



## Hausaufgabe 1

**Hinweis:** Für die Bearbeitung dieser Hausaufgabe ist Gruppenarbeit nicht zulässig! Bitte bedenken Sie dabei, dass Plagiate mit entsprechender Analysesoftware leicht zu erkennen sind. Bereits eine abgeschriebene Hausarbeit führt zum Nicht-Bestehen des gesamten Moduls. Bitte beachten Sie die eidesstattliche Erklärung am Ende dieser Aufgabenstellung. Sie müssen diese unterzeichnet mit Ihrer Lösung gemeinsam abgeben.

Abzugeben ist der Quelltext des Java-Programms (bitte übermitteln Sie eine jar-Datei inkl. der in dem Projekt vorhandenen Java-Dateien). Notieren Sie bitte im Quelltext Ihres Programmes Ihren **Namen** und Ihre **Matrikel-Nr.** als Kommentar zu Beginn jeder Java-Datei.

Bitte geben Sie diese Hausaufgabe **bis Freitag, den 17. Mai 2024 um 23:59 Uhr** per Upload unter <https://lms.hs-osnabrueck.de> in Rubrik Hausarbeiten / HA1 der Lehrveranstaltung OOP2 des aktuellen Semesters ab. **ACHTUNG: Das jar-File (HA1-IhrNachname.jar) muss die .java-Dateien enthalten.** Ebenfalls mit abzugeben ist die unterzeichnete **Eidesstattliche Erklärung**, die auf der letzten Seite Aufgabestellung aufgeführt ist. Ohne eine unterzeichnete eidesstattliche Erklärung kann die Hausarbeit nicht bewertet werden.

### Aufgabe: Online-Buchungs- und Ausleihsystem für die Bibliothek „Page Turner“ (20 Punkte)

#### Szenariobeschreibung

Die Bibliothek „Page Turner“ möchte ihre Angebote modernisieren und plant, ein Online-Buchungs- und Ausleihsystem einzuführen. Bisher mussten Mitglieder Bücher vor Ort ausleihen, was zu Unannehmlichkeiten führte, wenn gewünschte Bücher nicht verfügbar waren. Das Bibliotheksmanagement hat Sie gebeten, ein System zu entwickeln, das es Mitgliedern ermöglicht, Bücher online zu reservieren und auszuleihen.

Im Gespräch mit den Mitarbeiterinnen der Bibliothek besprechen Sie die folgenden Details und Anforderungen:

- Ein Buch hat eine interne BuchId, eine ISBN, einen Titel, ein Veröffentlichungsdatum, eine/n Autorin, eine Genrebezeichnung und eine Angabe darüber, wie viele Exemplare des Buches verfügbar sind. Die verfügbaren Bücher für den Monat April 2024 sind in der CSV-Datei „buecher.csv“ aufgelistet.
- Bibliotheksmitglieder haben eine Mitgliedsnummer (Integer), einen Vornamen, einen Nachnamen, eine E-Mail-Adresse, ein Geburtsdatum und ein Passwort (Buchstaben und



Zahlen). Die aktuellen Mitgliederdaten sollen sich in der Datei „mitglieder.csv“ befinden. Erzeugen Sie sich gern eine erste Version dieser Datei mit KI-Unterstützung.

- Des Weiteren gibt es Reservierungen, die beschreiben, welche Bücher von welchen Mitgliedern bis zu welchem Datum reserviert oder ausgeliehen wurden. Diese Daten sollen sich in der Datei „reservierungen.csv“ befinden. Vermutlich macht es Sinn, wenn Sie sich eine erste Version der Datei mit wenigen Reservierungen händisch erzeugen.
- Ihre Abgabe muss die Dateien „buecher.csv“, „mitglieder.csv“ und „reservierungen.csv“ mit jeweils mindestens fünf Entitäten enthalten.

### **Grundlegende Anforderungen:**

#### **AWF 1: Programm starten und Anmelden**

- Beim Programmstart sollen alle Mitglieder, Bücher und Reservierungen aus den CSV-Dateien geladen werden. Erstellen Sie dafür geeignete Klassen und speichern Sie diese Objekte in passenden Verwaltungsklassen (z.B. `MitgliederVerwaltung`, `BuchVerwaltung` & `ReservierungsVerwaltung`).
- Für das Laden der Mitglieder soll eine Methode `Collection<Mitglied> read(String fileName)` in der Klasse `MitgliederVerwaltung` angelegt werden. Die Datei zur Speicherung der Mitglieder hat den folgenden Aufbau: Mitgliedsnummer, Vorname, Nachname, E-Mailadresse, Geburtsdatum, Passwort.
- Analog soll das Laden der Bücher und auch der Reservierungen über entsprechende Verwaltungsklassen erfolgen. Der Aufbau der CSV-Dateien ist dabei wie folgt:
  - Bücher: `BuchId,ISBN,Titel,Veröffentlichungsdatum,Autor,Genre,Verfügbare Exemplare`
  - Reservierungen: `Mitgliedsnummer,ISBN,Reservierungsdatum`
- Beim Laden der Bücher und der Reservierungen sollen auch alle Zuordnungen zwischen den Objekten (Mitglied, Buch und Reservierung) wieder hergestellt werden.
- Anschließend muss man sich im Programm anmelden. Dazu soll ein Mitglied seine E-Mailadresse und ein Passwort über die Tastatur eingeben können. Prüfen Sie die eingegebenen Daten über die Methode `boolean login(String email, String password)` in der Klasse `MitgliederVerwaltung`. Stimmen E-Mail des Mitglieds und Passwort mit denen eines geladenen Mitglieds überein, wird der Mitglied in der

---

<sup>1</sup>AWF ist die Abkürzung für Anwendungsfall. Dies ist ein Fachbegriff aus der Objektorientierten Analyse und wird in der Veranstaltung OOAD des 3. Fachsemesters detailliert erläutert.



Klassenvariable `Mitglied` `Mitglied` der Klasse `Test` gespeichert und der AWF 2 wird ausgeführt.

- Wird eine falsche Kombination aus E-Mail des Mitglieds und Passwort eingegeben, soll ein entsprechender Hinweis angezeigt werden und das Programm wird beendet.

### **AWF 2: Menü anzeigen**

Erstellen Sie eine Test-Klasse bei deren Start ein geeignetes Menü mit folgenden Auswahlmöglichkeiten erscheint:

- a) Buch reservieren
- b) Meine Reservierungen anzeigen
- c) Reservierung stornieren
- d) Speichern
- e) Abmelden

Das Mitglied soll dann einen Menüpunkt über die Tastatur auswählen können und das Programm soll die dahinter liegenden Anwendungsfälle ausführen. Z.B. wählt das Mitglied den Menüpunkt b), werden dem Mitglied alle sein Reservierungen angezeigt. Danach erscheint wieder das obige Menü.

### **AWF 3: Reservieren eines Buches**

Über den Menüpunkt a) kann ein angemeldetes Mitglied (Klassenvariable `Mitglied` `mitglied` der Klasse `Test`) ein verfügbares Buch reservieren. Dazu sollen zunächst alle nicht reservierten/ausgeliehenen Bücher angezeigt werden. Das Mitglied wählt dann ein Buch über die Eingabe der BuchID aus, um das Buch zu reservieren. Eine Reservierung erfolgt standardmäßig für drei Tage ab dem Datum der Reservierung. Dabei wird die Reservierung in der Collection der Reservierungsverwaltung angelegt. Zusätzlich muss die Anzahl der verfügbaren Exemplare des gerade reservierten Buches um ein Exemplar reduziert werden. Anschließend wird wieder das Hauptmenü angezeigt.

### **AWF 4: Anzeigen meiner Reservierungen**

Über den Menüpunkt b) sollen die reservierten Buchungen des angemeldeten Mitglieds (Klassenvariable `Mitglied` `mitglied` der Klasse `Test`) angezeigt werden. Zeigen Sie dazu alle relevanten Informationen zu den reservierten Büchern des angemeldeten Mitglieds an. Anschließend wird wieder das Hauptmenü angezeigt.



### **AWF 5: Stornieren einer Reservierung**

Über den Menüpunkt c) kann das angemeldete Mitglied die eigenen reservierten Bücher wieder stornieren. Analog zu AWF 4 werden dazu zunächst die reservierten Bücher des Mitglieds angezeigt. Anschließend kann das Mitglied durch Eingabe einer Nummer ein reserviertes Buch stornieren. Dazu soll die Reservierung in der Collection der Reservierungsverwaltung gelöscht werden. Zusätzlich muss die Anzahl der verfügbaren Exemplare des entsprechenden Buches wieder um ein Exemplar erhöht werden. Anschließend wird wieder das Hauptmenü angezeigt.

### **AWF 6: Speichern**

Über den Menüpunkt d) kann das aktuell angemeldete Mitglied zunächst auswählen, ob (1) nur die Reservierungen des angemeldeten Mitglieds gespeichert werden sollen oder (2) alle Reservierungen aller Mitglieder.

Für (1) soll eine Funktion in der Reservierungsverwaltung erstellt werden, die die Reservierungsobjekte des aktuellen Mitgliedes in einer Binärdatei speichert.

Für (2) soll eine weitere Funktion in der Reservierungsverwaltung erstellt werden, die die Reservierungen aller Mitglieder in einer CSV-Datei analog in der Datei „reservierungen.csv“ speichert.

### **AWF 7: Abmelden**

Über den Menüpunkt e) kann das aktuell angemeldete Mitglied sich abmelden. Im Anschluss soll wieder der Anmeldedialog angezeigt werden, in dem ein Mitglied seine E-Mailadresse und ein Passwort über die Tastatur eingeben kann.

### **AWF 8: Unterscheiden zwischen Reservierung und Ausleihe**

Bisher unterscheidet Ihre Implementierung noch nicht zwischen Reservierung und Ausleihe. D.h. die Bücherei möchte wissen, ob ein Buch nur reserviert oder tatsächlich ausgeliehen ist. Überlegen Sie sich dazu eine Lösung und beschreiben Sie Ihren Lösungsansatz in Form eines Kommentars in der Startklasse (Test.java) Ihre Implementierung. Zusätzlich sollen im **AWF 4: Anzeigen meiner Reservierungen** beim Anzeigen der eigenen Reservierungen nur die Reservierungen angezeigt werden und nicht ausgeliehene Bücher.

Viel Erfolg und hoffentlich auch ein wenig Spaß!



**Eidesstattliche Erklärung**

Hiermit versichere ich, dass ich die Hausarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Unterschrift der Verfasserin / des Verfassers