



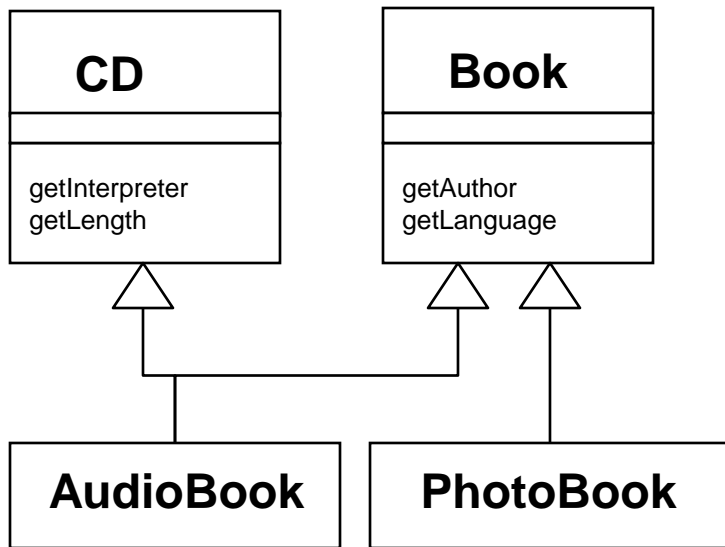
Übung „Einführung in die Softwaretechnik“

Lösungshinweise zum Übungsblatt 13

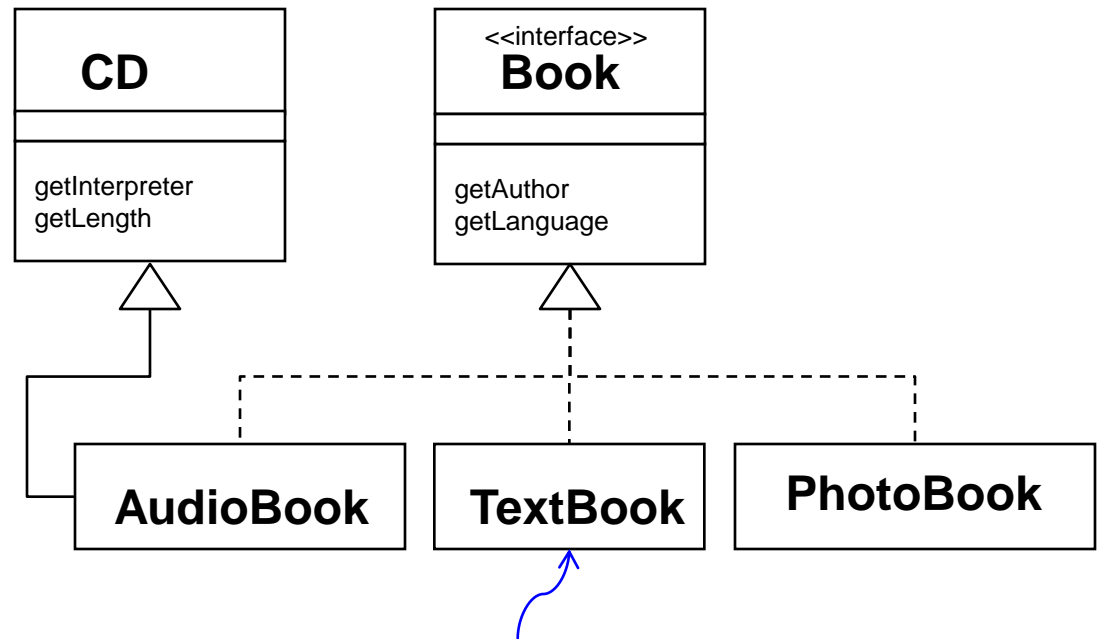


Aufgabe 32a

Logischer Entwurf:



Feinentwurf für Java:



*Alte Klasse Buch umbenennen oder
anderen Name für das Interface wählen*

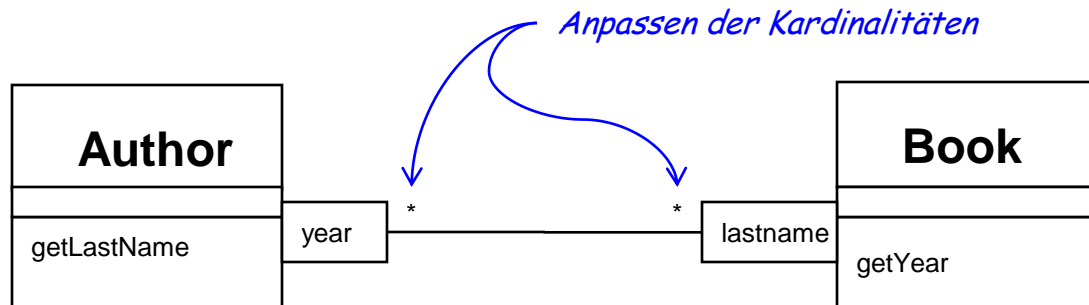


Aufgabe 32b

Ausgangssituation:



Qualifizierte Assoziationen:





Aufgabe 32b – Code - Datenstrukturen

```
public class Author {
```

```
    private String lastName;
```

Schlüssel

```
    private Map<Integer, Collection<Book>> books;
```

Wert

```
    public Author(String lastName) {
```

```
        books = new HashMap<Integer, Collection<Book>>();
```

```
        this.lastName = lastName;
```

```
    }
```

```
    ...
```

```
}
```



Aufgabe 32b – Code - Methoden

```
public void addBook(Book book) {  
    Collection<Book> yearbooks = books.get(book.getYear());  
  
    if (yearbooks == null) {  
        yearbooks = new ArrayList<Book>();  
        books.put(book.getYear(), yearbooks);  
    }  
  
    if (!yearbooks.contains(book)) {  
        yearbooks.add(book);  
        book.addAuthor(this);  
    }  
}
```

Passende Liste nach Jahr bestimmen

Liste anlegen falls noch keine vorhanden

Endlosschleife verhindern bei Buch hinzufügen und Autor dem Buch hinzufügen

```
public Collection<Book> getBooksByYear(int year) {  
    return books.get(year);  
}
```

*Bücher des Jahres auslesen.
! Kann auch „null“ liefern !*



Aufgabe 32c – Code - BuchInfo

```
public class BuchInfo {  
  
    private String titel;  
    private Date erscheinungsdatum;  
    private Collection<Buch> exemplare;  
  
    public BuchInfo() {  
        exemplare = new ArrayList<Buch>();  
    }  
  
    public String getTitel() {  
        return titel;  
    }  
  
    public void setTitel(String titel) {  
        this.titel = titel;  
    }  
  
    // Analog für erscheinungsdatum
```

Realisierung der Assoziation in der Klasse der Beschreibung

Abgeleitetes Attribut wird berechnet!

```
    public Collection<Buch> getExemplare()  
    {  
        return exemplare;  
    }  
  
    public void addExemplar(Buch buch) {  
        exemplare.add(buch);  
        buch.setBeschreibung(this);  
    }  
  
    public int getBestand() {  
        return exemplare.size();  
    }  
}
```



Aufgabe 32c – Code - Buch

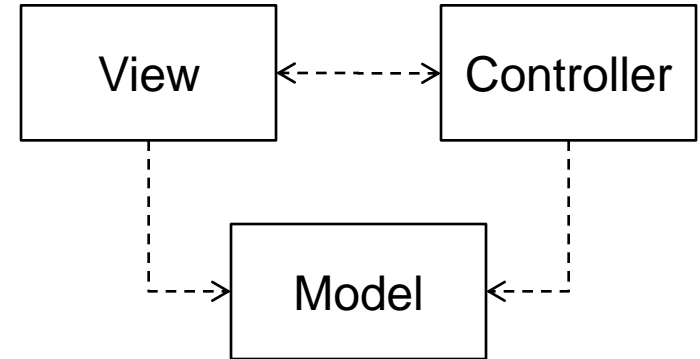
```
public class Buch {  
  
    private int preis;  
    private BuchInfo beschreibung;  
  
    public int getPreis() {  
        return preis;  
    }  
  
    public void setPreis(int preis) {  
        this.preis = preis;  
    }  
  
    public BuchInfo getBeschreibung() {  
        return beschreibung;  
    }  
  
    public void setBeschreibung(BuchInfo beschreibung) {  
        this.beschreibung = beschreibung;  
    }  
  
}
```

Realisierung der Assoziation in der Klasse der Exemplare



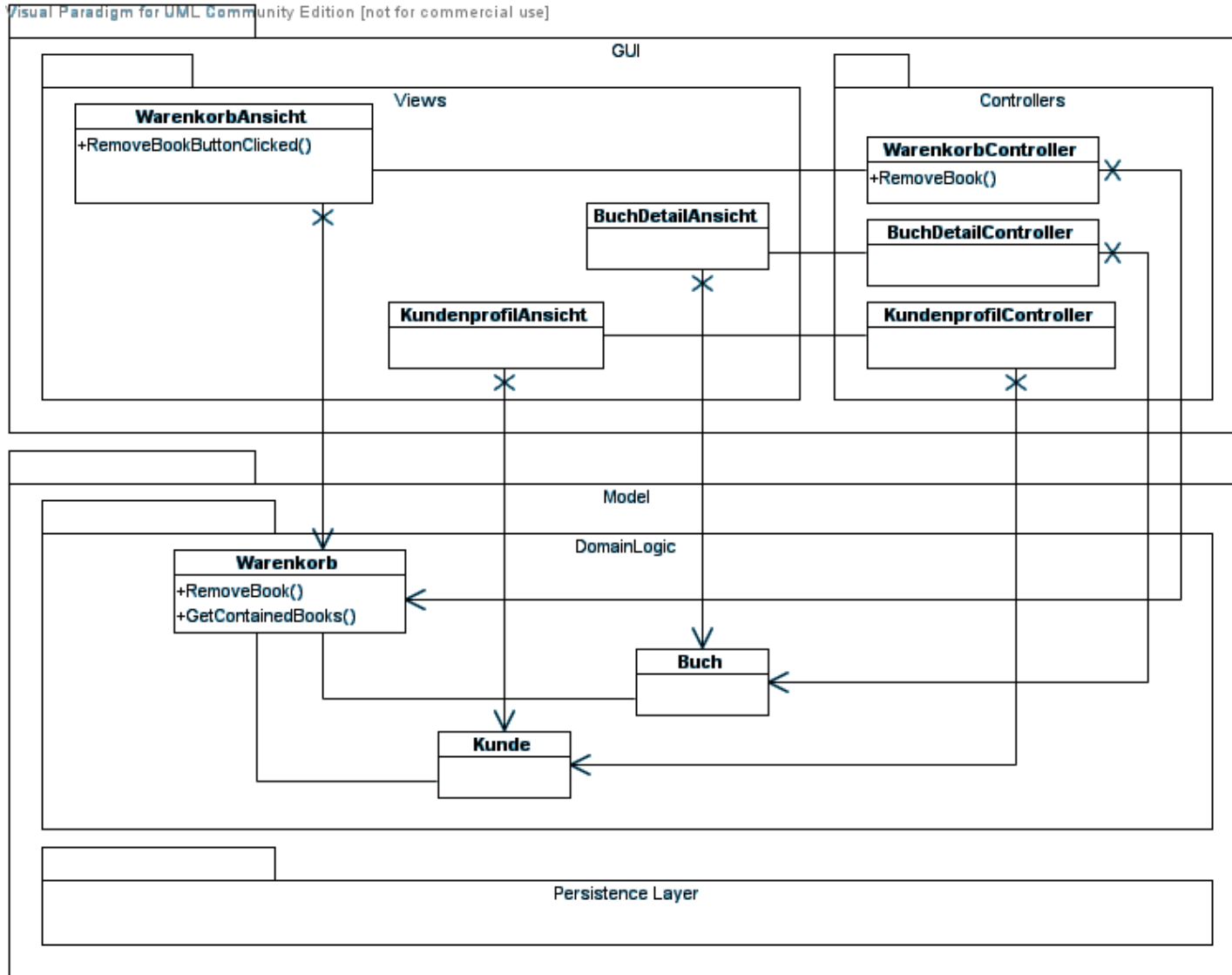
Aufgabe 33 a)

- **Model**
 - Kernfunktionalität und Daten
 - Fachlicher Kern und Persistenzschicht
 - Informiert View(s) im Fall von Änderungen an den Daten.
- **View(s)**
 - Darstellung der Informationen
 - Teil der Benutzungsschnittstelle
 - Leitet Benutzereingaben an den entsprechenden Controller weiter.
 - Holt Daten vom Modell im Fall einer Änderung und aktualisiert die Darstellung.
- **Controller(s)**
 - Behandlung von Benutzereingaben
 - Teil der Benutzungsschnittstelle
 - Nimmt Benutzereingaben von der entsprechenden View entgegen.
 - Veranlasst datenunabhängige Aktualisierung der View(s) und/oder
 - Ruft Änderungsoperationen des Modells auf.



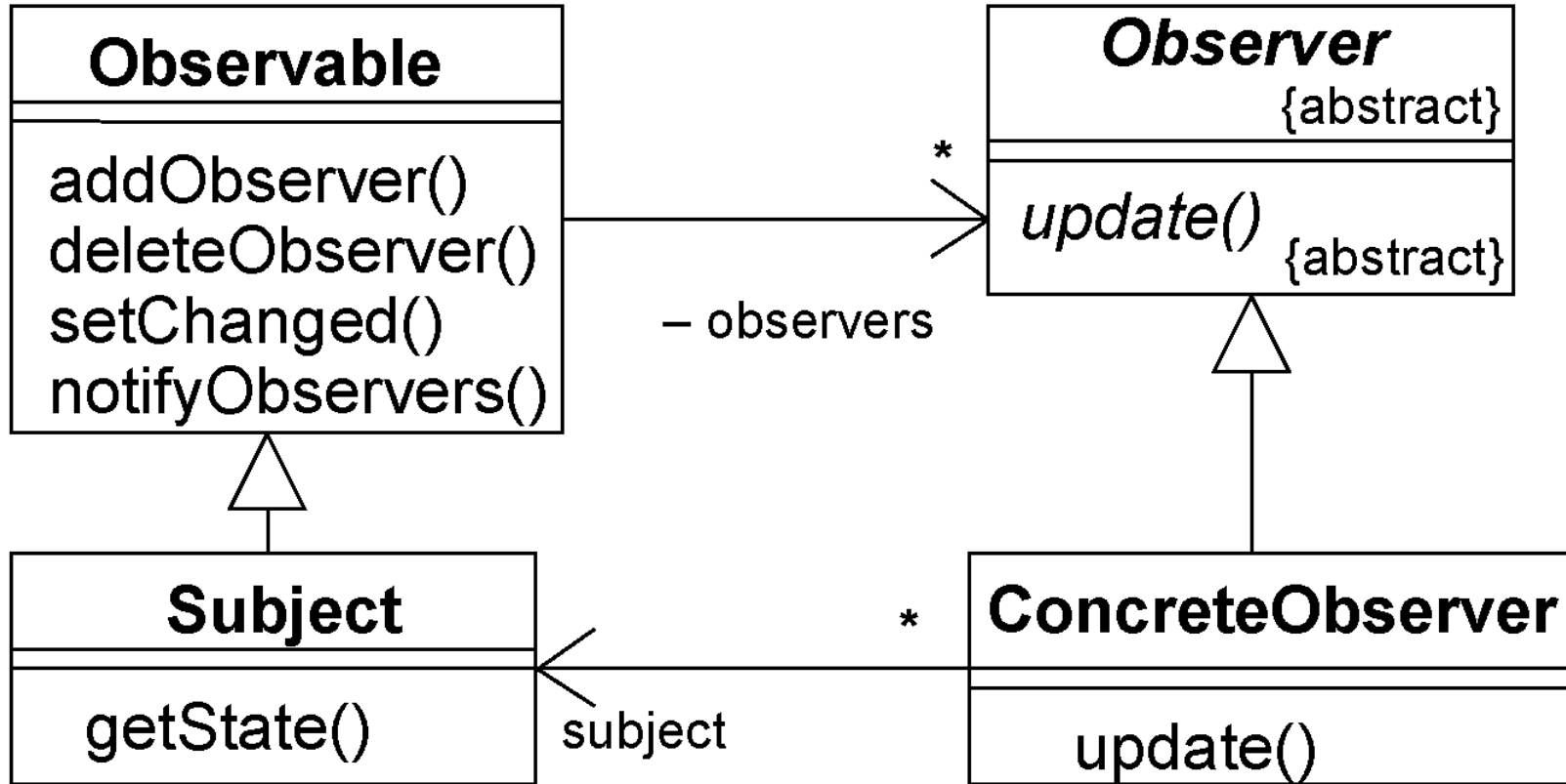


Aufgabe 33 b)



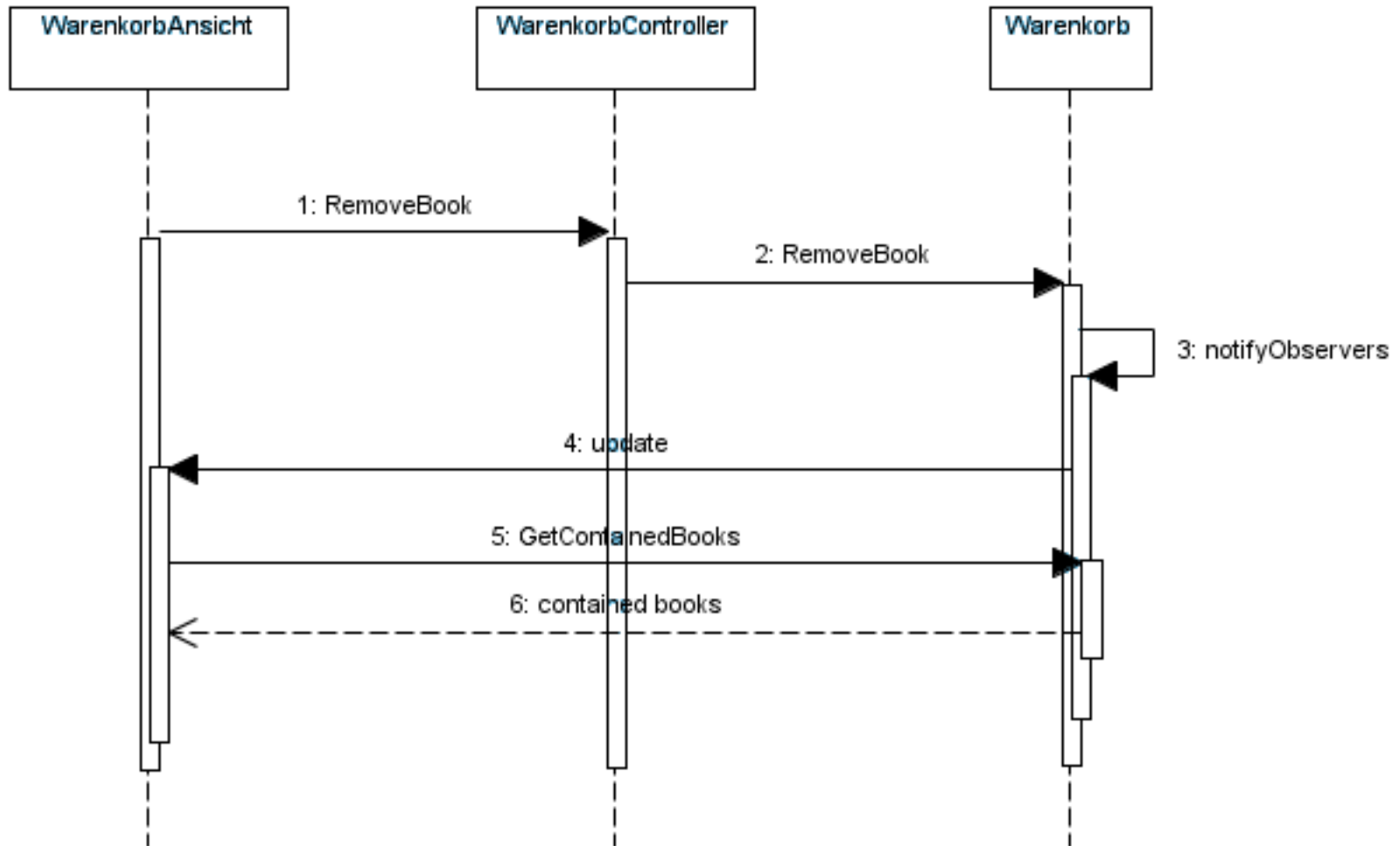


Aufgabe 33 c)





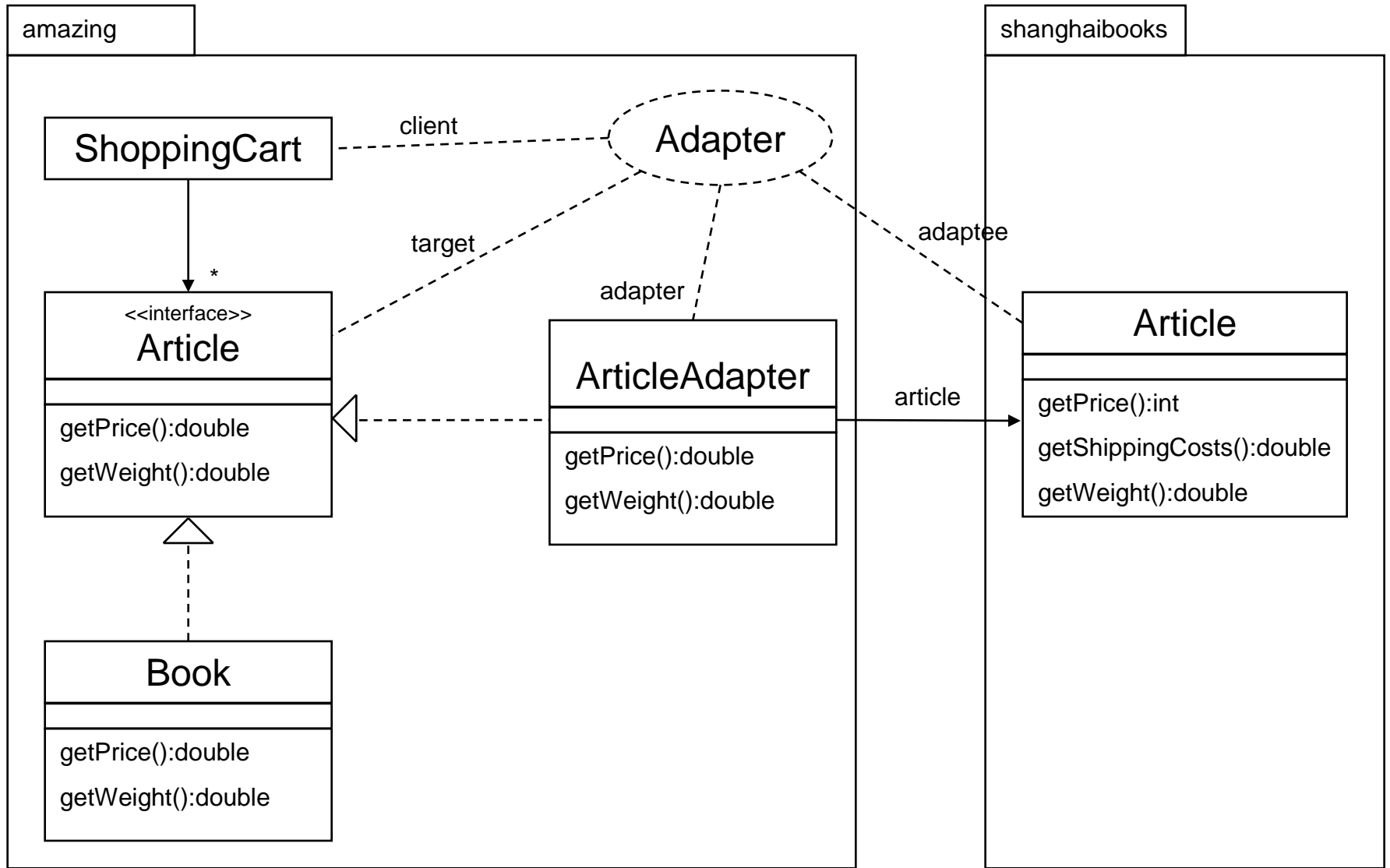
Aufgabe 33c)





Aufgabe 34

a)





Aufgabe 34

b)

