

Andre Droschinsky  
Roman Kalkreuth  
Denis Kurz  
Bernd Zey

Dortmund, den 13. Dezember 2018

# Praktikum zur Vorlesung Einführung in die Programmierung WS 18/19

## Blatt 8

Es können 22 Punkte erreicht werden.

### 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 Klasse mittels `File` → `New` → `Class` werden sie automatisch hinzugefügt.

### Aufgabe 1: Buch (6 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.
- Verwenden Sie für Zeichenketten die Klasse `std::string` der Standardbibliothek. Dafür müssen Sie den Header `string` inkludieren (siehe Kapitel 9, Folie 75 ff.).
- **Halten Sie sich genau an die genannten Namen für Klassen und Methoden, damit ihr Quelltext problemlos mit der bereitgestellten Testumgebung (siehe Aufgabe 3) zusammenarbeitet!**

a) Definieren Sie in der Headerdatei eine Klasse `Buch`, welche den Autor, den Titel, die Inventarnummer und die Mitgliedsnummer eines möglichen Ausleihers als *private* Attribute enthält. Die Inventarnummer und die Mitgliedsnummer 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 Mitgliedsnummer 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 3 sicher gestellt.

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

c) Implementieren Sie die folgenden öffentlichen Methoden:

- Eine Methode `bool istVerliehen()`, die dann `true` zurückgibt, wenn das Buch gerade verliehen ist, andernfalls `false`.
- Eine Methode `unsigned int inventarNummer()`, um die Inventarnummer auszulesen.
- Eine Methode `unsigned int ausgeliehenVon()`, um das Attribut für die Mitgliedsnummer 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 (siehe Aufgabenteil b)).
- Eine Methode `bool verleiheAn(unsigned int mitglied)`, um zu vermerken, dass das Buch an den angegebenen Benutzer ausgeliehen wurde. Falls das Buch bereits verliehen ist, soll `false` zurückgeben und der Zustand des Objektes nicht verändert werden. Andernfalls soll die Mitgliedsnummer 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, z.B. in der Form: `Buch - Inventarnummer 23551: Programmieren macht Spass von H. Mustermann, verfügbar`.

\_\_\_\_\_ (4)

## Aufgabe 2: Mitglied der Bibliothek (4 Punkte)

In dieser Aufgabe soll eine Klasse für die Mitglieder einer Bibliothek implementiert werden. Fügen Sie dazu eine Headerdatei `mitglied.h` und eine Quelldatei `mitglied.cpp` zu dem Projekt aus der ersten Aufgabe hinzu.

a) Definieren Sie in der Headerdatei eine Klasse `Mitglied`, welche den Namen und die Mitgliedsnummer als *private* Attribute enthält. Die Mitgliedsnummer ist eine vorzeichenlose Ganzzahl (Typ `unsigned int`), der Name ist eine Zeichenkette.

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

b) Implementieren Sie einen Konstruktor für die Klasse `Mitglied`, mit dem die zwei Attribute initialisiert werden können. Die Implementierung soll in der Datei `mitglied.cpp` erfolgen.

HINWEIS: Sie brauchen im Konstruktor keine Anstrengungen zu unternehmen, eindeutige Mitgliedsnummern zu garantieren – dies wird von der Bibliothek in Aufgabe 3 sicher gestellt.

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

c) Implementieren Sie die folgenden öffentlichen Methoden:

- Eine Methode `unsigned int mitgliedsNummer()`, um das Attribut für die Mitgliedsnummer auszulesen.
- Eine Methode `void print()`, die alle Attribute für Menschen lesbar auf der Konsole ausgibt, z.B. in der Form: `Mitglied 176: Bianca Beispiel`.

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

### Aufgabe 3: Bibliothek (12 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 bereitgestellt: Die Klasse `Bibliothek` (`bibliothek.h` und `bibliothek.cpp`), eine Implementierung des ADT `Liste` (`eidpliste.h`) und eine Testumgebung (`testumgebung.cpp`, `bibmanager.h` und `bibmanager.cpp`). Fügen Sie diese Elemente zu dem Projekt aus Aufgabe 1 hinzu.

a) Stellen Sie sicher, dass sich der gesamte Quelltext übersetzen lässt. Machen Sie sich dann mit der Klasse `Bibliothek` vertraut. Welche Attribute und welche Methoden enthält die Klasse? Was fällt Ihnen noch auf?

---

---

---

---

---

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

b) Leider wurde die Methode `void erfasse(std::string autor, std::string titel)` nicht fertig implementiert, mit der ein neu angeschafftes Buch erfasst und zur Liste der Bücher hinzugefügt werden kann. 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.

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

c) Implementieren Sie die Methode `bool rueckgabe(unsigned int buchnr)`, mit der vermerkt werden kann, dass ein Benutzer ein Buch zurückgegeben hat. Die Methode bekommt als Parameter die Inventarnummer des zurückgegebenen Buches. Überprüfen Sie in der Methode, ob diese Nummer tatsächlich vorhanden ist. Falls ja, soll bei dem entsprechenden Buch der Status auf nicht ausgeliehen gesetzt werden und die Methode `true` zurückgeben. Dies soll unabhängig davon geschehen, ob das Buch überhaupt verliehen war. Wurde die Inventarnummer nicht gefunden, soll `false` zurückgegeben werden.

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

d) Implementieren Sie die Methode `Resultat ausleihe(unsigned int buch, unsigned int mitglied)`, mit der vermerkt werden kann, dass ein Mitglied ein Buch ausgeliehen hat. Dabei kann es zu verschiedenen Fehlern kommen, die über den Rückgabewert der Methode angezeigt werden. Prüfen Sie zunächst, ob die übergebene Mitgliedsnummer überhaupt zu einem existierenden Mitglied gehört. Ist dies nicht der Fall, soll die Methode den Wert `MITGLIED_NICHT_VORHANDEN` zurückgeben. Prüfen Sie dann, ob die übergebene Inventarnummer zu einem existierenden Buch gehört. Ist dies nicht der Fall, soll die Methode den Wert `BUCH_NICHT_VORHANDEN` zurückgeben. Außerdem darf ein Benutzer nur maximal drei Bücher gleichzeitig ausleihen. Prüfen Sie also, ob bereits drei Bücher von diesem Benutzer ausgeliehen sind. Ist dies der Fall, soll die Methode den Wert `AUSLEIH_LIMIT_ERREICHT` 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 Mitglied ausgeliehen, soll die Methode den Wert `AUSLEIHE_OK` zurückgeben.

HINWEIS: Orientieren Sie sich beim Zugriff auf die Liste von Büchern an der Methode `void listeBuecherAuf()` (analog für die Liste der Mitglieder).

\_\_\_\_\_ (5)

e) Testen Sie Ihre Implementierung mit der bereitgestellten Testumgebung.

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