



Universität Augsburg
Institut für Informatik

Übung zur Vorlesung Informatik 2

SS 2018

Fakultät für Angewandte Informatik

Lehrprofessur für Informatik

PROF. DR. LORENZ, MARIUS BRENDLE, JOHANNES METZGER

07.06.2018

Übungsblatt 9

Abgabe: 13.06.2018, 12:00 Uhr (Postkasten der Veranstaltung und E-Mail an Tutor)

- Dieses Übungsblatt muss im Team abgegeben werden (Einzelabgaben sind nicht erlaubt!).
- Bitte zur Angabe von Namen, Übungsgruppe und Teamnummer das **Deckblatt** verwenden!
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

* leichte Aufgabe / ** mittelschwere Aufgabe / *** schwere Aufgabe

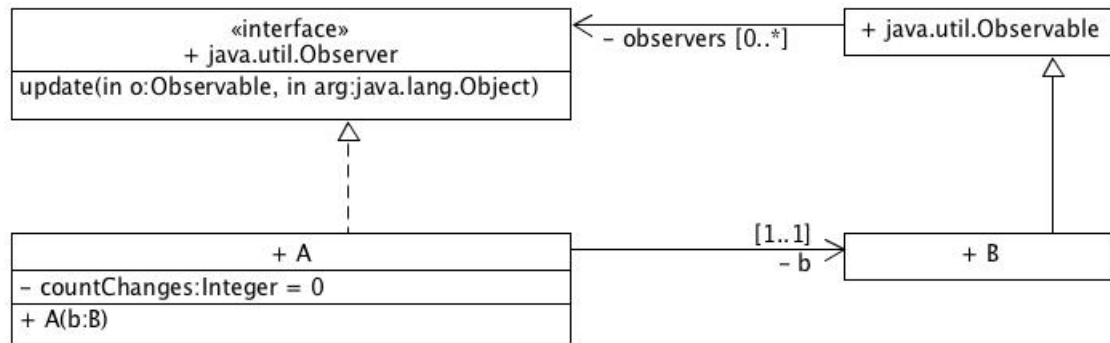
Aufgabe 33 * (*Muster in Java und UML, 14 Minuten*)

a) (*, *alte Klausuraufgabe*, 4 Punkte)

Implementieren Sie in Java eine Klasse **A** im Singletonmuster. Die Klasse soll keine weiteren Attribute oder Operationen besitzen außer diejenigen, die für das Singletonmuster notwendig sind.

b) (*, *alte Klausuraufgabe*, 6 Punkte)

Betrachten Sie das folgende Klassendiagramm:



Implementieren Sie in Java die Klasse **A**, s.d. ein Objekt der Klasse **A** als Beobachter des Referenzattributs **b** erzeugt wird, und bei jeder Änderung an **b** der interne Zähler **countChanges** um eins erhöht wird.

c) (*, *alte Klausuraufgabe*, 4 Punkte)

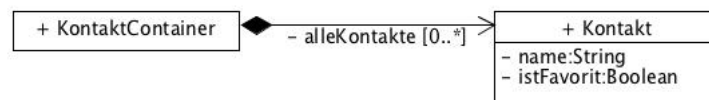
Modellieren Sie eine Klasse **A** im Singletonmuster in einem UML-Klassendiagramm.

Aufgabe 34 ** (GUI-Schicht in UML, 20 Minuten)

a) (*, alte Klausuraufgabe, 10 Punkte)

Binden Sie in einem UML-Klassendiagramm an die im Folgenden angegebenen Datenklassen einen Dialog **KontaktErfassen** zum Erfassen eines neuen Kontakts an. Er soll Eingabemöglichkeiten für alle im Modell enthaltenen Daten eines Kontakts haben. Durch Klick auf einen Menüeintrag **Erfassen** in einem Menü **Aktionen** soll der Benutzer einen Kontakt mit den eingegebenen Daten anlegen können.

Modellieren Sie den Dialog **KontaktErfassen** mit allen seinen für die Eingabe und die Ereignisbehandlung notwendigen Verbindungen zu Klassen und Schnittstellen der Java-API. Geben Sie dabei alle für die Eingabe und die Ereignisbehandlung notwendigen Operationen der verwendeten Klassen und Schnittstellen der Java-API an.



b) (*, alte Klausuraufgabe, 10 Punkte)

Binden Sie an die im folgenden UML-Klassendiagramm angegebenen Datenklassen einen schließbaren Dialog **KontakteAnzeigen** zum Anzeigen aller Kontakte in Form einer Liste von Zeichenketten an. Die Anzeige soll sich bei Änderungen an der Liste der Kontakte automatisch aktualisieren und soll neben dem Schließen auf keine weiteren Ereignisse reagieren.

Modellieren Sie den Dialog **KontakteAnzeigen** mit allen für die Anzeige, deren Aktualisierung und die Ereignisbehandlung notwendigen Klassen und Schnittstellen der Java-API und den notwendigen Verbindungen zwischen den modellierten Klassen und Schnittstellen. Geben Sie dabei alle für die Anzeige, deren Aktualisierung und die Ereignisbehandlung notwendigen nicht privaten Attribute und Operationen an.



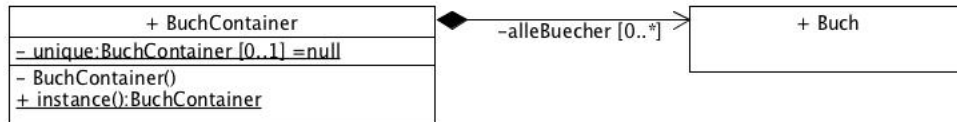
Aufgabe 35 * (Einfache Implementierung von Containerklassen, 20 Minuten)

a) (*, 10 Minuten)

Implementieren Sie vollständig eine Containerklasse zur Datenklasse `GeometricObject` aus Kapitel 06 unter Benutzung des Singletonmusters und des Iteratormusters ohne Anbindung einer Datenhaltungs- und GUI-Schicht.

b) (*, 5 Minuten)

Implementieren Sie die im Singletonmuster angegebene Klasse `BuchContainer` aus dem folgenden Klassendiagramm.

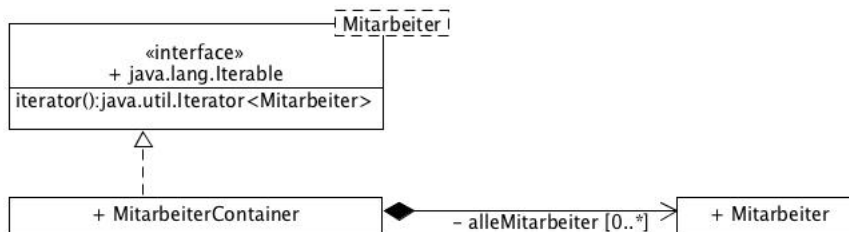


Hinweise:

- Geben Sie keine Importe an.
- Implementieren Sie keine Attribute oder Operationen, die nicht direkt dem Klassendiagramm entnommen werden können.

c) (*, 5 Minuten)

Implementieren Sie die Klasse `MitarbeiterContainer` aus dem folgenden Klassendiagramm. Berücksichtigen Sie dabei auch ggf. notwendige Importe.



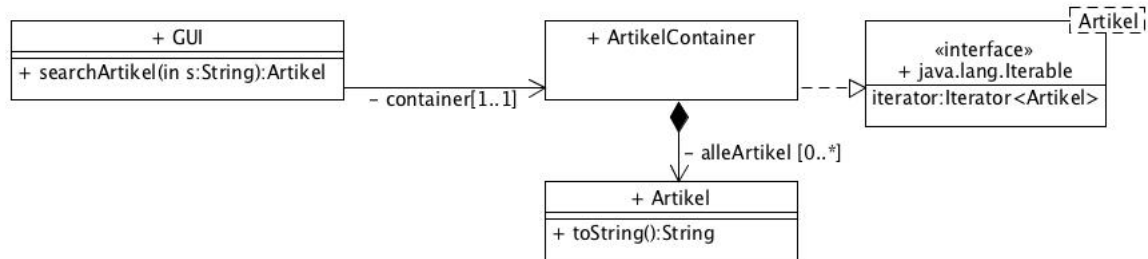
Hinweise:

- Implementieren Sie keine Attribute oder Operationen, die nicht direkt dem Klassendiagramm entnommen werden können.

Aufgabe 36 ** (Sequenzdiagramm modellieren, 16 Minuten)

a) (**, 8 Minuten)

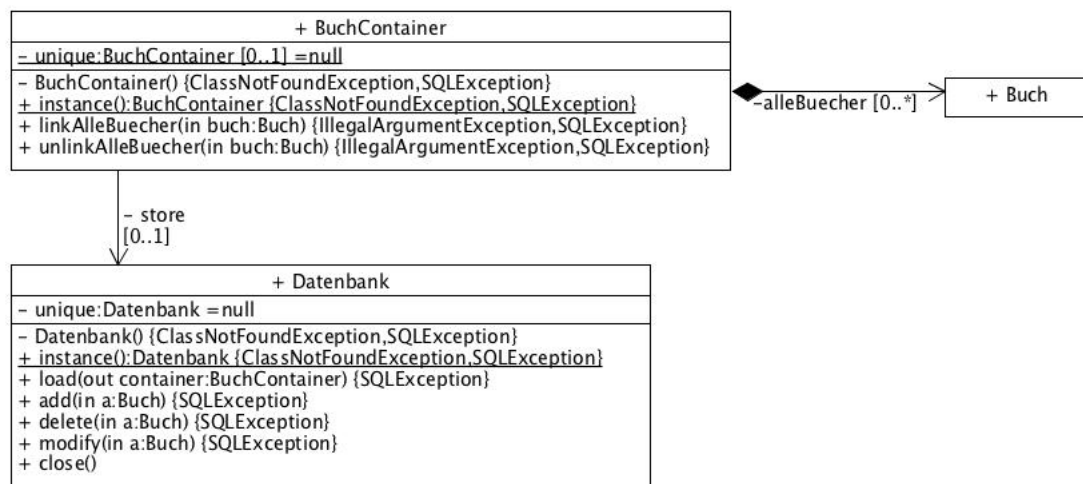
Betrachten Sie das folgende Klassendiagramm:



Implementieren Sie die Operation `searchArtikel` als Sequenzdiagramm, s.d. diese ein **Artikel**-Objekt sucht und zurückgibt, dessen Repräsentation als Zeichenkette der übergebenen Zeichenkette entspricht, bzw. `null` zurückgibt, falls kein solches **Artikel**-Objekt existiert.

b) (**, 8 Minuten)

Betrachten Sie das folgende Klassendiagramm:



Implementieren Sie den Konstruktor der Klasse **BuchContainer** als Sequenzdiagramm, s.d. bei Erzeugung des Containers alle seine Attribute initialisiert werden und die Daten aus der angebundenen Datenbank in den Container geladen werden.