

Elias Kuthe  
Roman Kalkreuth  
Bernd Zey

Dortmund, den 6. Dezember 2019

# Praktikum zur Vorlesung Einführung in die Programmierung WS 19/20

## Blatt 8

Es können 25 Punkte erreicht werden.

### Allgemeine Hinweise

1. Bitte lesen Sie vor der Bearbeitung **alle** Aufgaben sorgfältig durch! Dies erspart Ihnen unnötige Arbeit und somit auch Zeit!
2. Die einzigen Header, die Sie zur Bearbeitung der Aufgaben verwenden dürfen, sind `iostream` und solche, die laut Aufgabenstellung explizit erlaubt werden.
3. Lassen Sie sich fertiggestellte Aufgaben bitte möglichst **frühzeitig** testen. In der letzten halben Stunde vor Schluss werden bei diesem Blatt nur noch **drei** Teil-Aufgaben testiert!
4. Wir akzeptieren ein Testat nur, wenn die Lösung eigenständig auf Anhieb erklärt werden kann. Andernfalls müssen wir die entsprechende Teilaufgabe mit 0 Punkten bewerten.

### Grundlage: Include-Guards

Um zu vermeiden, dass Headerdateien mehrfach inkludiert werden, ist es sinnvoll, sogenannte *Include-Guards* zu verwenden (siehe <https://de.wikipedia.org/wiki/Include-Guard>). Beim Erstellen einer Header-Datei (`.h`-Datei) mittels `File` → `New` → `Header-File` oder beim Erstellen einer Klasse mittels `File` → `New` → `Class` werden sie automatisch hinzugefügt.

### Hinweis: Strings

Verwenden Sie für Zeichenketten die Klasse `std::string` der Standardbibliothek. Dafür müssen Sie den Header `string` inkludieren (siehe Kapitel 9, Folie 75 ff.).

### Aufgabe 1: Buch (10 Punkte)

In dieser Aufgabe soll eine einfache Klasse für Bücher in einer Bibliothek implementiert werden. Legen Sie dazu ein neues Projekt `Aufgabe_8_1` mit der Quelltextdatei `buch.cpp` und einer zugehörigen Headerdatei `buch.h` an.

#### EINSCHRÄNKUNG:

- Alle Methoden sollen in der separaten Quelltextdatei (hier `buch.cpp`) und nicht in der Headerdatei definiert werden. Generell darf in den Dateien, die zu einer Klasse gehören (Headerdatei und Quelldatei mit der Implementierung der Klasse) **keine** `main`-Funktion existieren.
- **Halten Sie sich genau an die genannten Namen für Klassen und Methoden, damit ihr Quelltext problemlos mit der bereitgestellten Testumgebung für Aufgabe 2 zusammenarbeitet!**

a) Definieren Sie in der Headerdatei eine Klasse `Buch`, welche den Autor, den Titel, die Inventarnummer und die Benutzernummer eines möglichen Ausleihers als *private* Attribute enthält. Die Inventarnummer und die Benutzernummer sind vorzeichenlose Ganzzahlen (Typ `unsigned int`), Autor und Titel sind Zeichenketten.

\_\_\_\_\_ (1)

b) Implementieren Sie einen Konstruktor für die Klasse `Buch`, mit dem die drei Attribute für Autor, Titel und Inventarnummer initialisiert werden können. Das Attribut für die Benutzernummer soll mit dem Wert 0 initialisiert werden. Dies bedeutet, dass das Buch an niemanden verliehen, also verfügbar, ist.

HINWEIS: Sie brauchen im Konstruktor keine Anstrengungen zu unternehmen, eindeutige Inventarnummern zu garantieren – dies wird von der Bibliothek in Aufgabe 2 sicher gestellt.

\_\_\_\_\_ (1)

c) Implementieren Sie die folgenden öffentlichen Methoden:

- Eine Methode `bool istVerliehen()`, die dann `true` zurückgibt, wenn das Buch gerade verliehen ist (also Benutzernummer  $\neq 0$ ), andernfalls `false`.
- Eine Methode `unsigned int inventarNummer()`, um die Inventarnummer auszulesen.
- Eine Methode `unsigned int ausgeliehenVon()`, um das Attribut für die Benutzernummer der Person auszulesen, an die das Buch möglicherweise verliehen ist. Das bedeutet auch, dass die Methode den Wert 0 zurückgibt, wenn das Buch gar nicht ausgeliehen ist.
- Eine Methode `bool verleiheAn(unsigned int benutzernummer)`, um zu vermerken, dass das Buch an den angegebenen Benutzer mit der Benutzernummer ausgeliehen wurde. Falls das Buch bereits verliehen ist, soll `false` zurückgeben und der Zustand des Objektes nicht verändert werden. Andernfalls soll die Benutzernummer des Ausleihers im Buch gespeichert und `true` zurückgegeben werden.
- Eine Methode `void rueckgabe()`, um zu vermerken, dass das Buch zurückgegeben wurde.
- Eine Methode `void print()`, die alle Attribute für Menschen lesbar auf der Konsole ausgibt. Orientieren Sie sich an der folgenden Form: `Buch - Inventarnummer 23551: Programmieren macht Spass von H. Mustermann, verfügbar`, bzw. wenn es ausgeliehen ist in der Form `Buch - Inventarnummer 23551: Programmieren macht Spass von H. Mustermann, ausgeliehen von Benutzer 112358`.

\_\_\_\_\_ (6)

d) Fügen Sie Ihrem Projekt eine Datei `buchTest.cpp` mit der `main`-Methode hinzu und inkludieren Sie die `buch.h`-Datei. Erstellen Sie in der `main`-Methode ein Buch mit einem beliebigen Autor, Titel und einer beliebigen Inventarnummer. Geben Sie die Daten zu dem Buch mit Hilfe von `print` aus. Verleihen Sie nun das Buch und geben Sie die Daten abermals aus.

\_\_\_\_\_ (2)

## Aufgabe 2: Bibliothek (15 Punkte)

In dieser Aufgabe soll eine Klasse implementiert werden, die eine Bibliothek repräsentiert. Ein Mitarbeiter hatte bereits damit angefangen, ist aber leider nicht mehr fertig geworden. Auf der Praktikumsseite sind folgende Elemente in der Datei `blatt08.zip` bereitgestellt:

- die Klasse `Bibliothek` (`bibliothek.h` und `bibliothek.cpp`), in der einige Methoden noch nicht fertig implementiert wurden
- die Klasse `Benutzer` (`benutzer.h` und `benutzer.cpp`)
- eine Implementierung des ADT `Liste` (`eidpliste.h`)
- eine Testumgebung (`testumgebung.cpp`, `bibmanager.h` und `bibmanager.cpp`).

a) Legen Sie ein neues Projekt `Aufgabe_8_2` an, laden Sie sich die Datei `blatt08.zip` herunter, entpacken Sie diese und fügen Sie die oben genannten Elemente zu dem Projekt hinzu. Kopieren Sie außerdem die Dateien `buch.h` und `buch.cpp` aus der ersten Aufgabe und fügen Sie sie dem Projekt hinzu. Stellen Sie sicher, dass sich der gesamte Quelltext übersetzen lässt.

\_\_\_\_\_ (2)

b) Starten Sie das Programm. Zu Beginn sehen Sie die Fehler-Ausgabe der `erfasse`-Methode: `erfasse()` - Leider noch nicht implementiert. Anschließend werden die unterstützten Aktionen beim Zugriff auf die Bibliothek angezeigt. Testen Sie das System indem Sie jede Aktion einmal ausführen.

\_\_\_\_\_ (1)

c) Machen Sie sich mit der Klasse `Bibliothek` vertraut. Welche Attribute und welche Methoden enthält die Klasse? Was fällt Ihnen noch auf?

---

---

---

---

---

\_\_\_\_\_ (2)

d) Leider wurde die Methode `void erfasse(std::string autor, std::string titel)` der `Bibliothek`-Klasse nicht fertig implementiert. Hiermit soll ein neu angeschafftes Buch erfasst und zur Liste der Bücher hinzugefügt werden. Implementieren Sie die Methode und stellen Sie dabei insbesondere sicher, dass ein neues Buch eine eindeutige Inventarnummer erhält!

Sie können sich bei Ihrem Vorgehen an der Methode `void anmelden(std::string name)` orientieren.

Testen Sie Ihre Implementierung mit der bereitgestellten Testumgebung und lassen Sie sich die Liste aller Bücher anzeigen.

\_\_\_\_\_ (2)

e) Fügen Sie der Klasse `Bibliothek` eine private Methode `Buch* findeBuch(unsigned int inventarnummer)` hinzu und implementieren Sie diese Methode. In `findeBuch` soll überprüft werden, ob das Buch mit der übergebenen Inventarnummer existiert. Wenn das Buch existiert, soll ein Zeiger auf das Buch zurück gegeben werden, ansonsten soll `nullptr` zurück gegeben werden. Um das Buch zu suchen muss die Liste der Bücher (`_buecher`) durchlaufen und für jedes Buch die Inventarnummer verglichen werden.

Zur Orientierung können Sie in der Methode `listeBuecherAuf()` sehen, wie die Liste der Bücher durchlaufen wird.

\_\_\_\_\_ (2)

f) Implementieren Sie die Methode `bool rueckgabe(unsigned int inventarnummer)`, mit der vermerkt werden kann, dass ein Benutzer ein Buch zurückgegeben hat. Die Methode bekommt als Parameter die Inventarnummer des zurückgegebenen Buches. Überprüfen Sie, indem Sie `findeBuch` aufrufen, ob das Buch mit der Inventarnummer existiert. Falls ja, soll bei dem entsprechenden Buch die Methode `rueckgabe` aufgerufen werden und die Methode soll `true` zurück geben. Dies soll unabhängig davon geschehen, ob das Buch überhaupt verliehen war. Wurde die Inventarnummer nicht gefunden, soll `false` zurückgegeben werden.

\_\_\_\_\_ (2)

g) Implementieren Sie die Methode `Resultat ausleihe(unsigned int inventarnummer, unsigned int benutzernummer)`, mit der vermerkt werden kann, dass ein Benutzer ein Buch ausgeliehen hat. Dabei kann es zu verschiedenen Fehlern kommen, die über den Rückgabewert der Methode angezeigt werden.

Prüfen Sie zunächst mit Hilfe von `Benutzer* findeBenutzer(unsigned int benutzernummer)`, ob die übergebene Benutzernummer überhaupt zu einem angemeldeten Benutzer gehört. Ist dies nicht der Fall, soll die Methode den Wert `BENUTZER_NICHT_VORHANDEN` zurückgeben. Die Methode `findeBenutzer` gibt eine Referenz auf den Benutzer zurück, falls er angemeldet ist, und `nullptr` andernfalls.

Prüfen Sie dann mit Hilfe von `findeBuch`, ob die übergebene Inventarnummer zu einem existierenden Buch gehört. Ist dies nicht der Fall, soll die Methode den Wert `BUCH_NICHT_VORHANDEN` zurückgeben.

Andernfalls kann das Buch an den Benutzer ausgeliehen werden. Dies soll im Buch-Objekt mit der passenden Inventarnummer vermerkt werden. Stellt sich dabei heraus, dass das Buch bereits ausgeliehen ist, soll die Methode den Wert `BUCH_AUSGELIEHEN` zurückgeben. Wurde das Buch erfolgreich an das Benutzer ausgeliehen, soll die Methode den Wert `AUSLEIHE_OK` zurückgeben.

\_\_\_\_\_ (3)

h) Testen Sie Ihre Implementierung mit der bereitgestellten Testumgebung.

\_\_\_\_\_ (1)