

Aufgabe 1: Sammlungen in Sammlung

In der Klasse Bibliothek (siehe Blatt 1) werden Objekte der Klasse Buch in zwei Sammlungen verwaltet, d.h. ein Buch-Objekt ist in der Bibliothek immer in beiden Sammlungen enthalten.

Die Hashmap im Attribut `autorenSammlung` enthält als Wert Objekte vom Typ `ArrayList<Buch>`, in denen die Bücher eines Autors gespeichert sind. Diese Bücherlisten werden in der Hashmap `autorenSammlung` unter dem Namen des Autors abgespeichert.

Die Hashmap im Attribut `titelSammlung` enthält alle Bücher der Bibliothek, wobei der Titel des Buches als Schlüssel für den Zugriff auf die zugehörige Instanz der Klasse `Buch` verwendet wird.

Schreiben Sie die Methode `loeschen(String autor, String titel)`, die den String mit dem Namen des Autors und den Titel des Buches übergeben bekommt und das zugehörige Buch aus der Bibliothek löscht. Es sollen in der Bibliothek nur Autoren verwaltet werden, von denen mindestens ein Buch in der Bibliothek ist. (9 Punkte)

Tipp: Eine kurze Zusammenfassung der wichtigsten Methoden für die Handhabung der Klasse `HashMap` finden Sie auf Blatt 5.

Aufgabe 2: Exceptions

a) Schreiben Sie die Methode

```
public void setzeInfos(String autor, String titel, boolean hardc,
int anzS)
```

der Klasse `Buch` (siehe Blatt 2), die Attribute des Buch-Objekts mit den Werten aus den Methodenparametern belegt. Die Parameter `titel` und `autor` sollen vor der Zuweisung daraufhin geprüft werden, ob sie ungültige Angaben enthalten (Wert `null` bzw. kein Zeichen oder nur Leerzeichen). Ist einer der beiden Werte ungültig, so soll eine `UnguelzigesBuchException` geworden werden. (7 Punkte)

Lösung:

```
public void setzeInfos (String autor, String titel, boolean hardc, int anzS)
    throws UnguelzigesBuchException {
    if (((titel == null) || (titel.trim().equals(",")) || ((autor == null) || (autor.trim().equals(",")))) {
        throws new UnguelzigesBuchException (????) {

    else {
        this.autor = autor;
        this.titel = titel;
        this.hardcover = hardc;
        this.seitenAnzahl = anzS;
    }
}
```

b) Schreiben Sie für die Klasse *Bibliothek* (siehe Blatt 1) die Methode `hinzufuegen(. .)`, die es erlaubt, ein neues *Buch*-Objekt in die Bibliothek einzufügen. Ist das Buch bereits in der Bibliothek vorhanden, so soll eine *BuchBereitsVorhandenException* geworfen werden. Denken Sie daran, dass Sie eine neue *ArrayList* für die Bücher des Autors erstellen und in die entsprechende Sammlung einfügen müssen, falls noch gar kein Buch des Autors in der Bibliothek enthalten ist. (15 Punkte)

Aufgabe 3: Datei Ein-/Ausgabe

Schreiben Sie die Methode `boolean laden(File path)` der Klasse *Bibliothek* (siehe Blatt 1). Diese bekommt den Pfad zu einer *.ser*-Datei übergeben, in der ein Array mit *Buch*-Objekten gespeichert ist. Die *Buch*-Objekte aus dem Array sollen zu der Bibliothek hinzugefügt werden. Die Methode gibt den Wert *true* zurück, wenn das Laden geklappt hat, sonst den Wert *false*. Sollten keine *Buch*-Objekte in der Datei gespeichert sein, so soll der Inhalt der Bibliothek nicht verändert werden. (18 Punkte)

Aufgabe 4: GUI 1

Die Abbildungen unten zeigen die *View*, die in der Klasse *BibliothekViewController* (siehe Blatt 3) initialisiert wird. Das Menü *Datei* erlaubt das Laden und Speichern von Büchern. Das Menü *Anzeigen* steuert das *Anzeigen* der Bücher in der *ListView*. Das Menü *bearbeiten* erlaubt es ein neues Buch anzulegen oder eines zu löschen.

Die Klasse *BibliothekViewController* enthält im Attribut `bibliothek` eine Instanz der Klasse *Bibliothek*, mit der der *BibliothekViewController* arbeiten soll.

a) Schreiben Sie die Methode `configureLoeschenMenu()`, die das Menü im Attribut `loeschenMenu` aufbaut. Die Methode wird aufgerufen, wenn die *View* initialisiert wird oder auch wenn das Menü aktualisiert werden muss. Die als Untermenüs angezeigten Autorennamen, sowie die Bücher, die als Menü-Items eingefügt werden, müssen aus dem *Bibliothek*-Objekt im Attribut `bibliothek` bezogen werden. Bei allen Menü-Items soll das Event Handler die Methode

```
privat void handleTitleSelection(ActionEvent e)
```

gemeldet werden. (12 Punkte)

b) Ergänzen Sie die Klasse *BibliothekViewController* (siehe Blatt 3) um die Methode `private void handleTitleSelection(ActionEvent e)`.

Diese wird aufgerufen, wenn im Menü „Buch Löschen“ der Titel eines Autors zum Löschen ausgewählt werden. Sorgen sie dafür, dass das zugehörige Buch aus der Bibliothek gelöscht wird. Danach muss das Menü sowie die *ListView* aktualisiert werden. Denken Sie daran, dass im *TextField* eventuell vorher ein Suchbegriff eingetragen wurde und die Liste der angezeigten Bücher dem Suchbegriff entsprechen sollte. (9 Punkte)

Aufgabe 5: GUI 2

View für das Anlegen bzw. Bearbeiten eines *Buch*-Objekts ist mit dem *ViewController* (siehe Blatt 4) gekoppelt. Der Konstruktor der Klasse *ViewController* bekommt im Methodenparameter *buch* den Wert null übergeben, wenn ein neues Buch angelegt und eine Instanz der Klasse *Buch*, wenn ein *Buch*-Objekt bearbeitet werden soll. In der Methode `initialize()` wird, je nachdem mit welchem Wert das Attribut *buch* initialisiert wurde, die GUI unterschiedlich konfiguriert.

Tipp: Bei einer Instanz von *CheckBox* lässt sich mit `boolean isSelected()` abfragen, ob die Checkbox selektiert ist, mit `void setSelected(boolean s)` lässt sich der Wert setzen.

a) Schreiben Sie die Methode `initBuchAnlegen()` und `initBuchBearbeiten()` so, dass die GUIs wie oben abgebildet und unten beschrieben initialisiert werden.

initBuchAnlegen(): Der Fenstertitel wird gesetzt, rechter Button (`speicherButton`) bekommt die Aufschrift „Speichern“, die Methode `speichern(..)` wird dort als Event Handler angemeldet.

initBuchBearbeiten(): Der Fenstertitel wird gesetzt, rechter Button (`speichernButton`) bekommt die Aufschrift „Aktualisieren“, die Eingabemaske wird mit den Informationen aus dem Buch in *buch* gefüllt und als Event Handler wird bei dem rechten Button die Methode `speichern()` angemeldet. (14 Punkte).

b) Schreiben Sie die Methode `speichern(..)` (siehe Blatt4). Diese wird aufgerufen, wenn man auf den Button mit der Aufschrift „Aktualisieren“ bzw. „Speichern“ gedrückt hat. Finden Sie heraus, ob das *Buch*-Objekt aus den Angaben im Fenster gespeichert oder aktualisiert werden muss und nutzen Sie dafür die passenden Methoden der Klasse *Bibliothek*. Zeigen Sie die hierbei möglicherweise auftretenden Exceptions mit Hilfe der Klasse *ViewHelper* (so wie in der Übung verwendet) an. Wenn alles gut gegangen ist, sollte das Fenster wieder geschlossen werden. (16Punkte)