**VIỆN ĐẠI HỌC MỞ HÀ NỘI**

**KHOA ĐÀO TẠO SAU ĐẠI HỌC**

**LỚP CAO HỌC CÔNG NGHỆ THÔNG TIN - KHÓA 2**

**--------------------------**

****

**BÀI TẬP LỚN**

**MÔN CƠ SỞ DỮ LIỆU NÂNG CAO**

**GIẢNG VIÊN HƯỚNG DẪN: TS. TRƯƠNG TIẾN TÙNG**

**NHÓM HỌC VIÊN THỰC HIỆN: NHÓM 8**

**Phạm Tuân: Nhóm Trưởng**

**Phan Bá Tiệp**

**Trần Thị Như Hoa**

**Phạm Thị Hải Vân**

**Hà Nội, tháng 06/2016**

**LỜI CẢM ƠN**

Quá trình tìm hiểu về môn học Cơ sở dữ liệu nâng cao và các nội dung liên quan đến môn học đã giúp chúng em có những cái nhìn khái quát hơn về môn học này. Giúp chúng em nắm được những kiến thức cơ bản đến cốt lõi của vấn đề.

Tuy nhiên, quá trình tìm hiểu về cơ sở dữ liệu vẫn còn nhiều hạn chế cả về khả năng lẫn kinh nghiệm thực tiễn, nên trong quá trình làm đồ án môn học của chúng em khó tránh khỏi những thiếu sót. Chúng em sẽ cố gắng tiếp tục tìm hiểu sâu hơn nữa, đặc biệt là hoàn thiện chuyên đề về Quản lý thư viện để có thể ứng dụng vào thực tiễn tốt hơn. Rất mong được sự thông cảm và những góp ý của thầy và các bạn để đồ án môn học được hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn TS. TrươngTiến Tùng đã truyền đạt cho chúng em những bài học rất hữu ích về chuyên đề này, đây là những kiến thức cần thiết hữu dụng sẽ sử dụng trong tất các các tầng của lập trình, giúp chúng em có cái nhìn tổng quát cũng như chuyên sâu về cơ sở dữ liệu nói chung và cơ sở dữ liệu hướng đối tượng nói riêng.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

[MỞ ĐẦU 1](#_Toc461574000)

[CHƯƠNG 1. KHÁI QUÁT VỀ CSDL VÀ CSDL HƯỚNG ĐỐI TƯỢNG 2](#_Toc461574001)

[1.1. Khái quát về cơ sở dữ liệu 2](#_Toc461574002)

[1.1.1. Khái niệm 2](#_Toc461574003)

[1.1.2. Ưu điểm của cơ sở dữ liệu 2](#_Toc461574004)

[1.1.3. Vấn đề cơ sở dữ liệu cần giải quyết 2](#_Toc461574005)

[1.1.4. Hệ quản trị cơ sở dữ liệu 3](#_Toc461574006)

[1.2. Cơ sở dữ liệu hướng đối tượng 4](#_Toc461574007)

[1.2.1. Khái niệm cơ sở dữ liệu hướng đối tượng và các khái niệm liên quan 4](#_Toc461574008)

[1.2.2. Ưu điểm của cơ sở dữ liệu hướng đối tượng 5](#_Toc461574009)

[1.2.3. Tính chất của mô hình quản lý cơ sở dữ liệu hướng đối tượng 6](#_Toc461574010)

[CHƯƠNG 2. THIẾT KẾ CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG 7](#_Toc461574011)

[2.1. Mô hình cơ sở dữ liệu hướng đối tượng 7](#_Toc461574012)

[2.2. Thiết kế cơ sở dữ liệu hướng đối tượng 10](#_Toc461574013)

[2.2.1. Phân lớp 10](#_Toc461574014)

[2.2.2. Tổng quát hóa và đặc biệt hóa 10](#_Toc461574015)

[2.2.3. Gộp 11](#_Toc461574016)

[2.3. Các giai đoạn thiết kế 12](#_Toc461574017)

[2.3.1. Phân tích hướng đối tượng (Object Oriented Analysis - OOA) 12](#_Toc461574018)

[2.3.2.Thiết kế hướng đối tượng (Object Oriented Design - OOD) 13](#_Toc461574019)

[2.3.3. Lập trình hướng đối tượng (Object Oriented Programming - OOP) 13](#_Toc461574020)

[2.4. Xây dựng cơ sở dữ liệu hướng đối tượng 14](#_Toc461574021)

[CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG CHO HỆ THỐNG QUẢN LÝ THƯ VIỆN 16](#_Toc461574022)

[3.1. Các yêu cầu của bài toán 16](#_Toc461574023)

[3.2. Phân tích các yêu cầu 16](#_Toc461574024)

[3.2.1. Các tác nhân ngoài 16](#_Toc461574025)

[3.2.2. Các ca sử dụng (chức năng chính) 17](#_Toc461574026)

[3.2.3. Xây dựng biểu đồ ca sử dụng 18](#_Toc461574027)

[3.2.3.1. Nếu bạn đọc không có phiếu đặt mượn trước 19](#_Toc461574028)

[3.2.3.2. Nếu bạn đọc có phiếu đặt mượn trước 19](#_Toc461574029)

[3.3. Phân tích miền bài toán – xây dựng mô hình khái niệm 19](#_Toc461574030)

[3.3.1. Xác định các lớp đối tượng 19](#_Toc461574031)

[3.3.2. Xây dựng biểu đồ lớp 21](#_Toc461574032)

[3.4. Thiết kế chi tiết 21](#_Toc461574033)

[3.4.1. Thiết kế chi tiết gói CSDL 22](#_Toc461574034)

[3.4.2. Thiết kế chi tiết gói các đối tượng tác nghiệp 23](#_Toc461574035)

[3.4.5. Kết quả tạo được trong bảng cơ sở dữ liệu 24](#_Toc461574036)

[KẾT LUẬN 27](#_Toc461574037)

# MỞ ĐẦU

Mô hình dữ liệu hướng đối tượng ra đời từ cuối những năm 80 và đầu những năm 90. Đây là một trong những mô hình tiên tiến dựa trên cách tiếp cận hướng đối tượng đã quen thuộc trong các phương pháp lập trình hướng đối tượng. Phương pháp này sử dụng các khái niệm như lớp (class), sự kế thừa (inheritance), kế thừa bội (tức là kế thừa từ nhiều lớp multi-inheritance). Đặc trưng cơ bản của các tiếp cận này là tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính tái sử dụng (resuability).

Thực tế cho thấy CSDL hướng đối tượng có một số các ưu điểm sau đây:

* Cho phép xét các liên kết đối tượng dưới dạng các phép lưu trữ với các đối tượng.
* Các đối tượng dùng chung giữa nhiều người sử dụng.
* Khả năng phát triển kho tri thức bằng cách thêm các đối tượng mới và các phép xử lý kèm theo.
* Phát triển hệ quản trị CSDL dựa trên việc xử lý các đối tượng phức tạp, giao diện chương trình, đối tượng động và trừu tượng.
* Phương pháp phân tích và thiết kế hướng đối tượng thực hiện theo các thuật ngữ và khái niệm của phạm vi lĩnh vực ứng dụng (tức là của doanh nghiệp hay đơn vị mà hệ thống tương lai cần phục vụ), nên nó tạo sự tiếp cận tương ứng giữa hệ thống và vấn đề thực ngoài đời. Ví dụ trong bán xe ô tô, mọi giai đoạn phân tích thiết kế và thực hiện đều xoay quanh các khái niệm như khách hàng, nhân viên bán hàng, xe ô tô v.v. Vì quá trình phát triển phần mềm đồng thời là quá trình cộng tác của khách hàng/người dùng, nhà phân tích, nhà thiết kế, nhà phát triển, chuyên gia lĩnh vực, chuyên gia kỹ thuật v.v. nên cách tiếp cận này khiến cho việc giao tiếp giữa họ với nhau được dễ dàng hơn.
* Một trong những ưu điểm quan trọng bậc nhất của phương pháp phân tích và thiết kế hướng đối tượng là tính tái sử dụng: bạn có thể tạo các thành phần (đối tượng) một lần và dùng chúng nhiều lần sau đó. Giống như việc bạn có thể tái sử dụng các khối xây dựng (hay bản sao của nó) trong một toà lâu đài, một ngôi nhà ở, một con tàu vũ trụ, bạn cũng có thể tái sử dụng các thành phần (đối tượng) căn bản trong các thiết kế hướng đối tượng cũng như code của một hệ thống kế toán, hệ thống kiểm kê, hoặc một hệ thống đặt hàng.

Vì các đối tượng đã được thử nghiệm kỹ càng trong lần dùng trước đó, nên khả năng tái sử dụng đối tượng có ưu điểm giảm thiểu lỗi và khó khăn trong việc bảo trì, giúp tăng tốc độ thiết kế và phát triển phần mềm.

Từ những lý do trên, chúng em quyết định chọn đề tài: “THIẾT KẾ VÀ XÂY DỰNG CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG CHO HỆ THỐNG QUẢN LÝ THƯ VIỆN” làm đề tài nghiên cứu cho môn học Cơ sở dữ liệu nâng cao.

# CHƯƠNG 1. KHÁI QUÁT VỀ CSDL VÀ CSDL HƯỚNG ĐỐI TƯỢNG

## 1.1. Khái quát về cơ sở dữ liệu

### 1.1.1. Khái niệm

Một cơ sở dữ liệu là một tập hợp các dữ liệu có liên quan với nhau, trữ trên máy tính, có nhiều người sử dụng và được tổ chức theo một mô hình. Dữ liệu là những sự kiện có thể ghi lại được và có ý nghĩa.

Ví dụ, để quản lý việc học tập trong môi trường đại học, các dữ liệu là các thông tin về sinh viên, môn học, điểm thi v.v. Chúng ta sẽ tổ chức các dữ liệu đó thành các bảng và lưu dữ chúng vào sổ sách hoặc sử dụng một phần mềm máy tính để lưu giữ chúng trên máy tính. Như vậy, ta có một tập các dữ liệu có liên quan đến nhau và mang nhiều ý nghĩa đó là một cơ sở dữ liệu.

### 1.1.2. Ưu điểm của cơ sở dữ liệu

- Giảm sự trùng lặp thông tin xuống mức thấp nhất. Do đó đảm bảo thông tin có tính nhất quán và toàn vẹn dữ liệu.

- Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.

- Nhiều người có thể sử dụng một cơ sở dữ liệu.

### 1.1.3. Vấn đề cơ sở dữ liệu cần giải quyết

*- Tính chủ quyền của dữ liệu*

+ Thể hiện ở phương diện an toàn dữ liệu.

+ Khả năng biểu diễn mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu.

+ Người khai thác cơ sở dữ liệu phải cập nhật cho CSDL những thông tin mới nhất.

*- Tính bảo mật và quyền khai thác thông tin của người sử dụng*

+ Do ưu điểm CSDL có thể cho nhiều người khai thác đồng thời, nên cần phải có một cơ chế bảo mật phân quyền khai thác CSDL.

+ Các hệ điều hành nhiều người sử dụng hay cục bộ đều cung cấp cơ chế này.

*- Tranh chấp dữ liệu*

+ Khi nhiều người cùng truy nhập CSDL với các mục đích khác nhau, rất có thể sẽ xảy ra hiện tượng tranh chấp dữ liệu.

+ Cần có cơ chế ưu tiên khi truy cập CSDL. Ví dụ: admin luôn có thể truy cập cơ sở dữ liệu.

+ Cấp quyền ưu tiên cho từng người khai thác.

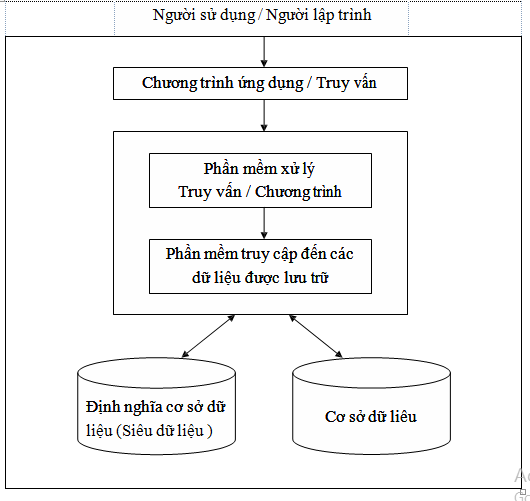
*- Đảm bảo an toàn dữ liệu khi có sự cố*

+ Khi CSDL nhiều và được quản lý tập trung, khả năng rủi ro mất dữ liệu rất cao. Các nguyên nhân chính là mất điện đột ngột hoặc hỏng thiết bị lưu trữ.

+ Hiện tại có một số hệ điều hành đã có cơ chế tự động sao lưu ổ cứng và fix lỗi khi có sự cố xảy ra.

### 1.1.4. Hệ quản trị cơ sở dữ liệu

Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu được gọi chung bằng một thuật ngữ là ***hệ cơ sở dữ liệu.***



**Hình 1.1: *Môi trường hệ cơ sở dữ liệu***

## 1.2. Cơ sở dữ liệu hướng đối tượng

### 1.2.1. Khái niệm cơ sở dữ liệu hướng đối tượng và các khái niệm liên quan

- Một cơ sở dữ liệu hướng đối tượng (hệ thống quản lý cơ sở dữ liệu hướng đối tượng – OODBMS) là một hệ thống quản lý cơ sở dữ liệu, trong đó thông tin được đại diện trong các hình thức của các đối tượng như được sử dụng trong lập trình hướng đối tượng.

Đối tượng là một thực thể có vai trò xác định rõ ràng trong lĩnh vực ứng dụng, có trạng thái, hành vi và được xác định tên.

Ví dụ về các đối tượng thuộc lớp Người. Chúng liên lạc với nhau thông qua thông báo. Thông báo là dạng các thao tác áp dụng lên đối tượng. Thao tác trong môi trường hướng đối tượng được gọi là phương pháp. Chẳng hạn, phương pháp kết hôn tác động lên đối tượng để biết đối tượng này kết hôn với ai.

Lớp đối tượng là tập các đối tượng có chung cấu trúc và hành vi.

- Sơ đồ lớp cho biết cấu trúc tĩnh của mô hình hướng đối tượng, đó là các lớp đối tượng, cấu trúc bên trong của chúng và mối quan hệ mà chúng tham gia.

- Sơ đồ đối tượng là đồ thị gồm các thể hiện của đối tượng, tương thích với sơ đồ lớp.

- Bao gói hay bao bọc là kỹ thuật che giấu, là mẫu những chi tiết về cài đặt bên trong của đối tượng đối với các truy cập từ bên ngoài.

- Cá thể hóa là quá trình khẳng định sự tồn tại của các đối tượng trong môi trường hướng đối tượng, bằng việc xác định lớp của chúng. Mỗi đối tượng là một cá thể của lớp, thường được dùng với thuật ngữ thể hiện của lớp.

- Kế thừa là tính chất cho phép các lớp con kế thừa những thuộc tính, phép toán của lớp cha.

Việc kế thừa nhiều lần xảy ra khi một lớp kế thừa từ nhiều lớp.

### 1.2.2. Ưu điểm của cơ sở dữ liệu hướng đối tượng

- Hỗ trợ những kiểu dữ liệu được định nghĩa bởi người sử dụng: Cơ sở dữ liệu hướng đối tượng có khả năng lưu trữ những kiểu dữ liệu phức hợp, kiểu được định nghĩa bởi người sử dụng, các lớp và thao tác trên chúng một cách dễ dàng.

- Cung cấp một hình mẫu phát triển cơ sở dữ liệu cho cả phân tích, thiết kế và cài đặt ứng dụng.

- Cải tiến đáng kể về chất lượng dữ liệu: Có thể đưa ra nhiều ràng buộc vào cấu trúc dữ liệu. Mô hình còn cho phép thể hiện cả những ràng buộc không cấu trúc để chương trình phải thỏa mãn khi nó thực hiện trong cơ sở dữ liệu. Một cơ sơ dữ liệu hướng đối tượng có thể chuyển đổi về một cơ sở dữ liệu quan hệ được chuẩn hóa.

- Tốc độ phát triển phần mềm nhanh hơn: Cấu trúc cơ sở dữ liệu nhất quán và rõ ràng giúp cho lập trình ứng dụng trở nên đơn giản và nhanh hơn. Những người phát triển ứng dụng có kinh nghiệm thường sử dụng những câu lệnh phức tạp của hệ quản trị cơ sở dữ liệu thay cho những đoạn chương trình của người sử dụng.

- Tích hợp dễ dàng: Việc tích hợp với nhiều hệ thống độc lập có thể giảm bớt sự sao chép dữ liệu của con người và mở rộng những câu truy vấn có thể trả lời. Mô hình hướng đối tượng cung cấp cách biểu diễn thống nhất làm thuận tiện hơn cho việc tìm hiểu và tích hợp thông tin.

### 1.2.3. Tính chất của mô hình quản lý cơ sở dữ liệu hướng đối tượng

Để đánh giá một hệ quản trị CSDL đối tượng, trước hết hệ thông phải có các chức năng của một hệ quản trị CSDL. Ngoài ra, phải đảm bảo một số tính chất dưới đây.

**a. Tính bền vững**

Các đối tượng chắc chắn phải được lưu trữ trên các phương tiện nhớ như đầu từ khi được một chương trình tạo ra.

Đối tượng bền vững: Là đối tượng được lưu giữ trong CSDL, có thời gian tồn tại dài hơn thời gian của chương trình tạo ra đối tượng đó.

Đối tượng tạm thời: Là đối tượng được lưu trong bộ nhớ trong. Do vậy thời gian tồn tại của nó không quá thời hạn của chương trình tạo ra đối tượng đó.

**b. Tính khai thác tương tranh**

CSDL đối tượng cho phép thức hiện nhiều thao tác trên cùng một đối tượng. Việc khóa các thao tác trên dữ liệu cần hạn chế để đảm bảo tính tương hợp và nhất quán về dữ liệu.

**c. Tính tin cậy của đối tượng**

Những đối tượng có thể khôi phục lại khi có sai xót xảy ra. Các thao tác cần chia nhỏ để đảm bảo hoặc chúng được thực hiện hoàn toàn hoặc không thực hiện được bất kì thao tác nào.

**d. Tiện lợi tra cứu**

Các đối tượng cần được thiết kế và quản lý các giá trị của thuộc tính, các kết quả quả của thao tác tính toán, các liên hệ giữa các đối tượng để thuận tiện cho việc tra cứu, tìm kiếm các đối tượng thỏa mãn một yêu cầu nào đó của người sử dụng.

# CHƯƠNG 2. THIẾT KẾ CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG

## 2.1. Mô hình cơ sở dữ liệu hướng đối tượng

Mô hình cơ sở dữ liệu hướng đối tượng (OODM – Oriented Object Database Model) là một mô hình dữ liệu mà ngữ nghĩa của các đối tượng được hỗ trợ cho lập trình hướng đối tượng.

Mô hình hướng đối tượng dựa trên cơ sở các gói dữ liệu và mã liên quan tới đối tượng. Mọi sự tương tác của người sử dụng với các đối tượng được quản lý này đều thông qua các thông điệp, hoặc sự tương tác giữa đối tượng này với đối tượng khác đều phải thông qua thông điệp. Đối tượng bao giờ cũng có mã thao tác, những mã này được gọi là những phương pháp. Những đối tượng có chung kiểu giá trị và cùng phương pháp thì được gộp vào một lớp. Thực thể là một vật thực hay một đối tượng tồn tại một cách khách quan, chúng ta có thể phân biệt chúng nhờ chỉ số của chúng trong máy tính. Nếu các thực thể không có chỉ số thứ tự tự nhiên thì chúng ta phải tạo ra các chỉ số cho chúng. Thuộc tính hay còn gọi là tính chất của các thực thể được biểu thị trong các trường (cột) của bảng. Giữa các thực thể có các mối quan hệ với nhau. Các quan hệ giữa các thực thể có thể là 1 – 1, 1 – nhiều, và nhiều – nhiều.

- Đối tượng (Object): Bộ sưu tập các yếu tố DL có cấu trúc, được đồng nhất bởi một dẫn trỏ (tham chiếu) duy nhất.

+ Mọi đối tượng đều được đặc trưng bằng một tên duy nhất, gọi là OID (Object Indentifier).

+ Hai đối tượng là đồng nhất (O1==O2) nếu chúng có cùng OID.

+ Hai đối tượng là bằng nhau (O1=O2) nếu chúng có cùng giá trị.

+ Các đối tượng đặc trưng bởi các tính chất.

- Tính chất (Property): Đặc trưng của một đối tượng được chỉ định bằng một tên có thể ứng với một thuộc tính, một hàm hay một đối tượng con thành phần, ví dụ:

+ Thuộc tính đơn : Tên của một người...

+ Hàm : Hàm tuổi (của một người) ...

+ Thuộc tính kép : Các con của một người...

- Lớp: Nhóm các đối tượng có cùng tính chất, được đặc trưng bởi một cấu trúc và tập các phép toán tác dụng lên các đối tượng của lớp bằng cách che dấu cấu trúc.

+ Việc đặc tả tiến triển của các lớp đối tượng làm thành một CSDL hướng đối tượng, cho phép mô hình hoá hành vi chung của các đối tượng một cách đơn thể và mở rộng được.

Chẳng hạn như : Các hình tròn…

- Phương pháp: Thao tác liên kết với một lớp, xử lý hay đưa trả lại trạng thái của một đối tượng hay một phần của đối tượng thuộc lớp.

+ Một đối tượng được thao tác bởi phương pháp của lớp và được thấy qua các phương pháp như nguyên lý bọc kín.

+ Phương pháp có thể áp dụng được cho nhiều đối tượng thuộc các lớp khác nhau : đa lớp 🡪 dùng để mô hình hoá các mối liên kết giữa các lớp.

|  |
| --- |
| **package** org.nhom8.dttn;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.Id;  @Entity  **publicclass** Sach **extends** DauMuc {  @Id  @GeneratedValue  **private** Long id;  **private** String author;  **private** String publish;  **privateint**year;  **privateint**numOfPage;  **privateint**defaultTime = 21;  **public** String getAuthor() {  **return**author;  }  **publicvoid** setAuthor(String author) {  **this**.author = author;  }  **public** String getPublish() {  **return**publish;  }  **publicvoid** setPublish(String publish) {  **this**.publish = publish;  }  **publicint** getYear() {  **return**year;  }  **publicvoid** setYear(**int**year) {  **this**.year = year;  }  **publicint** getNumOfPage() {  **return**numOfPage;  }  **publicvoid** setNumOfPage(**int**numOfPage) {  **this**.numOfPage = numOfPage;  }  **publicvoid** read() { }  **publicvoid** write() { }  **public** Long getId() {  **return**id;  }  @Override  **public** String toString() {  **return**"Sach [id=" + id + ", name=" + getName() + ", author=" + author + "]";  }  } |

**Hình 2.1: *Ví dụ về lớp Sach trong cơ sở dữ liệu hướng đối tượng***

Cơ sở dữ liệu hướng đối tượng thường được mô tả thông qua lược đồ. Mô tả của nó phát sinh một lược đồ bao gồm:

Mô tả các lớp, mỗi lớp bao gồm các tính chất (tuỳ theo tình hình được tổ chức thành các nhóm bởi các toán tử xây dựng) và các phương pháp.

Mô tả các mối liên kết giữa các lớp.

|  |
| --- |
|  |

**Hình 2.2: *Lược đồ đối tượng của cơ sở dữ liệu Quản lý thư viện***

## 2.2. Thiết kế cơ sở dữ liệu hướng đối tượng

### 2.2.1. Phân lớp

Quá trình phân lớp liên quan đến việc định tên đối tượng với các thuộc tính, hành vi tương tự nhau và nhóm các đối tượng vào cùng một lớp. Theo ví dụ về sơ đồ người ta xác định sơ đồ với các thuộc tính tên, ngày tạo, hình vẽ. Các phép toán chung là lưu trữ tìm kiếm, vẽ.

Trong đoạn chương trình trên, danh sách các trường và các kiểu dữ liệu đơn giản dùng cho các sơ đồ được liệt kê trong mục thuộc tính. Tiếp theo là các phương thức, có tên và các tham số. Có một số phương thức như tạo mới, xóa v.v. áp dụng cho tất cả các đối tượng trong CSDL.

Tất cả các định nghĩa về giao diện lớp đối tượng cần có phép toán tạo mới và hủy bỏ đối tượng. Quá trình phân lớp sẽ tạo ra lớp của các đối tượng có các thuộc tính, phương thức chung và một vài đối tượng có thuộc tính và phương thức riêng. Lúc đó người ta cần đến khái niệm tổng quát hóa và chuyên biệt hóa.

### 2.2.2. Tổng quát hóa và đặc biệt hóa

Tổng quát hóa là quá trình xác định lớp đối tượng mang các thuộc tính tương tự và theo sự tương tự này người ta có thể trừu tượng hóa để được lớp cao hơn, hay lớp cha. Chẳng hạn, ban đầu người ta xác định lớp hình tam giác, hình vuông, hình chữ nhật và hình tròn rồi trừu tượng hóa thành lớp cao hơn về sơ đồ, gồm các thuộc tính chung của tất cả các sơ đồ.

Lớp trừu tượng là lớp không có thể hiện trực tiếp, nhưng các thành phần sau nó có thể có thể hiện trực tiếp.

Lớp cụ thể là lớp có cách thể hiện trực tiếp như sơ đồ dưới

|  |  |
| --- | --- |
| |  | | --- | |  | |
| **Hình 2.3: Kế thừa của các lớp** |

Chuyên biệt hóa là quá trình ngược lại với tổng quát hóa. Bắt đầu từ lớp sơ đồ, người ta có thể xác định lớp con để phân biệt các loại sơ đồ khác nhau. Mỗi lớp con chia sẻ thuộc tính và phương thức chung trong lớp sơ đồ nhưng có các thuộc tính và phương thức dùng riêng.

### 2.2.3. Gộp

Gộp là quá trình liên kết các lớp đối tượng với nhau để tạo nên một lớp gộp. Chẳng hạn, CSDL ngân hàng gồm khách hàng, tài khoản, chi nhánh và mối quan hệ giữa chúng. Khi thực hiện CSDL này theo kỹ thuật quan hệ, người ta tạo các quan hệ tách biệt đối với mỗi thực thể và dùng khái niệm khóa ngoài để thể hiện mối quan hệ 1 – N. Người ta cũng dùng quan hệ khác có khóa ngoài để thể hiện quan hệ 1 – N giữa khách hàng và tài khoản. Để thiết lập khung nhìn về tất cả thông tin liên quan đến một tài khoản nào đó, người ta cần nối 4 bảng cơ sở.

Trong CSDL hướng đối tượng người ta giải quyết vấn đề này nhờ lớp đối tượng gộp. Lớp đăng ký được tạo, lên kết các đối tượng kiểu khách hàng, tài khoản và chi nhánh. Mỗi đối tượng đăng ký sẽ liên kết một khách hàng với một tài khoản, có thuộc tính đặc biệt cho biết thời gian thực hiện đăng ký này.

## 2.3. Các giai đoạn thiết kế

Mô hình hướng đối tượng đã thể hiện cấu trúc, cho phép mô hình hóa các đối tượng và các liên kết giữa các đối tượng. Mặt khác mô hình hướng đối tượng còn thể hiện khía cạnh động, cho phép quản lý cấu trúc của các đối tượng theo chức năng, còn được gọi là phương pháp thiết kế hướng đối tượng.

Phương pháp hướng đối tượng giúp chúng ta xử lý các vấn đề phức tạp trong phát triển phần mềm và tạo ra các thế hệ phần mềm có khả năng thích ứng và bền chắc. Các giai đoạn của chu trình phát triển phần mềm với mô hình hướng đối tượng:

### 2.3.1. Phân tích hướng đối tượng (Object Oriented Analysis - OOA)

Là giai đọan phát triển một mô hình chính xác và súc tích của vấn đề, có thành phần là các đối tượng và khái niệm đời thực, dễ hiểu đối với người sử dụng. Trong giai đoạn OOA, vấn đề được trình bày bằng các thuật ngữ tương ứng với các đối tượng có thực. Thêm vào đó, hệ thống cần phải được định nghĩa sao cho người không chuyên Tin học có thể dễ dàng hiểu được. Dựa trên một vấn đề có sẵn, nhà phân tích cần ánh xạ các đối tượng hay thực thể có thực như khách hàng, ô tô, người bán hàng, … vào thiết kế để tạo ra được bản thiết kế gần cận với tình huống thực. Mô hình thiết kế sẽ chứa các thực thể trong một vấn đề có thực và giữ nguyên các mẫu hình về cấu trúc, quan hệ cũng như hành vi của chúng. Nói một cách khác, sử dụng phương pháp hướng đối tượng chúng ta có thể mô hình hóa các thực thể thuộc một vấn đề có thực mà vẫn giữ được cấu trúc, quan hệ cũng như hành vi của chúng.

Đối với ví dụ một thư viện, giai đoạn OOA sẽ nhận biết được các thực thể như:

- Bạn đọc

- Thủ thư

- Phiếu đặt mua

- Phiếu đặt mượng

- Sách

Tương tác và quan hệ giữa các đối tượng trên là:

- Bạn đọc sẽ đến thư viện tìm sách mình cần mượn.

- Bạn đọc chọn một quyển sách hoặc tạp chí cần mượn.

- Bạn đọc viết tên sách, tạp chí cần mượn.

- Thủ thư ghi phiếu đặt mượn.

- Sách hoặc tạp chí được giao cho bạn đọc.

### 2.3.2. Thiết kế hướng đối tượng (Object Oriented Design - OOD)

Là giai đoạn tổ chức chương trình thành các tập hợp đối tượng cộng tác, mỗi đối tượng trong đó là thực thể của một lớp. Các lớp là thành viên của một cây cấu trúc với mối quan hệ thừa kế.

Mục đích của giai đoạn OOD là tạo thiết kế dựa trên kết quả của giai đoạn OOA, dựa trên những quy định phi chức năng, những yêu cầu về môi trường, những yêu cầu về khả năng thực thi.... OOD tập trung vào việc cải thiện kết quả của OOA, tối ưu hóa giải pháp đã được cung cấp trong khi vẫn đảm bảo thoả mãn tất cả các yêu cầu đã được xác lập.

Trong giai đoạn OOD, nhà thiết kế định nghĩa các chức năng, thủ tục (operations), thuộc tính (attributes) cũng như mối quan hệ của một hay nhiều lớp (class) và quyết định chúng cần phải được điều chỉnh sao cho phù hợp với môi trường phát triển. Đây cũng là giai đoạn để thiết kế ngân hàng dữ liệu và áp dụng các kỹ thuật tiêu chuẩn hóa.

Về cuối giai đoạn OOD, nhà thiết kế đưa ra một loạt các biểu đồ (diagram) khác nhau. Các biểu đồ này có thể được chia thành hai nhóm chính là Tĩnh và động. Các biểu đồ tĩnh biểu thị các lớp và đối tượng, trong khi biểu đồ động biểu thị tương tác giữa các lớp và phương thức hoạt động chính xác của chúng. Các lớp đó sau này có thể được nhóm thành các gói (Packages) tức là các đơn vị thành phần nhỏ hơn của ứng dụng.

### 2.3.3. Lập trình hướng đối tượng (Object Oriented Programming - OOP)

Giai đoạn xây dựng phần mềm có thể được thực hiện sử dụng kỹ thuật lập trình hướng đối tượng. Đó là phương thức thực hiện thiết kế hướng đối tượng qua việc sử dụng một ngôn ngữ lập trình có hỗ trợ các tính năng hướng đối tượng. Một vài ngôn ngữ hướng đối tượng thường được nhắc tới là C++ và Java. Kết quả chung cuộc của giai đoạn này là một loạt các code chạy được, nó chỉ được đưa vào sử dụng sau khi đã trải qua nhiều vòng quay của nhiều bước thử nghiệm khác nhau.

## 2.4. Xây dựng cơ sở dữ liệu hướng đối tượng

Một số điều liên quan đến lược đồ CSDL hướng đối tượng xuất hiện khi phân lớp đối tượng như tổng quát hóa, đặc biệt hóa và gộp. Các vấn đề này xảy ra song song, người ta không trình bày quá trình chuyển từ mô hình thực thể E/R sang mô hình hướng đối tượng bởi lẽ không có điểm tương tự giữa một bên là thực thể và mối quan hệ, một bên là lớp đối tượng. Hơn nữa, mô hình E/R hầu như hướng dữ liệu chứ không biết về hành vi của thực thể khi dựa vào hạng của quan hệ mà thực thể tham gia.

Việc chuyển hóa mô hình đối tượng ngữ nghĩa sang CSDL hướng đối tượng được thực hiện như sau:

- Đối với mỗi đối tượng ngữ nghĩa, người ta tạo một lớp đối tượng.

- Khi có quan hệ cha con giữa hai đối tượng ngữ nghĩa, người ta thực hiện liên kết thừa kế giữa lớp đối tượng thể hiện kiểu con và lớp đối tượng thể hiện kiểu cha.

- Tiến hành gộp và liên kết các đối tượng ngữ nghĩa về kiểu đối tượng để được các lớp đối tượng gộp.

Giữ tính toàn vẹn về hạng giữa các đối tượng ngữ nghĩa, duy trì việc hợp nhóm của các đối tượng ngữ nghĩa và phân rã kiểu con bằng các phương thức.

Vong đời phát triển hướng đối tượng gồm 3 pha:

*- Pha phân tích:* Người ta phát triển mô hình ứng dụng trên thế giới thực, chỉ ra các thuộc tính quan trọng. Cần xác định các khái niệm trừu tượng về mô hình xuất phát từ lĩnh vực ứng dụng và các mô tả về hệ thống. Người ta xác định các hành vi chức năng của hệ thống, độc lập với môi trường sẽ thực hiện hướng đối tượng.

- *Pha thiết kế*: Cho phép xác định cách thức sẽ thực hiện mô hình phân tích hướng đối tượng.

- *Pha thực hiện*: Người ta xác định và khảo sát tất cả các kết quả của việc thực hiện thiết kế. Tất cả các quyết định thiết kế chiến lược, như là cách thực tích hợp hệ quản trị CSDL, cách liên lạc giữa các đối tượng, cách xử lý sai sót v.v. đều được triển khai. Tiếp theo người ta tích hợp các quyết định này vào mô hình thiết kế ban đầu, rồi hình thức hóa mô hình thiết kế để các đối tượng có thể tương tác với các đối tượng khác theo các kịch bản.

# CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG CHO HỆ THỐNG QUẢN LÝ THƯ VIỆN

## 3.1. Các yêu cầu của bài toán

- Hệ thống Thư viện quản lý các tư liệu, sách, báo, gọi chung là các tài liệu và cho các độc giả - những người đã đăng ký làm thẻ bạn đọc của Thư viện được mượn các tài liệu đó.

- Căn cứ vào nhu cầu của bạn đọc và tình hình xuất bản của các nhà xuất bản, hệ thống xử lý việc đặt mua những tài liệu mới, đối với những đầu sách phổ thông thì có thể mua nhiều cuốn. Đồng thời, những tài liệu đã hết hạn sử dụng, hoặc đã bị hư hỏng thì phải loại bỏ khỏi Thư viện.

- Thủ thư là nhân viên của Thư viện làm nhiệm vụ giao tiếp với đọc giả và công việc của họ được hệ thống hỗ trợ.

- Độc giả có thể đặt mượn sách, báo mà hiện tại Thư viện chưa có. Thư viện sẽ đặt mua theo yêu cầu và thông báo cho độc giả biết. Phiếu đặt mượn sẽ bị huỷ bỏ khi độc giả đã được mượn tài liệu hoặc trực tiếp được độc giả huỷ bỏ, hoặc bị thủ thư huỷ bỏ do điều kiện nào đó không được đáp ứng (như không mua được chẳng hạn), v.v.

- Hệ thống dễ dàng tạo lập, cập nhật, xoá bỏ thông tin về các mục tài liệu, độc giả, người mượn và phiếu đặt trước, v.v.

- Hệ thống có thể chạy được trên các môi trường phổ dụng UNIX, Windows, OS/2, v.v. với giao diện đồ hoạ thân thiện cho người sử dụng.

- Hệ thống có tính mở, dễ tương thích, dễ bổ sung, thay đổi theo yêu cầu của người sử dụng.

## 3.2. Phân tích các yêu cầu

Phân tích các yêu cầu để định nghĩa các ca sử dụng – các chức năng, dịch vụ của hệ thống và các tác nhân ngoài.

### 3.2.1. Các tác nhân ngoài

**+***Nhân viên thư viện (thủ thư)*: Người sử dụng hệ thống, được hệ thống hỗ trợtrong công việc.

**+** *Bạn đọc*: Những người được quyền đọc, mượn tài liệu của Thư viện.

**+** *Nhà xuất bản:* Nhà xuất bản cung ứng sách, báo mới theo yêu cầu của Thư viện.

*Lưu ý:* Bạn đọc không được phép mượn và trả lài liệu trực tiếp với hệ thống mà phải thông qua các thủ thư của Thư viện.

### 3.2.2. Các ca sử dụng (chức năng chính)

Dựa vào các yêu cầu mô tả nêu trên và dựa vào việc khảo sát hoạt động của hệ thống thư viện, chúng ta thấy hệ thống quản lý thư viện có những chức năng chính như sau:

1. *Đặt mượn*: Bạn đọc có thể đặt mượn những tài liệu mới chưa có trong Thư viện hoặc những tài liệu đã có trong Thư viện nhưng đã cho mượn hết.

2. *Huỷ phiếu đặt mượn:* Bạn đọc hoặc thủ thư có thể huỷ bỏ phiếu đặt trước vì tài liệu mới không mua được hay vì những lý do nào đó không đáp ứng được. Nếu thủ thư huỷ phiếu đặt trước thì phải thông báo cho bạn đọc.

3. *Quản lý bạn đọc:* Bạn đọc muốn đọc hay mượn tài liệu của Thư viện thì phải đăng ký và làm thẻ đọc, nghĩa là hệ thống có chức năng *bổ sung bạn đọc.* Trường hợp ngược lại, hệ thống cũng có thể *cập nhật lại* các thông tin về bạn đọc hoặc khi bạn đọc đã trả hết tài liệu và không còn là bạn đọc thì hệ thống *loại bỏ* họ ra khỏi danh sách.

4. *Cho mượn tài liệu:* Những người có trong danh sách bạn đọc được mượn những tài liệu có trong Thư viện theo phiếu mượn và nếu những đầu tài liệu đó còn trong thư viện.

5. *Nhận trả tài liệu:* Nhận lại những tài liệu mà bạn đọc trả.

6. *Nhắc trả tài liệu:* Hệ thống gửi thông báo nhắc trả những tài liệu mà bạn đọc giữ quá hạn hay những tài liệu có bạn đọc đặt mượn.

7. *Đặt mua tài liệu:* Dựa vào yêu cầu của bạn đọc và thông tin xuất bản để đặt mua bổ sung các tài liệu mới.

*8. Quản lý hạng mục: Hạng mục* là một đầu sách, là một bản tài liệu cụ thể. Khi có những tài liệu mới được nhập về thì thực hiện *bổ sung hạng mục*, khi tài liệu bị thất thoát hay hết hạn sử dụng thì *cập nhật* hoặc *loại bỏ chúng.*

9. *Quản lý đề mục: Đề mục* là định danh của tài liệu, như tên sách, tác giả, nhà xuất bản, năm xuất bản và thường một đề mục sách có nhiều bản (hạng mục – item).

Trong thư viện, người ta phân biệt *đề mục* với *hạng mục. Hạng mục* xác định một một đầu sách, một bản tài liệu cụ thể, còn *đề mục* xác định tên gọi của những cuốn sách đó. Ví dụ, đề mục “*Lập trình hướng đối tượng với Java, Đoàn van Ban, Thống kê, Hà Nội 2003*” có 5 cuốn sách (hạng mục) được mã số khác nhau trong thư viện.

### 3.2.3. Xây dựng biểu đồ ca sử dụng

Trong những hệ thống lớn, phức tạp, chúng ta nên gộp một số chức năng thành một gói (khối chức năng) các ca sử dụng. Trong hệ thống quản lý thư viện, mỗi chức năng được xem như là một ca sử dụng, trong đó có những ca sử dụng lại có thể phân tách nhỏ thành ca sử dụng con như *quản lý tài liệu* được phân thành *bổ sung,* và *loại bỏ tài liệu, quản lý bạn đọc* được phân thành *bổ sung* và *loại bỏ bạn đọc*. Biểu đồ của hệ thống thư viện được xây dựng như hình 3-1.

|  |
| --- |
|  |

**Hình 3.1: *Biểu đồ ca sử dụng của thư viện***

Bước tiếp theo là thảo luận với các nhân viên thư viện, trao đổi với bạn đọc để hiểu rõ hơn từng qui trình nghiệp vụ của những ca sử dụng trên. Đối với những ca sử dụng phức tạp nên xây dựng các kịch bản mô tả hoạt động của chúng. Ví dụ: ca sử dụng “*Cho mượn tài liệu”* được mô tả theo kịch bản thực hiện như sau:

#### 3.2.3.1. Nếu bạn đọc không có phiếu đặt mượn trước

- Dựa vào phiếu yêu cầu để tìm *đề mục* của tài liệu mà bạn đọc muốn mượn

- Xác định bạn đọc

- Nếu có một hạng mục của đề mục đó thì

+ Cho mượn

+ Ghi lại thông tin về người mượn: đề mục, thời hạn trả và cập nhật lại hạng mục.

- Ngược lại, từ chối cho mượn.

#### 3.2.3.2. Nếu bạn đọc có phiếu đặt mượn trước

- Dựa vào phiếu mượn để xác định bạn đọc

- Xác định đề mục

- Nếu có một hạng mục của đề mục thì

+ Cho mượn

+ Ghi lại thông tin về người mượn: đề mục, thời hạn trả và cập nhật lại hạng mục.

- Ngược lại, từ chối cho mượn.

Tương tự, xây dựng kịch bản cho những trường hợp còn lại.

## 3.3. Phân tích miền bài toán – xây dựng mô hình khái niệm

Nhiệm vụ chính của giai đoạn này là phân tích miền hoạt động của hệ thống để xác định những lớp đối tượng chính và mối quan hệ của chúng.

### 3.3.1. Xác định các lớp đối tượng

Dựa vào những kết quả khảo sát, mô tả bài toán, những kịch bản và dựa vào biểu đồ ca sử dụng để xác định các lớp. Trong thư viện có nhiều loại tài liệu khác nhau và thường được phân thành các loại sách, báo ngày, tạp chí, v.v.

+ Lớp**DauMuc:** lớp cơ sở cho những loại tài liệu khác nhau kế thừa. Lớp này có những thuộc tính chung như: *tên gọi* (*tenGoi*), *số lượng* (*soLuong*) và hàm thành phần chung như: *tìm kiếm (timKiem()), tạo lập (taoLap()), loại bỏ (loaiBo()), v.v.*

+ Lớp **Sach**: lớp con của **DauMuc**được bổ sung thêm những thuộc tính mô tả về những cuốn sách như: *tác giả (tacGia), nhà xuất bản (nhaXB), năm xuất bản (namXB), số trang (soTrang), thời hạn cho mượn (thoiHan)* mặc định như là 21 ngày*, v.v.*

+ Lớp **BaoNgay**: lớp con của **DauMuc**được bổ sung thêm những thuộc tính mô tả về các loại báo phát hành hàng ngày như: *ngày phát hành (ngayPhatHanh), thời hạn cho mượn (thoiHan)* mặc định như là 1 ngày*, v.v.*

+ Lớp **TapChi**: lớp con của **DauMuc**được bổ sung thêm những thuộc tính mô tả về các loại tạp chi phát hành hàng ngày như: *số* (*so*)*, tập (tap), thời hạn cho mượn (thoiHan)* mặc định như là 7 ngày*, v.v.*

+ Lớp **HangMuc**: lớp đối tượng cho các đầu sách, từng bản tài liệu là *bản copy* của **DauMuc** và có thuộc tính *mã số (maSo)* và các hàm như *tìm theo đề mục (timDeMuc), tìm theo mã số (timMaSo),* ngoài ra còn có những hàm để *tạo lập, huỷ bỏ đối tượng* giống như các lớp khác.

+ Lớp **PhieuDatMuon**: phiếu mượn ghi lại những thông tin về những tài liệu đặt trước. Lớp này có thuộc tính *ngày đặt mượn (ngayDat), tài liệu đặt mượn (taiLieu)* và liên kết với lớp **DauMuc**.

+ Lớp **NguoiMuon**: lớp lưu lại những thông tin về người mượn, những tài liệu đã đặt mượn và đã được mượn. Lớp này có quan hệ với lớp **DauMuc**, lớp **ThongTinBanDoc**.

+ Lớp **ThongTinBanDoc**: tập các đối tượng lưu trữ các thông tin về bạn đọc, những người đã đăng ký làm thẻ bạn đọc của Thư viện. Thông tin lưu trữ bao gồm: *họ tên (tenGoi), địa chỉ (diaChi)*, số thẻ (*soThe*)*,* v.v. Lớp này có quan hệ kết hợp với lớp **NguoiMuon, lớp DatMuon**.

+ Lớp **PhieuDatMua**: dựa vào những thông tin xuất bản và yêu cầu của bạn đọc để đặt mua những tài liệu mới. Lớp này có quan hệ kết hợp với **DauMuc**, lớp **PhieuDatMuon** và có thuộc tính *tên nhà xuất bản (nhaXB),tài liệu, (taiLieu), sốlượng đặt mua (soLuongMua), ngày đặt mua (ngayDatMua).*

+ Lớp **PhieuNhacTra**: gồm những thuộc tính về tài liệu mà bạnđọc mượn đã quá hạn như: *tài liệu (taiLieu), người mượn (nguoiMuon), v.v.*

### 3.3.2. Xây dựng biểu đồ lớp

Dựa vào những kết quả phân tích như trên, chúng ta có biểu đồ lớp như sau:

|  |
| --- |
|  |

**Hình 3. 2: *Biểu đồ lớp của hệ thống thư viện***

## 3.4. Thiết kế chi tiết

Mục đích của giai đoạn này là:

+ Mở rộng và chi tiết hoá các đối tượng tác nghiệp đã được xác định ở các giai đoạn trước.

+ Mô tả các lớp kỹ thuật thực hiện các chức năng giao diện (gói *giao diện)* và tổ chức lưu trữ (gói *CSDL).*

### 3.4.1. Thiết kế chi tiết gói CSDL

|  |
| --- |
|  |

**Hình 3.5: *Các gói trong hệ thống thư viện***

Gói *CSDL* phải cung cấp các dịch vụ đảm bảo lưu trữ nhất quán các đối tượng. Khi thiết kế gói *CSDL* có thể:

+ Dựa vào những hệ QT CSDL hướng đối tượng có trên thị trường, như O2, ObjectStore, Jasmin, v.v.

+ Dựa vào CSDL quan hệ và thực hiện biến đổi các bảng thành đối tượng.

Trong những hệ thống tương đối nhỏ như hệ thống quản lý thư viện có thể chọn giải pháp đơn giản nhất như sau:

Lưu trữ các đối tượng thành các tệp trên đĩa:

+ Các phép toán phục vụ lưu trữ như: save(), update(), delete(), find(), read() được cung cấp cho NSD và họ không cần biết chi tiết.

+ Mọi thực hiện liên quan đến việc xử lý dữ liệu của đối tượng luôn đảm bảo được tính nhất quán, được thực hiện thông qua một lớp được gọi là Persistent.

+ Lớp **Persistent** là lớp cơ sở trừu tượng để cho các lớp cần có các đối tượng nhất quán kế thừa. Các lớp con của nó phải cài đặt thêm những thao tác như: *write(), read()* cho từng đối tượng cụ thể.

+ Một thành tố quan trọng nữa tham gia vào việc bảo vệ tính nhất quán trong hệ thống là thuộc tính **id**, được sử dụng để truy cập (tham chiếu) tới chính các đối tượng yêu cầu trong hệ thống. Các đối tượng được xác định bằng định danh ID và thông qua lệnh việc khởi tạo tự động cho tất cả các thực thể

### 3.4.2. Thiết kế chi tiết gói các đối tượng tác nghiệp

Nhiệm vụ chính của giai đoạn này là thiết kế các thành phần của các lớp tác nghiệp và chỉ ra cách cài đặt chúng như thế nào.

Cách thực hiện như sau:

+ Tất cả các lớp đối tượng tác nghiệp cần phải lưu trữ đều kế thừa lớp **Persistent** và cài đặt bổ sung những phép toán cần thiết như: *read(), save()* như đã nêu ở gói *CSDL.*

+ Những phương thức đã phân tích ở trên có thể chuyển sang mô hình thiết kế chi tiết hoặc một số có thể thay đổi, như đổi tên cho phù hợp hơn và thực hiện chi tiết hoá để tiện lợi cho việc cài đặt ở giai đoạn sau.

+ Tất cả các hàm (phương thức) phải được định nghĩa chính xác về tên gọi, kiểu trả lại, đối số và gán giá trị ban đầu nếu cần.

Biểu đồ lớp được thiết kế chi tiết như trong hình 3.6.

|  |
| --- |
|  |
|  |
|  |

**Hình 3.6: *Biểu đồ lớp chi tiếp***

### 3.4.5. Kết quả tạo được trong bảng cơ sở dữ liệu

Từ việc sử dụng ngôn ngữ lập trình java, kết hợp với các công cụ hỗ trợ thiết kế cơ sở dữ liệu tự động, sau khi thiết kế xong chúng em tạo được các cấu trúc bảng theo ý muốn

Bảng dạng kế thừa: Bảng chung là DauMuc, các dữ liệu kế thừa Sach, Tapchi, Baongay

|  |
| --- |
|  |
| Bảng dạng kế thừa: Bảng chung là Phieu, các dữ liệu kế thừa PhieuMuon, PhieuMua, PhieuNhacTra |
|  |

Bảng cơ sở dữ liệu dạng độc lập

Mặc dù cả 2 class đều kế thừa từ BaseEntity class nhưng chỉ định tạo 2 bảng độc lập

|  |
| --- |
|  |
|  |
|  |

# KẾT LUẬN

Thông qua bài tập này chúng em hiểu được những khái niểm về hệ quản trị cơ sở dữ liệu hướng đối tượng, hiểu được phương pháp thiết kế cũng như áp dụng vào bài toán thực tế để thiết kế ra được một hệ thống quản lý thư viện chuẩn. Tận dụng được những thế mạnh của hệ quản trị cơ sở dữ liệu hướng đối tượng, phương pháp thiết kế và ngôn ngữ lập trình hướng đối tượng để đáp ứng yêu cầu bài toán cụ thể.

Cùng với việc đó chúng em cũng đã tổng hợp đánh giá và biết được trong việc sử dụng cơ sở dữ liệu hướng đối tượng vẫn có những hạn chế như:

**Schema thay đổi**: Trong một RDBMS sửa đổi giản đồ cơ sở dữ liệu bằng cách tạo ra, cập nhật hoặc xóa bảng thường là độc lập với ứng dụng thực tế. Trong một ứng dụng dựa trên OODBMS sửa đổi giản đồ bằng cách tạo ra, cập nhật hoặc sửa đổi một lớp persistent thường có nghĩa là thay đổi phải được thực hiện cho các lớp khác trong ứng dụng tương tác với các trường hợp của lớp đó. Điều này thường có nghĩa là tất cả thay đổi lược đồ trong một OODBMS sẽ liên quan đến một hệ thống đều phải biên dịch lại toàn bộ. Cũng cập nhật tất cả các đối tượng tự trong cơ sở dữ liệu có thể mất một thời gian dài của thời gian tùy thuộc vào kích thước của cơ sở dữ liệu.

**Ngôn ngữ phụ thuộc**: An OODBMS thường được gắn với một ngôn ngữ cụ thể thông qua một API cụ thể. Điều này có nghĩa rằng dữ liệu trong một OODBMS thường chỉ truy cập từ một ngôn ngữ cụ thể sử dụng một API cụ thể, mà thường là không phải là trường hợp với một RDBMS.

Từ đó sẽ làm cơ sở để lựa chọn cũng như tìm ra được những cách thức, công nghệ phù hợp để quết định phần thiết kế cơ sở dữ liệu của mình sau này.

Sau khi hoàn thành bài toán chúng em thấy để thiết kế được một cơ sở dữ liệu chuẩn cần nhiều thời gian và cần có sự phân tích đúng đắn tin thần làm việc nhóm đặc biệt là những kiến thức truyền tải của thầy trong môn học này giúp chúng em hoàn thành môn học. Chúng em xin chân thành cảm ơn.