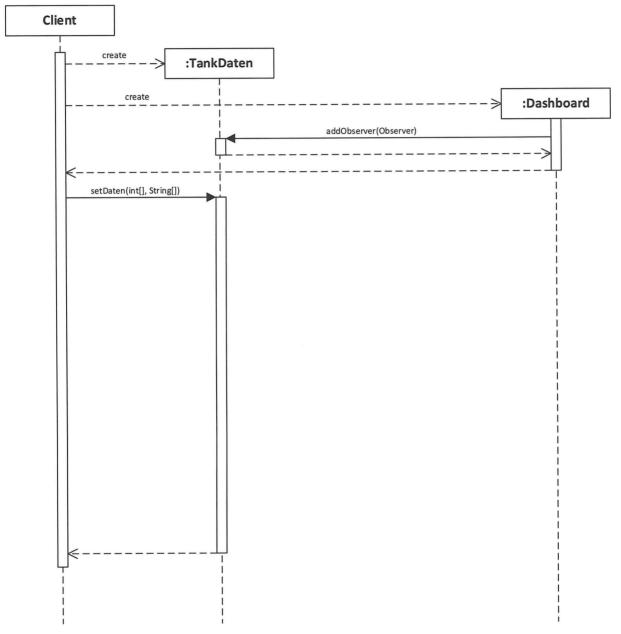
bb) Erläutern Sie die Art der Beziehung zwischen den Klassen Tankdaten und ObservedSubject.	2 Punkte				

Korrekturrand

6 Punkte

- ca) Ergänzen Sie das folgende UML-Sequenzdiagramm.
 - 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

