

Bài tập 1: Lập trình hướng đối tượng

Hệ thống quản lý thư viện

QUAN TRỌNG: Nhiệm vụ này phải được thực hiện riêng lẻ

Đọc **Phần 1** để hiểu các yêu cầu lập trình, **Phần 2** để hiểu các nhiệm vụ lập trình mà bạn cần thực hiện và **Phần 3** để biết cách gửi tác phẩm của bạn.

1. Mô tả

Trong bài tập này, bạn sẽ tạo một chương trình Java có tên **LibMancho** một hệ thống quản lý thư viện đơn giản. Mục đích của chương trình này là cho phép khách hàng của thư viện kiểm tra (mượn) sách, thủ thư quản lý bộ sưu tập của thư viện và tính tiền phạt cho những cuốn sách quá hạn. Bạn được cung cấp một **chung** gói chứa ba lớp (**Ngày sử dụng**, **thể loại** và **Loại người bảo trợ**) mà bạn sẽ sử dụng trong bài tập này. Vui lòng không sửa đổi các lớp được cung cấp.

Người bảo trợ thư viện có các thuộc tính sau:

- ID người bảo trợ (mã định danh duy nhất được hệ thống tự động tạo)
- Tên
- Ngày sinh (DOB)
- Email
- Số điện thoại
- Loại khách hàng quen (thông thường, cao cấp)

ID người bảo trợ được hệ thống tạo tự động bằng công thức: chữ cái **P** theo sau là một số duy nhất, tự động tăng từ **1**. Số phải được đệm bằng số 0 để có ít nhất 3 chữ số. Ví dụ: người bảo trợ đầu tiên sẽ có ID người bảo trợ là **P001**, người bảo trợ thứ hai sẽ có **P002**, và như thế. 100 **quần què** người bảo trợ sẽ có ID của **P100**, phần đệm bằng 0 không còn cần thiết nữa vì số này đã có 3 chữ số.

Sách Thư viện có các thuộc tính sau:

- ISBN (Số sách tiêu chuẩn quốc tế, mã định danh duy nhất) Tiêu đề
-
- Tác giả
- thể loại
- Năm xuất bản
- Số lượng bản sao có sẵn

ISBN là mã nhận dạng sách duy nhất được hệ thống tự động tạo ra. Nó kết hợp các yếu tố như tên viết tắt của tác giả, mã số cho thể loại và năm xuất bản. Ví dụ, một cuốn sách của **John Doe** trong năm **2023** với mã thể loại dành cho lập trình (ví dụ: **02**) sẽ có ISBN như **JD-02-2023**.

Sách trong thư viện có thể được phân loại thành nhiều thể loại khác nhau, chẳng hạn như lập trình, tiểu thuyết, phi hư cấu, khoa học viễn tưởng, v.v. Bạn sẽ cần sử dụng sách được cung cấp **chung.Thể loại** lớp enum.

Có hai loại khách hàng quen trong hệ thống thư viện: *thường xuyên* khách hàng quen và *phần thưởng* khách hàng quen. Khách hàng thường xuyên có thể kiểm tra tối đa **3** sách cùng một lúc, trong khi những người bảo trợ cao cấp có thể xem tới **5** sách. Bạn đọc có trách nhiệm trả sách đúng hạn để tránh bị phạt. Bạn được cung cấp các **chung.PatronType** lớp enum. Bạn không nên thực hiện hoặc sửa đổi nó.

Chương trình của bạn nên cho phép khách hàng mượn và trả sách, thủ thư thêm sách mới vào bộ sưu tập, cập nhật thông tin sách và tính tiền phạt đối với sách quá hạn dựa trên các tiêu chí đã xác định.

2. Yêu cầu nhiệm vụ

1. Bạn phải tạo gói cấp cao nhất có tên **a1_sid** cho chương trình nơi **bên** là thẻ sinh viên của bạn.
2. Sao chép được cung cấp **chung.package** dưới dạng gói cấp cao nhất của dự án Java của bạn.
3. Chỉ định và triển khai các lớp Java phù hợp với các yêu cầu của chương trình đã nêu.
4. Chỉ định và thực hiện một **Sách** lớp đại diện cho một cuốn sách trong hệ thống quản lý thư viện. Lớp này phải có những điều sau đây:
(Một) **tạoISBN()** - Tạo ISBN duy nhất cho mỗi cuốn sách.
(b) **getNumCopiesAvailable()** - Theo dõi số lượng bản sao có sẵn.
5. Chỉ định và thực hiện một **người bảo trợ** lớp đại diện cho một người bảo trợ thư viện. Lớp này phải có những điều sau đây:
(Một) **tạoPatronID()** - Tạo một ID duy nhất cho mỗi Người bảo trợ (ví dụ: **P001, P002...**)
6. Các **Thư viện Giao dịch** lớp đại diện cho một giao dịch trong thư viện, liên quan đến một khách hàng quen, một cuốn sách, ngày thanh toán, ngày đến hạn, ngày trả lại và số tiền phạt. Lớp này phải có những điều sau đây:
(a) **Cái tính toánFine()** Phương pháp tính số tiền phạt dựa trên sự chênh lệch giữa ngày trả lại và ngày đến hạn, áp dụng quy tắc phạt như quy định dưới đây.
- Tiền phạt quá hạn được áp dụng cho mỗi cuốn sách được khách hàng kiểm tra, tùy thuộc vào số ngày cuốn sách bị quá hạn. Cách tính phạt như sau:
 - \$1,00 mỗi ngày đối với sách quá hạn từ 1 đến 7 ngày \$2,00
 - mỗi ngày đối với sách quá hạn từ 8 đến 14 ngày \$3,00 mỗi
 - ngày đối với sách quá hạn trên 14 ngày

(b) Việc `getDescription()` phương pháp này tạo ra mô tả chi tiết về giao dịch, bao gồm ID khách hàng, ISBN của sách, ngày thanh toán, ngày đến hạn, ngày trả lại (nếu có) và số tiền phạt (nếu có).

Ví dụ:

Chi tiết giao dịch:

ID người bảo trợ: P001

ISBN sách: A1-01-2021

Ngày thanh toán: Thứ Hai, ngày 08 tháng 5 năm

2023 Ngày đến hạn: Thứ Tư, ngày 10 tháng 5 năm

2023 Ngày trả lại: Chủ nhật, ngày 27 tháng 5 3923

Số tiền phạt: \$2081933,00

7. Chỉ định và thực hiện một `Trình quản lý thư viện` lớp chịu trách nhiệm quản lý các giao dịch sách thư viện. Lớp này phải có các hoạt động sau:

(a) Thuộc tính:

- `Danh sách<Sách> sách`: Danh sách này chứa tất cả các cuốn sách trong thư viện.
- `Danh sách các giao dịch<LibraryTransaction>`: Danh sách này chứa tất cả các giao dịch đã xảy ra trong thư viện.

(b) Phương pháp:

- `addBook(Sách sách)`-Phương pháp này thêm một cuốn sách vào thư viện.
- `Danh sách<LibraryTransaction> getCheckedOutBooks(Người bảo trợ)`-
Phương pháp này truy xuất danh sách các cuốn sách đã được kiểm tra của một người bảo trợ cụ thể.
- `thanh toánSách(Người bảo trợ, Sách sách, Ngày thanh toánNgày, Ngày ngày đáo hạn)`-Phương pháp này cho phép người bảo trợ kiểm tra một cuốn sách. Đầu tiên, nó sẽ kiểm tra xem người bảo trợ có vượt quá giới hạn thanh toán hay không dựa trên loại người bảo trợ của họ. Nếu không, nó sẽ tạo ra một cái mới `Thư việnGiao dịch`, thêm nó vào danh sách giao dịch và cập nhật số lượng bản sao có sẵn cho cuốn sách.
- `returnBook(Thư việnGiao dịch giao dịch, Ngày returnDate)`-
Phương pháp này cho phép khách hàng trả lại sách, tính tiền phạt (nếu có) và cập nhật số lượng bản có sẵn. Nó đặt ngày hoàn trả trong giao dịch, tính toán tiền phạt, cập nhật số lượng bản sao có sẵn cho cuốn sách và in thông báo thành công.
- `Danh sách<LibraryTransaction> getOverdueBooks()`-Phương thức này trả về danh sách các giao dịch thư viện đại diện cho những cuốn sách quá hạn chưa được trả lại.

Ghi chú: Để lấy ngày hiện tại dùng để tính số ngày quá hạn, bạn cần sử dụng hàm `getCurrentDate()` phương pháp trong phần được cung cấp `chung.DateUtils` lớp (bạn không nên sửa đổi lớp này).

- `loại()`-Phương pháp này sắp xếp danh sách các giao dịch theo ID người bảo trợ.

8. Chỉ định và thực hiện `Thư Viện Người Đàn Ông` `Prog` class, là lớp chương trình chính. Lớp này có một **chủ yếu** phương pháp thực hiện các nhiệm vụ sau:

- (a) Khởi tạo ít nhất **10** sách trong bộ sưu tập thư viện.
- (b) Khởi tạo ít nhất **3** khách hàng quen bao gồm cả khách hàng quen và khách hàng cao cấp.
- (c) Khởi tạo và sử dụng để tạo **5** giao dịch sổ sách
- (d) In danh sách sách hiện đang được mượn
- (e) In danh sách sách quá hạn chưa trả
- (f) Khách hàng trả lại cuốn sách
- (g) Sắp xếp giao dịch theo ID khách hàng
- (h) Kết thúc chương trình.

3. Yêu cầu nộp hồ sơ

Bạn phải gửi một tệp zip chứa mã nguồn của mình tới cổng trước ngày đáo hạn. Kích thước tối đa cho tệp zip của bạn là **1 MB**.

(*)**Ghi chú:** nội dung gửi của bạn chỉ được chứa **haigói** (`chung.Và s1_sid`). Đặt tất cả các lớp cần thiết của bạn vào `s1_sid` thư kiện.

Tên tệp zip phải có dạng `a1_sid.zip`, Ở đây **ên** là mã định danh học sinh của bạn (các bit còn lại của tên tệp không được thay đổi). Ví dụ: nếu id sinh viên của bạn là `1512345678` thì tệp zip của bạn phải được đặt tên `a1_1512345678.zip`.

QUAN TRỌNG: Việc không đặt tên tệp như được hiển thị sẽ dẫn đến không có điểm nào được đưa ra!

KHÔNG đạo văn, sẽ áp dụng hình phạt nghiêm khắc là giảm điểm!