

Lập trình hướng đối tượng

Kết tập và kế thừa

Thông tin:

Họ và tên:.....

Mã số sinh viên:.....

* * * * *

Yêu cầu bài tập: Xây dựng một chương trình đơn giản giúp bác tổ trưởng tổ dân phố quản lý thông tin các cuốn sách trong một thư viện (tủ sách nhỏ) của tổ dân phố 7, phường Bách Khoa. Chương trình có giao diện dòng lệnh (console application), khi người dùng muốn thực hiện một chức năng thì sẽ nhập vào số thứ tự của mục chức năng tương ứng ☐ nhấn Enter và chương trình sẽ thực thi.

Các chức năng chính của chương trình gồm có:

1. Xem thông tin của thư viện ☐ hiển thị thông tin thư viện như: tên thư viện, số đầu sách hiện có trong thư viện và danh sách tiêu đề các cuốn sách
2. Thêm đầu sách mới ☐ thêm một đầu sách mới vào thư viện
3. Tìm kiếm sách ☐ tìm kiếm thông tin đầu sách trong thư viện theo ID đầu sách
4. Mượn một cuốn sách ☐ khi có người đến mượn một cuốn sách, bác tổ trưởng sẽ dùng chức năng này để cập nhật số lượng sách còn lại của đầu sách tương ứng
5. Trả một cuốn sách ☐ khi có người đến trả một cuốn sách, chức năng này giúp cập nhật số lượng sách của đầu sách trong thư viện
6. Kết thúc và thoát chương trình

Giao diện chương trình có dạng như sau:

```
===== Main menu =====
1. Show library information
2. Add new book
3. Find book
4. Borrow a book
5. Return a book
6. Exit
=====
Enter menu ID (1-6):
```


1. Mô hình hoá yêu cầu bài toán qua biểu đồ usecase

Trả lời:

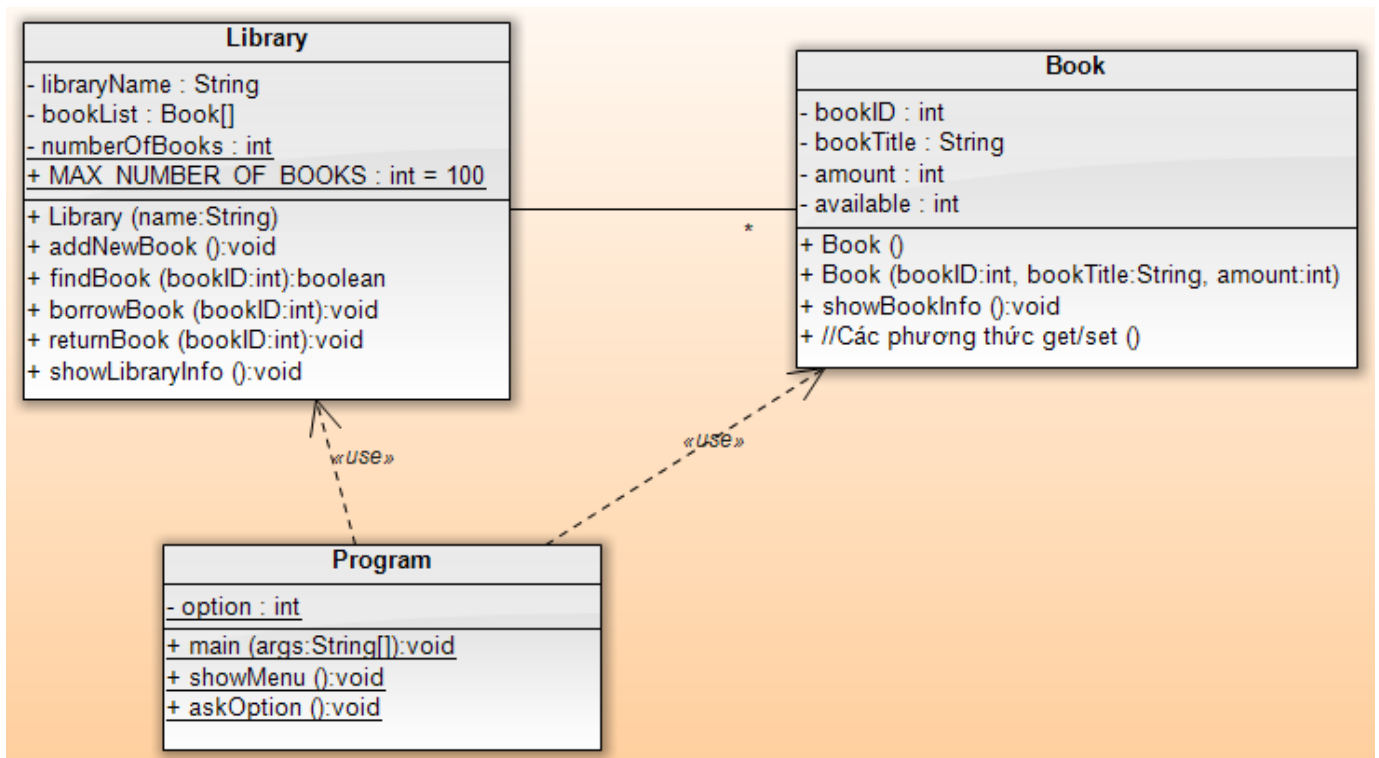
- Biểu đồ usecase:

2. Xây dựng mã nguồn chương trình quản lý thư viện đơn giản

Phân tích: để giải quyết bài toán này chúng ta xây dựng 3 lớp

- Lớp **Book** đóng gói các thông tin về đầu sách trong thư viện, mỗi đối tượng của lớp Book là một đầu sách trong thư viện.
- Lớp **Library** chứa các thông tin chung về thư viện như: tên thư viện,... và danh sách các đầu sách mà người dân trong tổ dân phố có thể mượn trong thư viện, để thể hiện điều này, đơn giản là chúng ta xây dựng một mảng (array) các đối tượng Book trong lớp Library. Các chức năng chính (thêm sách, tìm sách, mượn sách, trả sách) sẽ được thực hiện trong lớp Library này.
- Lớp **Program** chứa phương thức **main()** để thực thi chương trình. Lớp Program cũng chịu trách nhiệm giao tiếp với người sử dụng chương trình (hiển thị menu chương trình, đọc lựa chọn người dùng nhập vào, thực thi chức năng qua lời gọi phương thức của đối tượng Library).

Thiết kế: sơ đồ lớp



Chi tiết từng lớp:

Lớp **Book**: đóng gói các thông tin về đầu sách trong thư viện, mỗi đối tượng của lớp Book là một đầu sách trong thư viện.

Book
- bookID : int - bookTitle : String - amount : int - available : int
+ Book () + Book (bookID:int, bookTitle:String, amount:int) + showBookInfo ():void + //Các phương thức get/set ()

+ Mỗi đầu sách gồm các thông tin: mã đầu sách (bookID), tiêu đề (bookTitle), số lượng các cuốn sách của đầu sách này (amount), số lượng các cuốn sách hiện còn lại trong thư viện (available). Ban đầu thì available = amount, khi có người mượn sách của đầu sách tương ứng thì available giảm đi 1 thể hiện có 1 cuốn sách được mượn. Ngược lại khi trả sách thì available sẽ tăng thêm 1.

+ Các thuộc tính được thiết kế với giới hạn truy cập private, viết các phương thức public get/set cho các thuộc tính này (chú ý kiểm tra giá trị hợp lý cho các thuộc tính)

+ Lớp Book có hai phương thức khởi tạo:

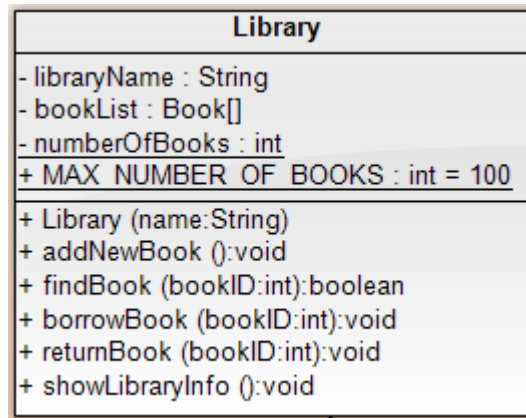
- Constructor không tham số, khởi tạo giá trị mặc định cho các thuộc tính
- Constructor với 3 tham số cung cấp thông tin chi tiết cho đối tượng đầu sách mới được tạo ra (**khi mới tạo thì available = amount**)

+ Phương thức **showBookInfo()** hiển thị thông tin của cuốn sách ra màn hình như sau:

```
Book ID: ...  
Book Title: ...  
Total Amount: ...  
Available: ...
```

Trả lời Mã nguồn lớp Book:

Lớp **Library** chứa các thông tin chung về thư viện như: tên thư viện,... và các chức năng xử lý chính:



+ Một thư viện có các thông tin: tên thư viện (libraryName), danh sách các đầu sách của thư viện (mảng bookList: Book[]). Thông tin về số lượng đầu sách hiện có trong thư viện được lưu trong thuộc tính static (numberOfBooks). Do thư viện trong bài toán của chúng ta là một thư viện nhỏ (tủ sách nhỏ) nên số lượng đầu sách có giới hạn, không quá 100. Thông tin giới hạn này được ghi nhận trong một hằng static (MAX_NUMBER_OF_BOOKS), nếu số lượng đầu sách hiện tại đã đến giới hạn thì không thể thêm mới đầu sách vào thư viện.

```
private static int nbBooks;  
public static final int MAX_NUMBER_BOOKS = 100;
```

+ Xây dựng phương thức khởi tạo cho lớp Library.

+ Phương thức **addNewBook()**: phương thức này cho phép người dùng nhập các thông tin mới của một đầu sách từ bàn phím và thêm nó vào danh sách đầu sách của thư viện.

Ví dụ khi người dùng chọn chức năng 2 trong menu chương trình:

```
2. Add new book  
-----  
Enter book's information:  
(1)Book ID: 1  
(2)Book Title: Big Java 8 (2017)  
(3)Amount: 20
```

Chú ý: trước khi thêm một đầu sách mới cần kiểm tra xem số lượng đầu sách trong thư viện đã quá giới hạn chưa. Và kiểm tra để các bookID không được trùng nhau.

+ Phương thức **findBook(bookID)**: tìm kiếm thông tin đầu sách trong thư viện, trong bài toán đơn giản này thì sẽ tìm kiếm theo bookID

Nếu tìm thấy:

Return true, In ra thông tin đầu sách tìm thấy

Ngược lại:

Return false.

Ví dụ khi người dùng chọn chức năng 3 trong menu chương trình: người dùng sẽ được yêu cầu nhập vào BookID cần tìm, giá trị ID truyền vào cho lời gọi **findBook** để thực hiện tìm kiếm.

```
3. Find book
-----
Enter book's ID: 1
Book title: Big Java 8 (2017)
BookID:      1
Total amount: 20
Left amount: 20
```

+ Phương thức **borrowBook(bookID)**: xử lý mượn sách

Kiểm tra nếu đầu sách có thể cho mượn (giá trị available>0) không

Nếu có thể thì thông báo cho mượn thành công và cập nhật số lượng còn lại của đầu sách đó

Hiển thị lại thông tin sau khi cập nhật.

Ví dụ khi người dùng chọn chức năng 4 trong menu chương trình: người dùng sẽ được yêu cầu nhập vào BookID để mượn sách, giá trị ID truyền vào cho lời gọi **borrowBook** để xử lý mượn sách.

```
4. Borrow a book
-----
Enter book's ID: 1
Book title:      Big Java 8 (2017)
BookID:          1
Total amount:    20
Left amount:     20

The book is borrowed successfully!
Book title:      Big Java 8 (2017)
BookID:          1
Total amount:    20
Left amount:     19
```

+ Phương thức **returnBook(bookID)**: xử lý trả sách

Cập nhật số lượng còn lại của đầu sách đó

Hiển thị lại thông tin sau khi cập nhật.

Ví dụ khi người dùng chọn chức năng 5 trong menu chương trình: người dùng sẽ được yêu cầu nhập vào BookID để trả sách, giá trị ID truyền vào cho lời gọi **returnBook** để xử lý trả sách.

```
5. Return a book
-----
Enter book's ID: 1

The book is returned successfully!
Book title:      Big Java 8 (2017)
BookID:          1
Total amount:    20
Left amount:     20
```


+ Phương thức **showLibraryInfo()** hiển thị thông tin chi tiết của thư viện: tên thư viện, số lượng đầu sách hiện có và danh sách các đầu sách trong thư viện.

Trả lời Mã nguồn lớp **Library**:

Lớp **Program** chứa phương thức **main()** để thực thi chương trình.

Program
- <u>option : int</u>
+ <u>main (args:String[]):void</u>
+ <u>showMenu ():void</u>
+ <u>askOption ():void</u>

+ Các thuộc tính và phương thức của lớp Program đều được khai báo static để gọi và sử dụng trong phương thức **main()**.

+ Thuộc tính **option** chứa thứ tự chức năng người dùng lựa chọn.

+ Phương thức **showMenu()** hiển thị menu chính của chương trình:

```
===== Main menu =====
1. Show library information
2. Add new book
3. Find book
4. Borrow a book
5. Return a book
6. Exit
=====
Enter menu ID (1-6):
```

+ Phương thức **askOption()** đọc lựa chọn của người dùng và lưu vào **option**.

Trả lời Mã nguồn lớp **Program**:

Xây dựng thêm các phương thức cho lớp Library

+ Xây dựng phương thức nạp chồng: **findBook(String keyword): boolean**

- Tìm kiếm đầu sách theo từ khoá trong tiêu đề
- Đầu vào của phương thức là xâu ký tự chứa các từ khoá tìm kiếm
- Thực hiện xử lý tìm kiếm các đầu sách mà tiêu đề có chứa từ khoá tìm kiếm trên
 - + Nếu tìm thấy: trả về true, In ra số lượng đầu sách tìm thấy và danh sách đầu sách tương ứng
 - + Nếu không tìm thấy, trả về false

Chú ý: ví dụ keyword tìm kiếm là “Harry Potter” thì các đầu sách chứa một trong hai từ “Harry” hoặc “Potter” hoặc chứa cả hai thì đều thoả mãn.

(Việc tìm chính xác ký tự hoặc tìm mà không phân biệt ký tự viết hoa, viết thường □ sinh viên tuỳ ý lựa chọn 1 trong 2 cách)

Trả lời Mã nguồn phương thức **findBook(String keyword): boolean**

+ Xây dựng phương thức: **getALuckyBook(): Book**

- Sắp đến ngày lễ 30/04 – 01/05 bác tổ trưởng quyết định tặng cho mỗi bạn nhỏ đến tủ sách khu phố một quyển sách bất kỳ. Phương thức **getALuckyBook()** giúp bác tổ trưởng chọn ngẫu nhiên một trong các đầu sách của thư viện để tặng, sau đó cập nhật số lượng còn lại của đầu sách đó.

Trả lời Mã nguồn phương thức **getALuckyBook(): Book**

Chụp **ảnh các màn hình** kết quả chạy chương trình và chèn vào đây:

+ Yêu cầu chức năng mới:

- Thư viện được mở rộng cần quản lý thêm thông tin của các cuốn tạp chí
- Mỗi đầu tạp chí chỉ có một bản duy nhất với các thông tin:
 - o Mã tạp chí (magID),
 - o Tiêu đề (magTitle),
 - o Số phát hành (magIssue),
 - o Trạng thái mượn (status),
 - o Giá tiền (price)
- Hãy tái cấu trúc (refactoring) sơ đồ thiết kế lớp để đáp ứng yêu cầu chức năng mới này, biết rằng tương lai sẽ có thêm các đối tượng mới mà thư viện cần quản lý thông tin (ví dụ như: CompactDisc, DigitalVideoDisc,...).

Trả lời: Sơ đồ lớp sau khi refactoring

./HẾT/.