Übungsblatt 10

Ralph Gauges Ursula Rost Katja Wegner

16.01.2008

Inhalt dieser Aufgaben ist es, eine kleine Bücherei mit einer elektronischen Verwaltung von Büchern zu versehen, so daß die Bücher im Bestand

- abgerufen werden und
- ihr Status (ausgeliehen / nicht ausgeliehen) abgefragt und verändert werden kann.

Aufgabe 1:

Schreiben Sie zunächst eine Klasse *Book*, die folgende Attribute besitzt:

- Nachname des Autors (Zeichenkette)
- Vorname des Autors (Zeichenkette)
- Titel (Zeichenkette)
- eine (eindeutige) ID für das Buch (Zeichenkette)
- ein Flag, das anzeigt, ob dieses Buch gerade ausgeliehen ist oder nicht
- eine Liste der Ausleiher (bei der die ID des aktuellen Entleihers immer am Ende eingefügt werden soll (alte Einträge werden nicht gelöscht)

Verwenden Sie für Zeichenketten die *std::string* Klasse der C++ Strings library. Für die IDs sollen ebenfalls Zeichenketten verwendet werden.

Verwenden Sie für die Listenstruktur die Container Klasse *std::list*.

Die Klasse *Book* sollte ausserdem zwei Konstruktoren haben:

- einen leeren Konstruktor, der den Attributen Defaultwerte zuweist (z.B. leere Strings und eine ID, die von einem Menschen als ungültig interpretierbar ist)
- einen Konstruktor, der die Parameter Nachname, Vorname, Titel und ID verarbeitet

Implementieren Sie dazu folgende Funktionalität:

- einen Ausgabeoperator, der die in der Klasse gespeicherte Informationen in einer formatierten Weise ausgibt
- `get`-Methoden für
 - den Nachnamen des Autors,
 - die ID des Buchs
 - für den aktuellen Entleiher des Buches
 - ob das Buch ausgeliehen ist und
- `set`-Methoden für
 - ob das Buch aktuell entliehen ist
 - den aktuellen Entleiher (am Ende der Liste einfügen)

Hinweis: Das Ergebnis für die Abfrage des aktuellen Entleihers ist abhängig davon, ob das Buch ausgeliehen ist oder nicht (testen Sie die entsprechenden Bedingungen in der Methode ab und geben Sie den jeweils entsprechende Wert zurück).

Aufgabe 2:

Schreiben Sie nun eine Klasse *Library*, die die vorhandenen Bücher verwaltet:

- Bücher sollen in die Bücherei eingefügt und dann ausgeliehen bzw. zurückgegeben werden können
- Es soll Suchfunktionen geben, die in Abhängigkeit von einer vorgegebenen (eindeutigen) ID eines Buchs oder eines (nicht eindeutigen) Nachnamens eines Autors ein Buch in der Bücherei suchen.

Versehen Sie daher die Klasse *Library* zunächst mit zwei Attributen:

- `idMap`: eine Map, die IDs von Büchern auf Buch-Objekte abbildet
- `nameMap`: eine Multimap, die den Nachnamen eines Autors auf die ID des zugehörigen Buches abbildet.

Hinweise: Für das Attribut `idMap` verwenden Sie die STL-Container Klasse `std::map`, bei der es jeweils nur einen Eintrag für jeden verwendeten Schlüssel gibt.

Für das Attribut `nameMap` verwenden die die STL-Container Klasse `std::multimap`, bei der es zu jedem Schlüssel mehrere mögliche Einträge gibt

Implementieren Sie nun die Funktionalität der *Library*-Klasse durch folgende Methoden:

- `insertBook`: falls das betreffende Buch noch nicht in der Bücherei vorhanden ist (abprüfen !), wird es eingetragen, indem es über seine ID in die `idMap` eingetragen wird und über den Nachnamen des Autors der `nameMap` hinzugefügt wird (jeweils zusammen mit dem jeweiligen Schlüssel). Falls ein Eintrag erfolgte, soll die Methode `true` zurückgeben, sonst `false` (Parameter: ein Buch)
- `deleteBook`: falls das betreffende Buch in der Bücherei vorhanden ist, soll es hier gelöscht werden (d.h. die zugehörigen Einträge in den maps sollen gelöscht werden) und die Methode soll `true` zurückgeben, ansonsten `false` (Parameter: die ID eines Buchs)
- `findBook` soll anhand einer ID überprüfen, ob das betreffende Buch in der Bücherei eingetragen ist (egal, ob ausgeliehen oder nicht), falls das Buch gefunden wurde, soll ein Pointer auf dieses Buch zurückgegeben werden, ansonsten ein null-Pointer (Parameter: die ID eines Buchs)
- `findBooks`: soll über einen beliebigen angegebenen Namen eine Liste (STL Klasse `std::list` von Pointern auf Bücher erzeugen, die von einem Autor dieses Namens geschrieben wurden. (Parameter: Nachname eines potenziellen Autors) **Hinweis:** Sie müssen erst in der `nameMap` suchen und dann mit allen Ergebnissen die ID nehmen und in der `idMap` suchen.

- **lendBook**: falls das betreffende Buch in der Bücherei eingetragen ist und nicht ausgeliehen ist, soll es als ausgeliehen markiert werden und der Ausleiher eingetragen werden. Falls der Vorgang erfolgreich war, soll die Methode `true` zurückgeben, sonst `false`. (Parameter: die ID eines Ausleihers und die ID eines Buchs)
- **returnBook**: falls das betreffende Buch in der Bücherei eingetragen ist und ausgeliehen ist, soll es als nicht ausgeliehen markiert werden. Falls der Vorgang erfolgreich war, soll die Methode `true` zurückgeben, sonst `false`. (Parameter: die ID eines Buchs)
- **printAllBooks** soll eine formatierte Liste aller in der Bücherei eingetragenen Bücher produzieren
- **printBookList** soll eine formatierte Liste aller in einer als Parameter übergebenen Liste (STL Klasse `std::list`) von Pointern auf Bücher (also Pointer auf Book Objekte) ausgeben.
- **showLenderOfBook** falls das Buch in der Bücherei vorhanden und den Status ausgeliehen hat, soll der zugehörige Ausleiher ausgegeben und als Rückgabewert zurückgegeben werden (Parameter: die ID eines Buchs)

Testen Sie die implementierte Funktionalität, indem Sie in einer `main`-Routine folgendes tun (durch entsprechende Aufrufe der Methoden)

1. Erzeugen Sie ein `Library` Objekt.
2. Erzeugen Sie mehrere Buch-Objekte (mit teilweise sich wiederholenden Autoren Nachnamen)
3. Fügen Sie die erzeugten Objekt in die Bücherei ein
4. Erzeugen Sie eine Liste von Pointern auf `Book` und initialisieren Sie sie durch Aufruf der Methode `findBooks` mit einem Namen als Parameter, der als Autor mehrfach vorkommt.
5. Geben Sie die Liste aus
6. Erzeugen Sie zwei Ausleiher-IDs, leihen Sie damit für einen der beiden Ausleiher das Buch aus, versuchen Sie es dann, für den Anderen auszuliehen, lassen Sie das Buch dann zurückgeben und versuchen erneut, es

für den zweiten Ausleiher auszuleihen. Lassen Sie sich zwischendurch jeweils den aktuellen Ausleiher des Buchs ausgeben.

7. Löschen Sie ein Buch aus der Bibliothek
8. Suchen Sie nach einem Buch und lassen Sie sich das Ergebnis ausgeben.
9. Überprüfen Sie jeweils, ob die Methoden das Gewünschte getan haben und ergänzen Sie ggf. die Ausgaben der Methoden, um jeweils eine passende Rückmeldung zu erhalten.