

Báo cáo đồ án thực hành cuối kỳ:

Xây dựng hệ thống quản lý nhà sách trực tuyến

Sinh viên thực hiện: Nguyễn Hoàng Phi Hùng

MSSV: 23127194

Lớp: 23CLC06

Môn học: Phương pháp lập trình hướng đối tượng

I. Đánh giá mức độ hoàn thành

STT	Chức năng	Mức độ hoàn thành (%)	Ghi chú
1	Chức năng quản lý của hệ thống (Admin)	100%	
	1.1 Quản lý sách	100%	
	1.2 Quản lý tác giả	100%	
	1.3 Quản lý đơn hàng	100%	
	1.4 Báo cáo thống kê	100%	
	1.5 Đọc thông tin từ file	100%	
2	Tìm kiếm sách	100%	
3	Xem thông tin chi tiết sách	100%	
4	Đặt hàng	100%	
5	Thanh toán đơn hàng	90%	Chưa thiết lập được hai trạng thái đơn hàng cùng lúc như đề bài yêu cầu.
6	Quản lý đơn hàng	100%	
7	Quản lý tài khoản	100%	Hoàn thành các chức năng mà đề bài yêu cầu nhưng chưa sử dụng được cấu trúc dữ liệu phù hợp để quản lý cũng như bảo mật mật khẩu của tài khoản khách hàng.

Tổng kết: Đánh giá mức độ hoàn thành chung của toàn bộ hệ thống: 90%

II. Mô tả hệ thống

- **Mô tả các lớp:**

- **Lớp TacGia:** đây là lớp đại diện cho một tác giả và là lớp cơ bản của hệ thống. Nó chứa các thuộc tính và phương thức để quản lý thông tin cơ bản của tác giả. Lớp TacGia chứa danh sách các đối tượng Sach* để quản lý thông tin các đầu sách có liên quan đến tác giả.

- **Lớp Sach:** đây là lớp đại diện cho một cuốn sách và là lớp cơ bản trong hệ thống. Nó chứa các thuộc tính và phương thức để quản lý thông tin cơ bản của sách. Lớp Sach chứa danh sách các đối tượng TacGia* để quản lý thông tin các tác giả liên quan đến sách.

- **Lớp SachCoMoTa:** là lớp kế thừa từ lớp Sach, được mở rộng thêm thông tin mô tả.

- **Lớp KháchHang:** là lớp đại diện cho một khách hàng dùng để khai báo thông tin của khách hàng (mã khách hàng, số điện thoại,...)

- **Lớp trừu tượng TrangThaiDonHang:** dùng để biểu diễn trạng thái đơn hàng, được sử dụng tính năng đa hình để các lớp con thừa kế và triển khai các trạng thái cụ thể. Lớp cơ sở TrangThaiDonHang đóng vai trò là giao diện chung, cho phép chương trình sử dụng tính đa hình để quản lý trạng đơn hàng một cách linh hoạt. Mỗi trạng thái cụ thể của đơn hàng được triển khai bởi một lớp riêng, đảm bảo rằng mọi trạng thái có thể được xử lý độc lập nhưng vẫn có chung giao diện.

- **Lớp DonHang:** đây là lớp đại diện cho một đơn hàng và là lớp cơ bản của hệ thống. Nó được thiết kế để quản lý các thông tin về đơn hàng bao gồm sản phẩm, trạng thái đơn hàng và thông tin người mua.

+ **Lớp DonHang** có một thuộc tính trangThai kiểu TrangThaiDonHang* để biểu thị trạng thái hiện tại của đơn hàng.

+ **Lớp DonHang** có thuộc tính dsSanPham để lưu sản phẩm với mỗi sản phẩm là đối tượng Sach* và số lượng của nó.

+ **Lớp DonHang** có thuộc tính kháchHang kiểu KháchHang* để lưu thông tin người mua.

- **Lớp QuanLySach:** Đây là **lớp cơ sở trừu tượng** (abstract class) được thiết kế để quản lý các hành động trên sách. Mỗi lớp trong QuanLySach đại diện cho một hành động cụ thể trong việc quản lý đơn hàng bao gồm: thêm đơn sách, xoá sách, chỉnh sửa sách.

- **Lớp QuanLyTacGia:** Đây là **lớp cơ sở trừu tượng** (abstract class) được thiết kế để quản lý các hành động trên tác giả. Mỗi lớp trong QuanLyTacGia đại diện cho một hành động cụ thể trong việc quản lý tá giả bao gồm: thêm tác giả, xoá tác giả, chỉnh sửa tác giả.

- **Lớp QuanLyDonHang:** Đây là **lớp cơ sở trừu tượng** (abstract class) được thiết kế để quản lý các hành động trên đơn hàng. Mỗi lớp trong QuanLyDonHang đại diện cho một hành động cụ thể trong việc quản lý đơn hàng bao gồm: thêm đơn hàng, xoá đơn hàng, chỉnh sửa đơn hàng.
- **Lớp ThanhToan:** Đây là **lớp cơ sở trừu tượng** (abstract class) đại diện cho phương thức thanh toán. Mỗi lớp dẫn xuất của lớp ThanhToan đại diện cho một phương thức thanh toán cụ thể.
- **Lớp BoiCanhThanhToan:** Là lớp đóng vai trò quản lý chiến lược thanh toán. Sử dụng **mẫu thiết kế Strategy** để linh hoạt thay đổi phương thức thanh toán.
- **Lớp TaiKhoan:** quản lý thông tin tài khoản của khách hàng, bao gồm mã khách hàng và mật khẩu.
- **Lớp HeThongQuanLy:** được thiết kế để quản lý toàn bộ hệ thống bao gồm sách, tác giả, đơn hàng, khách hàng, và tài khoản nên nó sẽ chứa các danh sách sách, tác giả, khách hàng cũng như đơn hàng. Lớp này sử dụng mẫu thiết kế Singleton để đảm bảo rằng chỉ có một thể hiện duy nhất của hệ thống được khởi tạo.
- **Lớp NguoaiDung:** là một lớp cơ sở trừu tượng (abstract class) sử dụng phương pháp **interface** để đảm bảo các lớp con (như QuanTriVien, NguoaiDungThongThuong, NguoaiDungProxy) triển khai đầy đủ các phương thức cần thiết cho chức năng của từng loại người dùng trong hệ thống. Lớp này có chức năng phân quyền cho người dùng truy cập vào hệ thống.

- **Sơ đồ UML:**
- Xem đính kèm trong file **uml.pdf**

III. Mẫu thiết kế và kỹ thuật:

STT	Mẫu thiết kế / Kỹ thuật	Lý do lựa chọn	Mô tả cách áp dụng
1	Singleton	Đảm bảo chỉ có một thể hiện duy nhất trong suốt chương trình chạy, cung cấp cách thức truy xuất toàn cục cho chương trình, nhằm tránh xung đột tài nguyên.	Lớp HeThongQuanLy được thiết kế theo mẫu Singleton được áp dụng để đảm bảo chỉ có một hệ thống quản lý duy nhất trong chương trình, và tất cả các thành phần khác đều truy cập cùng một thể hiện của hệ thống này.

2	Proxy	Để phân quyền người dùng khi người dùng truy cập vào chương trình từ đó sẽ điều phối nhữn hàm mà người dùng được sử dụng với vai trò tương ứng.	Lớp NguoiDung được thiết kế theo mẫu Proxy Pattern. Trong chương trình lớp NguoiDungProxy được thiết lập để kiểm soát quyền truy cập vào các hàm của hai loại người dùng là người dùng thông thường và quản trị viên lần lượt được triển khai trong các lớp NguoiDungThongThuong và QuanTriVien.
3	Command	Đóng gói các yêu cầu vào một đối tượng riêng biệt cho phép ta tách biệt giữa 2 hành vi của người dùng là gửi yêu cầu và thực thi yêu cầu.	Trong các lớp QuanLySach, QuanLyTacGia và QuanLyDonHang được áp dụng command pattern để để quản lý các hành động liên quan đến việc xử lý như thêm, hủy và chỉnh sửa.
4	Factory	Thiết lập đối tượng mà không làm lộ chi tiết của lớp đối tượng hoặc khi ta cần dễ dàng thay đổi loại đối tượng được tạo ra mà không ảnh hưởng đến chương trình gốc.	Được áp dụng trong lớp TrangThaiDonHang để khi ta chỉnh sửa đơn hàng đặc biệt là trạng thái đơn hàng ta có thể dễ dàng thay đổi trạng thái của đơn hàng (Ví dụ: DaDat -> DaThanhToan)
5	Strategy	Cho phép thay đổi hành vi của đối tượng mà không cần thay đổi mã nguồn của nó. Strategy pattern tách biệt các thuật toán (hoặc chiến lược) thành các lớp riêng biệt và cho phép các đối tượng thay đổi chiến lược này tại thời điểm chạy.	Được áp dụng ở lớp thanhToan để người dùng có thể linh hoạt trong việc thay đổi phương thức thanh toán khi phương thức thanh toán hiện tại không thành công mà không ảnh hưởng đến chương trình.

IV. Nhìn lại và rút kinh nghiệm

1. Đánh giá quá trình làm việc:

- Thuận lợi:

- + Được hướng dẫn vẽ UML để hệ thống hoá chương trình trước khi code.
- + Được dạy trên lớp một vài design pattern đơn giản để áp dụng (Singleton, Factory)

- Khó khăn:

- + Tìm hiểu các pattern mới cần sử dụng cho chương trình trong thời gian ngắn gây ra nhiều khó khăn trong việc triển khai trong việc lập trình.
- + Khó khăn trong việc quản lý các dữ liệu đầu vào trong lúc đọc file do các Lớp có các khai báo lồng nhau (Lớp Sach có dsTacgia và Lớp TacGia có dsSach) đòi hỏi người viết chương trình cần phải xử lý một cách hợp lý.
- + Khó khăn trong việc mở rộng thêm trạng thái đơn hàng cũng như bảo mật mật khẩu cho tài khoản dành cho khách hàng.

2. Bài học kinh nghiệm:

- Cần phân tích và thiết kế hệ thống kỹ lưỡng trước khi bắt tay vào viết code.
- Luôn tìm design pattern phù hợp với chương trình rồi mới code tránh trường hợp code rồi mới thêm design pattern vào sau.
- Tìm hướng đọc file phù hợp cho chương trình để tránh tình trạng memory leak.
- Phân chia thời gian hợp lý hơn cho các yêu cầu nhằm tránh tình trạng không làm kịp.

3. Hướng phát triển và cải tiến hệ thống (nếu có):

-Thêm hoạt động thuê sách để cho người dùng có thể thuê sách và trả lại sau.

-Phân chia khách hàng thành khách hàng thường và khách hàng thân thiết để có những ưu đãi phù hợp

V. Tham khảo

(Liệt kê các nguồn tài liệu, sách, trang web đã tham khảo trong quá trình làm bài)