BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**TIỂU LUẬN CUỐI KỲ**

**ÁP DỤNG MẪU THIẾT KẾ TRONG QUẢN LÝ DANH SÁCH CÁC LOẠI SÁCH**

**MÔN: MẪU THIẾT KẾ PHẦN MỀM**

Ngành: **Công nghệ thông tin**

Chuyên ngành: **Kỹ thuật phần mềm**

Giảng viên hướng dẫn: **ThS. Lê Huỳnh Phước**

Sinh viên thực hiện: **Nguyễn Gia Bảo**

Mã số sinh viên: **22150450**

Lớp: **221407**

TP. Hồ Chí Minh, tháng 07 năm 2024

**Khoa Công nghệ thông tin**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: MẪU THIẾT KẾ PHẦN MỀM**

1. **Sinh viên:** Nguyễn Gia Bảo
2. **Tên đề tài**: Áp dụng mẫu thiết kế trong quản lý danh sách các loại sách
3. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên:…………………………………………………………………………….

Điểm số: ……….…… Điểm chữ: …………………………………………………..

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng … năm 20……*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |

**LỜI CẢM ƠN**

Em xin chân thành cảm ơn sâu sắc và được bày tỏ lòng biết ơn đến với Quý Thầy Cô của khoa Công Nghệ Thông Tin Trường Đại học Gia Định và đặc biệt đối với thầy **ThS. Lê Huỳnh Phước** đã tận tình hướng dẫn, đồng hành, động viên và chỉ bảo giúp tôi hoàn thành tốt bài báo cáo này cũng như truyền đạt những kiến thức bổ ích và rất quan trọng đối với quá trình đi làm của chúng em sau này.

Trong thời gian nghiên cứu đề tài tiểu luận vừa qua, em đã có nhiều cố gắng trong suốt quá trình thực hiện tiểu luận, tích cực trao đổi thông tin, sưu tầm, tham khảo tài liệu và học hỏi thêm kiến thức từ các Thầy Cô và các học viên khác, không chỉ nhận được những kiến thức đầy bổ ích về chuyên môn mà còn ở những lĩnh vực khác. Những trải nghiệm quý báu đó không chỉ giúp chúng em hoàn thành tốt bài báo cáo mà còn là hành trang quan trọng theo tôi trong suốt thời gian học tập và giúp tôi có thể tự tin bước vào đời làm việc sau này.

Do kiến thức và kinh nghiệm còn hạn chế cho nên bài báo cáo của em còn nhiều thiếu sót, kính mong được sự đánh giá, góp ý của thầy.

Cuối cùng, xin kính chúc thầy luôn luôn khỏe mạnh, vui vẻ và đạt được nhiều thành công cao trong công tác giảng dạy. Chúc trường Đại học Gia Định sẽ luôn là nền tảng vững chắc cho nhiều thế hệ sinh viên tiếp bước trên con đường học tập.

**Xin chân thành cảm ơn!**

Nguyễn Gia Bảo

**MỤC LỤC**

CHƯƠNG 1: LÝ THUYẾT VỀ MẪU THIẾT KẾ 5

**1.1. Tổng quan về đề tài 5**

**1.1.1. Giới thiệu đề tài 5**

**1.1.2. Lý do chọn đề tài 5**

**1.1.3. Mục tiêu của đề tài 5**

**1.1.4. Ý nghĩa của đề tài 5**

**1.1.5. Nhu cầu phần mềm 6**

**1.1.6. Giới hạn của đề tài 6**

**1.1.7. Đối tượng sử dụng 7**

**1.1.8. Giả định 7**

**1.2. Lý thuyết về mẫu thiết kế 8**

**1.2.1. Tổng quan về công nghệ lập trình Java 8**

**1.2.2. Tìm hiểu về kiến trúc MVC trong Java 11**

**1.2.3. Tổng quan về hệ quản trị cơ sở dữ liệu MySQL 13**

**1.2.4. Tổng quan về Mẫu thiết kế cho phần mềm 14**

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH 16

**2.1. Cơ sở dữ liệu 16**

**2.1.1. Tạo bảng 16**

**2.1.2. Dữ liệu bảng 16**

**2.1.3. Nhập thông tin vào bảng 17**

**2.2. Áp dụng mẫu MVC Processor vào đề tài 17**

**2.3. Phân tầng các phần 19**

**2.3.1. Tầng Presentation 19**

**2.3.2. Tầng Domain 19**

**2.3.3. Tầng Persistence 19**

**2.4. Mối quan hệ giữa các phần 20**

**2.5. Quy trình các bước khi chạy chương trình 22**

CHƯƠNG 3: THỰC NGHIỆM, KẾT QUẢ CHƯƠNG TRÌNH 24

**3.1. Hiện danh sách các sách hiện đang có 24**

**3.2. Thêm sách mới vào database 24**

**3.3. Sửa thông tin sách có trong danh sách 26**

**3.4. Xóa sách đã thêm vào bảng 27**

**3.5. Tìm sách theo Mã Sách 28**

**3.6. Tìm nhà xuất bản của sách hiện có trong danh sách 29**

**3.7. Tính thành tiền sách 29**

**3.7.1. Tính thành tiền sách giáo khoa mới 29**

**3.7.2. Tính thành tiền sách giáo khoa cũ 30**

**3.7.3. Tính thành tiên sách tham khảo 31**

CHƯƠNG 4: KẾT LUẬN VÀ ĐÁNH GIÁ 32

**4.1. Kết quả đạt được 32**

**4.2. Cần cải thiện 32**

# CHƯƠNG 1: LÝ THUYẾT VỀ MẪU THIẾT KẾ

* 1. **Tổng quan về đề tài**

***1.1.1. Giới thiệu đề tài***

Trong thư viện X, việc quản lý danh sách các loại sách là một nhiệm vụ quan trọng để đảm bảo rằng thông tin về các sách giáo khoa và sách tham khảo được lưu trữ, cập nhật và truy xuất một cách chính xác và hiệu quả. Đề tài này tập trung vào việc thiết kế và phát triển một hệ thống phần mềm quản lý các loại sách với các thông tin chi tiết và các tính năng cần thiết để tính toán giá trị thành tiền của từng loại sách.

***1.1.2. Lý do chọn đề tài***

Nhu cầu quản lý một lượng lớn sách trong thư viện đòi hỏi phải có một hệ thống tự động để giảm thiểu sai sót và nâng cao hiệu quả công việc.

Hệ thống quản lý thông tin sách giúp thủ thư và người quản lý có thể đưa ra các quyết định kịp thời và chính xác liên quan đến việc mua sắm, phân loại, và xử lý sách.

Tính năng tự động tính toán thành tiền của sách giáo khoa và sách tham khảo giúp tiết kiệm thời gian và công sức cho người quản lý thư viện.

Hệ thống giúp lưu trữ và quản lý thông tin chi tiết về từng cuốn sách, từ đó hỗ trợ việc tra cứu và kiểm tra thông tin dễ dàng hơn.

* + 1. ***Mục tiêu của đề tài***

Xây dựng một hệ thống quản lý sách cho thư viện X với khả năng lưu trữ và xử lý thông tin về sách giáo khoa và sách tham khảo.

Thiết kế giao diện người dùng thân thiện và dễ sử dụng.

Tự động tính toán giá trị thành tiền của từng loại sách dựa trên các quy tắc cụ thể.

* + 1. ***Ý nghĩa của đề tài***

Thực tiễn: Giúp thư viện quản lý sách một cách hiệu quả, giảm thiểu sai sót và nâng cao chất lượng dịch vụ.

Giáo dục: Cung cấp một ví dụ thực tế về việc ứng dụng kiến thức về mẫu thiết kế phần mềm và lập trình trong việc giải quyết các vấn đề quản lý thực tế.

Công nghệ: Khám phá và áp dụng các công nghệ mới trong phát triển phần mềm, bảo mật dữ liệu, và giao diện người dùng.

* + 1. ***Nhu cầu phần mềm***

*a) Lưu trữ thông tin sách*

Thông tin chi tiết về sách giáo khoa: mã sách, ngày nhập, đơn giá, số lượng, nhà xuất bản, tình trạng.

Thông tin chi tiết về sách tham khảo: mã sách, ngày nhập, đơn giá, số lượng, nhà xuất bản, thuế.

*b) Tính toán thành tiền*

Sách giáo khoa mới: thành tiền = số lượng \* đơn giá.

Sách giáo khoa cũ: thành tiền = số lượng \* đơn giá \* 50%.

Sách tham khảo: thành tiền = số lượng \* đơn giá + thuế.

*c) Quản lý kho sách*

Theo dõi số lượng sách hiện có trong kho.

Cảnh báo khi số lượng sách đạt mức tối thiểu.

*d) Báo cáo và phân tích*

Cung cấp các báo cáo chi tiết về số lượng sách, giá trị thành tiền, và tình trạng sách.

Phân tích dữ liệu để hỗ trợ quyết định mua sắm và quản lý sách.

***1.1.6. Giới hạn của đề tài***

Phạm vi thông tin: Hệ thống chỉ quản lý các thông tin liên quan đến sách giáo khoa và sách tham khảo, không bao gồm các loại sách khác.

Phạm vi tính toán: Hệ thống chỉ tính toán giá trị thành tiền dựa trên các công thức đã cho, không bao gồm các yếu tố khác như chiết khấu hay khuyến mãi.

Phạm vi người dùng: Hệ thống chủ yếu dành cho nhân viên quản lý thư viện và thủ thư, không bao gồm các chức năng dành cho độc giả hoặc các bên thứ ba khác.

* + 1. ***Đối tượng sử dụng***

Nhân viên quản lý thư viện: Quản lý thông tin sách, nhập dữ liệu sách mới, và theo dõi tình trạng kho sách.

Thủ thư: Tra cứu thông tin sách, cập nhật tình trạng sách, và hỗ trợ quản lý kho sách.

Người quản lý: Theo dõi và phân tích dữ liệu, đưa ra các quyết định mua sắm và quản lý sách dựa trên các báo cáo từ hệ thống.

### *1.1.8. Giả định*

Dữ liệu chính xác: Thông tin nhập vào hệ thống là chính xác và đầy đủ.

Kỹ năng sử dụng máy tính: Nhân viên quản lý thư viện và thủ thư đều có kỹ năng cơ bản về sử dụng máy tính và các ứng dụng web.

Hạ tầng công nghệ: Thư viện có kết nối internet ổn định và đủ thiết bị phần cứng để sử dụng phần mềm quản lý.

## 1.2. Lý thuyết về mẫu thiết kế

### *1.2.1. Tổng quan về công nghệ lập trình Java*

*a) Môi trường lập trình*

Java là một ngôn ngữ lập trình đa mục đích, được phát triển bởi James Gosling và các nhà phát triển khác tại Sun Microsystems (hiện thuộc Oracle Corporation). Nó ra mắt lần đầu tiên vào năm 1995 và được thiết kế để làm việc trên nền tảng "Write Once, Run Anywhere" (WORA), có nghĩa là mã Java có thể được viết một lần và chạy trên nhiều nền tảng khác nhau mà không cần biên dịch lại mã nguồn. Java Development Kit (JDK - Bộ công cụ cho người phát triển ứng dụng bằng ngôn ngữ lập trình Java) là một tập hợp những công cụ phần mềm được phát triển bởi Sun Microsystems dành cho các nhà phát triển phần mềm, dùng để viết những applet Java hay những ứng dụng Java.

*b) Ngôn ngữ lập trình Java*

Java là một ngôn ngữ lập trình đa mục đích và phổ biến được phát triển bởi James Gosling và nhóm lập trình viên tại Sun Microsystems (nay là Oracle Corporation) vào những năm đầu của thập kỷ 1990. Nó là một ngôn ngữ lập trình hướng đối tượng, có khả năng chạy trên nhiều nền tảng khác nhau nhờ vào việc biên dịch mã nguồn thành bytecode và thực thi trên máy ảo Java (JVM - Java Virtual Machine).

*c) Đặc điểm chính của ngôn ngữ lập trình Java*

Độc lập nền tảng: Mã nguồn Java sau khi được biên dịch thành bytecode có thể chạy trên bất kỳ hệ điều hành nào hỗ trợ JVM. Điều này giúp Java thỏa mãn phương châm "Write Once, Run Anywhere" (WORA), cho phép tái sử dụng mã nguồn một cách hiệu quả trên nhiều hệ thống.

Hướng đối tượng: Java là ngôn ngữ lập trình hướng đối tượng (OOP), trong đó dữ liệu và mã được tổ chức trong các đối tượng có thuộc tính và phương thức riêng. OOP giúp quản lý mã dễ dàng, dễ hiểu và bảo trì.

Quản lý bộ nhớ tự động: Java có bộ thu gom rác (garbage collector) để tự động giải phóng bộ nhớ không sử dụng, giảm thiểu công việc quản lý bộ nhớ thủ công và giúp ứng dụng tránh hiện tượng rò rỉ bộ nhớ.

Đa luồng (Multithreading): Java hỗ trợ lập trình đa luồng, cho phép chạy nhiều tác vụ (luồng) đồng thời trong một ứng dụng. Điều này tận dụng hiệu quả nguồn lực hệ thống và cải thiện hiệu suất ứng dụng.

Thư viện chuẩn (Standard Library): Java đi kèm với một thư viện chuẩn phong phú (Java Standard Library) cung cấp nhiều lớp và phương thức để giải quyết các vấn đề phổ biến, như xử lý chuỗi, thao tác với tệp, xử lý số học, v.v.

An toàn và bảo mật: Java có cơ chế kiểm soát quyền truy cập (access control) và chế độ cách ly (sandbox) để giới hạn quyền truy cập vào hệ thống, giúp tăng cường bảo mật cho ứng dụng và người dùng.

Java là một ngôn ngữ hướng đối tượng, đa mục đích với các cú pháp rất giống với C và C++. Ban đầu thì đa số mọi người nghĩ là Java sẽ chủ yếu được sử dụng để lập trình nên những applet hay những chương trình nhỏ chạy trên các trình duyệt web, tuy nhiên đến giờ thì mọi người đã thay đổi quan điểm. Một số người vốn trước đây tin rằng applet chính là đất sống của Java thì nghĩ rằng Java đã chết do sự xuất hiện của các đoạn phim hoạt hình Flash. Nhưng Java đã thay đổi. Cái thời mà người ta nghĩ rằng ứng dụng chủ yếu của Java là làm các applet động trên các trang web đã qua. Ngày nay, Sun, IBM, BEA... và các công ty khổng lồ khác đã liên kết để phát triển Java thành một môi trường đa năng chứ không chỉ dừng lại là một thứ ngôn ngữ lập trình đa nền tảng nữa. Java đã có mặt ở khắp mọi nơi: từ những chiếc điện thoại di động nhỏ bé mang nhãn hiệu Nokia, Samsung, Motorola, Ericsson..., từ các thiết bị PDA dùng hệ điều hành Palm cho đến các con chíp điện tử dùng trên các tấm thẻ tín dụng, các thiết bị chẩn đoán và phân tích dùng trong y tế, khai thác năng lượng, điều khiển và quản lý thiết bị. ... từ các phần mềm trên server, các trang web động, cho đến các ứng dụng trên desktop. Bạn có biết rằng người máy Người tìm đường Sao Hỏa dùng phần mềm điều khiển bằng ngôn ngữ Java không? Những điều có lẽ còn cuốn hút bạn hơn khi bạn biết rằng ngay từ năm 1997, năm mà Java còn chưa tốt như bây giờ, Trung tâm Vũ Trụ NASA đã chính thức công nhận ngôn ngữ Java là ngôn ngữ chính được sử dụng để lập trình cho các thiết bị và phần mềm dùng cho Trung tâm.

Giờ đây, khi nhắc đến Java, người ta cần phải hiểu đó là: thứ nhất: đó là một môi trường phát triển và triển khai ứng dụng; thứ hai: đó là một ngôn ngữ lập trình toàn năng. Sự xuất hiện phổ biến của Flash không hề đe dọa đến Java. Rõ ràng với sự đầu tư của Sun và các công ty hỗ trợ Java khác, chỉ trong vòng 5 năm, nó đã trở thành một ngôn ngữ toàn năng nhất trong các ngôn ngữ lập trình được sử dụng trên thế giới hiện nay. Điều người ta quan tâm nhất ở Java là khả năng viết một lần chạy mọi nơi nghĩa là bạn có thể viết chương trình trên một máy tính cài Window, chạy chip của Intel nhưng chương trình đó vẫn chạy tốt và cho cùng một kết quả hoạt động khi chạy nó trên Macintosh hay Unix. Điều này là không tưởng đối với C, C++, VB... Khả năng chuyển đổi nền tảng, dễ phân phối, đa tầng, hướng đối tượng chính là những gì mà Java chứng tỏ nó ưu việt hơn các ngôn ngữ khác.

Với C, C++ tức là cha mẹ đẻ của Java thì điều này là rõ ràng. Với Visual Basic, ngôn ngữ lập trình quan trọng nhất của Microsoft, ngoài những điểm vừa nói ở trên, Java còn được giới lập trình chuyên nghiệp trên thế giới trong đó các hacker thế hệ thứ nhất đánh giá rằng đây là một ngôn ngữ có cú pháp và cấu trúc tốt hơn nhiều. Sử dụng Java bạn có thể dễ dàng mở rộng dự án lập trình của mình với quy mô không giới hạn, việc quản lý cũng hết sức dễ dàng trong khi đó Visual Basic với cấu trúc thiết kế không thực sự tốt, nó chỉ thích hợp với các dự án nhỏ, ít có nhu cầu mở rộng hay quản lý.

*d) Tính năng chính của Java bao gồm*

Độc lập nền tảng: Mã Java được biên dịch thành bytecode, một ngôn ngữ trung gian độc lập nền tảng, sau đó được thực thi bởi máy ảo Java (JVM - Java Virtual Machine) trên mọi hệ điều hành hỗ trợ JVM.

Cú pháp dễ đọc: Java sử dụng cú pháp dễ hiểu và giống với các ngôn ngữ lập trình khác như C++ và C#, giúp người lập trình dễ dàng học và sử dụng.

Quản lý bộ nhớ tự động: Java có hệ thống thu gom rác (garbage collector) giúp tự động thu hồi bộ nhớ không sử dụng, giảm thiểu việc phải quản lý bộ nhớ thủ công.

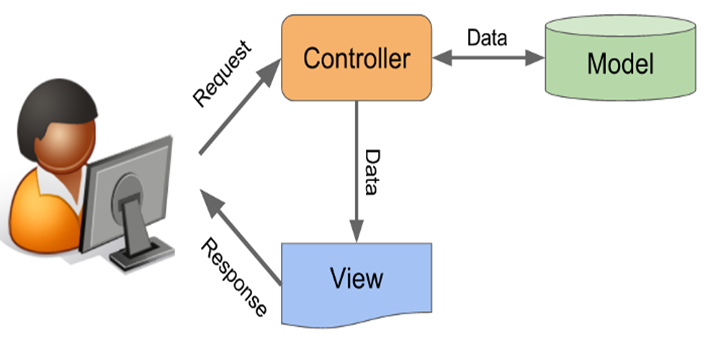
Hỗ trợ đa luồng: Java hỗ trợ lập trình đa luồng (multithreading), cho phép xử lý nhiều tác vụ cùng một lúc và tận dụng hiệu quả nguồn lực hệ thống.

Thư viện chuẩn (Standard Library): Java đi kèm với một thư viện chuẩn phong phú (Java Standard Library) cung cấp nhiều công cụ và lớp giúp giải quyết nhiều vấn đề phổ biến.

An toàn và bảo mật: Java có hệ thống kiểm soát quyền truy cập (access control) và chế độ cách ly (sandbox) để giới hạn quyền truy cập vào hệ thống, giúp tăng cường bảo mật cho ứng dụng.

Java được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm phát triển ứng dụng web, ứng dụng di động, phần mềm máy chủ, trò chơi, Internet of Things (IoT), và nhiều hệ thống phức tạp khác. Cộng đồng lập trình Java rất lớn và mạnh mẽ, với nhiều tài liệu, framework và thư viện hỗ trợ cho việc phát triển ứng dụng.

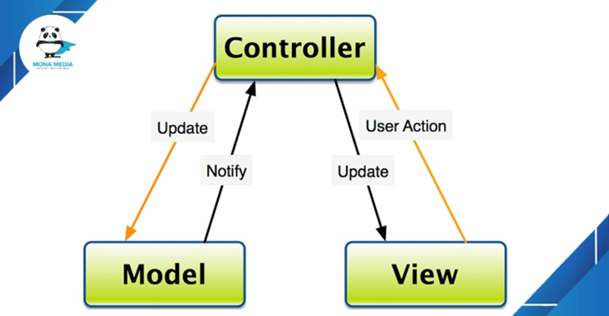
### *1.2.2. Tìm hiểu về kiến trúc MVC trong Java*

Kiến trúc MVC (Model-View-Controller) là một mô hình thiết kế phần mềm phổ biến trong lập trình ứng dụng. Nó giúp phân tách logic ứng dụng thành ba phần riêng biệt: Model (mô hình), View (giao diện) và Controller (bộ điều khiển). Đây là một cách tiếp cận có lợi cho việc phát triển ứng dụng có sự tách biệt giữa các thành phần, giúp dễ dàng bảo trì và mở rộng

MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Dưới đây là mô tả về từng thành phần trong kiến trúc MVC khi áp dụng trong Java:

* Model (Mô hình): Model đại diện cho dữ liệu và các logic xử lý dữ liệu. Đây có thể là lớp Java hoặc một tập hợp các lớp Java. Model thường không chứa bất kỳ logic hiển thị nào và chỉ tập trung vào quản lý dữ liệu và các phương thức để truy xuất, cập nhật và xử lý dữ liệu. Các thay đổi trong Model sẽ thông báo cho Controller để cập nhật giao diện View.
* View (Giao diện): View là thành phần dùng để hiển thị dữ liệu cho người dùng. Trong Java, nó có thể là một thành phần giao diện người dùng như JFrame, JPanel, JSP, hoặc các thành phần UI khác. View không chứa logic xử lý dữ liệu và không thực hiện các thay đổi dữ liệu trực tiếp. Nó chỉ hiển thị thông tin từ Model và chuyển các tương tác của người dùng đến Controller.
* Controller (Bộ điều khiển): Controller là thành phần trung gian giữa Model và View. Nhiệm vụ chính của Controller là tiếp nhận các yêu cầu từ View, xử lý các yêu cầu này bằng cách tương tác với Model, sau đó cập nhật lại View dựa trên kết quả. Controller có thể thực hiện kiểm tra hợp lệ, xử lý lỗi và quyết định cách mà Model sẽ phản ứng với các yêu cầu từ View.



Với cơ sở là kiến trúc MVC, ta có thể xây dựng các ứng dụng của mình, tránh được rất nhiều những vất vả khi bảo trì, thay đổi. Những thay đổi ở mỗi thành phần thường rất ít khi ảnh hưởng đến các thành phần khác.

### *1.2.3. Tổng quan về hệ quản trị cơ sở dữ liệu MySQL*

*a) Tìm hiểu về MySQL*

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. Các ứng dụng web lớn nhất như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

*b) Đặc điểm của MySQL*

Tốc độ truy cập nhanh, ổn định, dễ sử dụng.

Có tính khả chuyển, hoạt động trên nhiều nền tảng hệ điều hành.

Cung cấp hệ thống thư viện hàm lớn.

Khả năng bảo mật tốt.

Hoạt động như một client/server hoặc trong hệ thống nhúng.

Thích hợp cho các hệ thống trên Internet (Google, Youtube, Alcatel, …).

Được hỗ trợ bởi nhiều ngôn ngữ lập trình.

### *1.2.4. Tổng quan về Mẫu thiết kế cho phần mềm*

Mẫu thiết kế cho phần mềm (Design Patterns) là các giải pháp đã được chứng minh và chuẩn hóa để giải quyết các vấn đề phổ biến trong thiết kế phần mềm. Chúng không phải là mã nguồn cụ thể mà là các mô tả về cách giải quyết vấn đề mà có thể được áp dụng trong nhiều tình huống khác nhau.

*a) Định nghĩa*

Mẫu thiết kế cho phần mềm là các giải pháp thiết kế tái sử dụng để giải quyết các vấn đề lặp đi lặp lại trong phát triển phần mềm.

Mẫu thiết kế phần mềm thường được chia thành ba loại chính:

* Mẫu thiết kế kiến trúc (Architectural Patterns): Được sử dụng để tổ chức cấu trúc tổng thể của hệ thống phần mềm.
* Mẫu thiết kế cấu trúc (Structural Patterns): Được sử dụng để tổ chức và quản lý các mối quan hệ giữa các thành phần trong hệ thống.
* Mẫu thiết kế hành vi (Behavioral Patterns): Được sử dụng để quản lý và điều phối các tương tác giữa các đối tượng trong hệ thống.

*b) Các mẫu thiết kế phổ biến*

Mẫu thiết kế kiến trúc:

* MVC (Model-View-Controller): Tách biệt các khía cạnh của một ứng dụng thành ba thành phần: Model (dữ liệu), View (giao diện người dùng), và Controller (logic điều khiển).

Mẫu thiết kế cấu trúc:

* Adapter: Cho phép các đối tượng không tương thích có thể làm việc cùng nhau bằng cách chuyển đổi giao diện của một lớp thành một giao diện khác mà lớp đó mong đợi.
* Decorator: Cho phép thêm các chức năng bổ sung vào một đối tượng một cách linh hoạt mà không thay đổi cấu trúc của nó.

Mẫu thiết kế hành vi:

* Observer: Cho phép một đối tượng thông báo cho các đối tượng khác về các thay đổi trạng thái của nó, giúp giảm sự phụ thuộc giữa các đối tượng.
* Strategy: Cho phép lựa chọn thuật toán từ một tập hợp các thuật toán tại thời điểm chạy (runtime).

*c) Lợi ích của mẫu thiết kế*

Tái sử dụng mẫu: Giảm thời gian và công sức phát triển bằng cách tái sử dụng các giải pháp đã được chứng minh.

Dễ bảo trì: Cải thiện tính bảo trì và khả năng mở rộng của phần mềm bằng cách tạo ra các thiết kế rõ ràng và nhất quán.

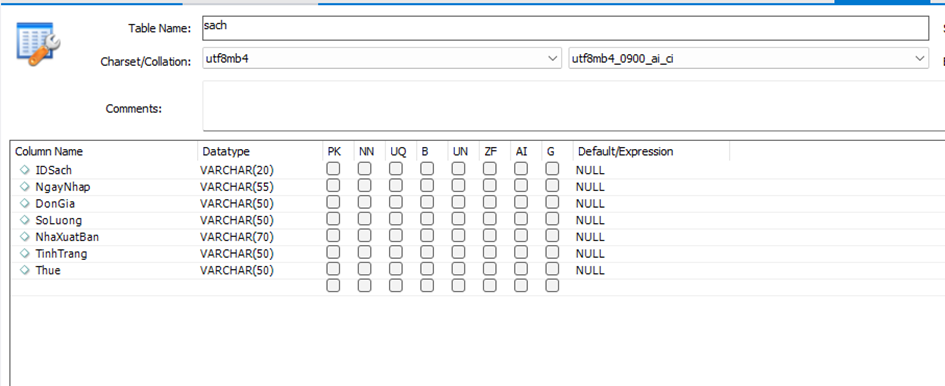
Giảm sự phức tạp: Giảm sự phức tạp của thiết kế phần mềm bằng cách cung cấp các giải pháp tổ chức và quản lý các thành phần trong hệ thống.

* Mẫu thiết kế phần mềm là các công cụ mạnh mẽ giúp giải quyết các vấn đề thiết kế phổ biến và cải thiện chất lượng phần mềm. Việc hiểu và áp dụng các mẫu thiết kế này không chỉ giúp tạo ra các hệ thống phần mềm hiệu quả và dễ bảo trì mà còn giúp tiết kiệm thời gian và công sức trong quá trình phát triển phần mềm. Trong đề tài " Quản lý danh sách các loại sách", việc áp dụng các mẫu thiết kế phù hợp sẽ giúp xây dựng một hệ thống quản lý sách hiệu quả, linh hoạt và dễ sử dụng.

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH

## 2.1. Cơ sở dữ liệu

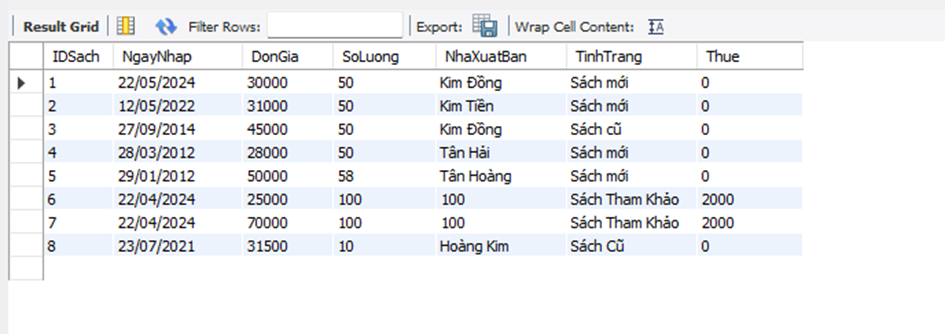
### *2.1.1. Tạo bảng*

Dựa theo yêu cầu đề bài, tạo ra bảng gồm các cột: IDSach, NgayNhap, DonGia, SoLuong, NhaXuatBan, TinhTrang, Thue

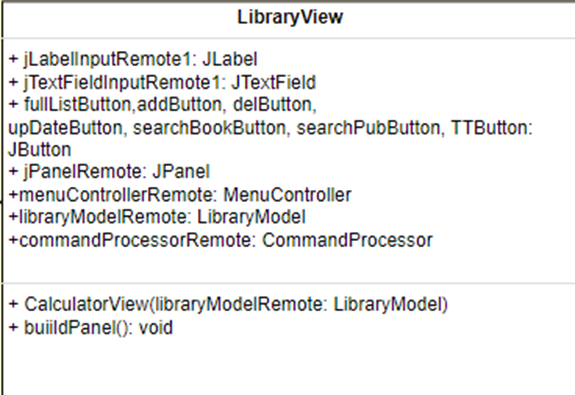
### *2.1.2. Dữ liệu bảng*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Trường** | **Mô Tả** | **Kiểu dữ liệu** | **Độ dài** |
| IDSach | Mã Sách | Varchar | 20 |
| NgayNhap | Ngày Nhập Sách | Varchar | 55 |
| DonGia | Đơn Giá | Varchar | 50 |
| SoLuong | Số Lượng | Varchar | 50 |
| NhaXuatBan | Nhà Xuất Bản | Varchar | 70 |
| TinhTrang | Tình Trạng | Varchar | 50 |
| Thue | Thuế | Varchar | 50 |

### *2.1.3. Nhập thông tin vào bảng*

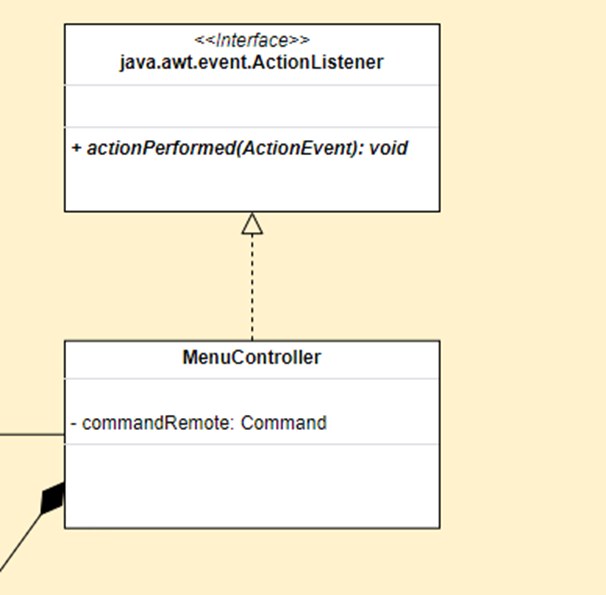
Nhập thông tin thông qua câu lệnh truy vấn INSERT có trong Sql để thêm thông tin các cột vào bảng. Kết quả bảng sau khi thêm:

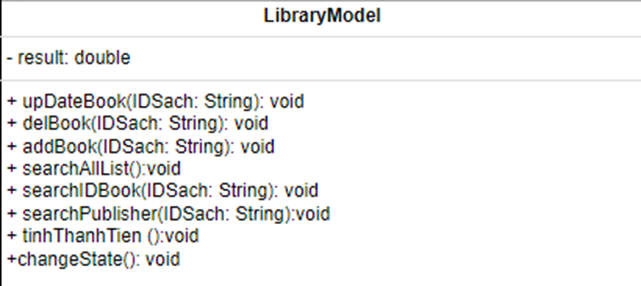
## 2.2. Áp dụng mẫu MVC Processor vào đề tài

Ở phần LibraryView, ta sẽ thêm các nút, tiêu đề để người dùng có thể tương tác

Ở phần MenuController, sẽ “lắng nghe” các tương tác của người dùng ở phần LibraryView khi ấn các nút và điền thông tin vào các TextBox thông qua Interface ActionListener.

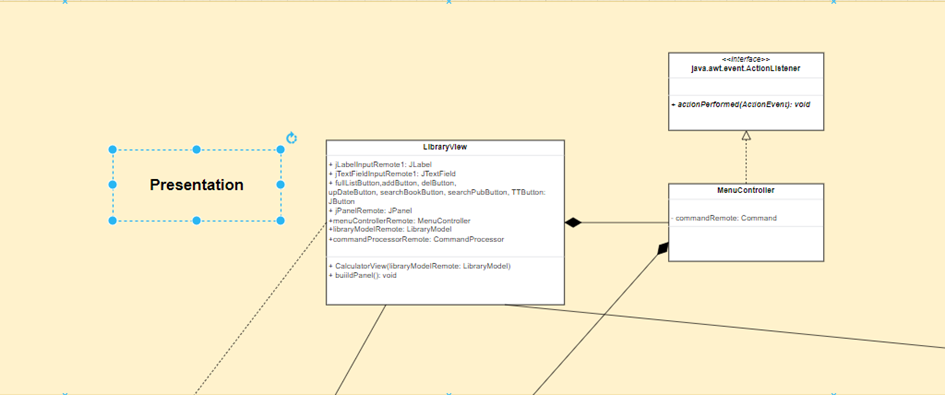
Ở actionPerformed ta sẽ cài các sự kiện của các nút sau khi “lắng nghe” tương tác của người dùng và thực hiện sự kiện đó



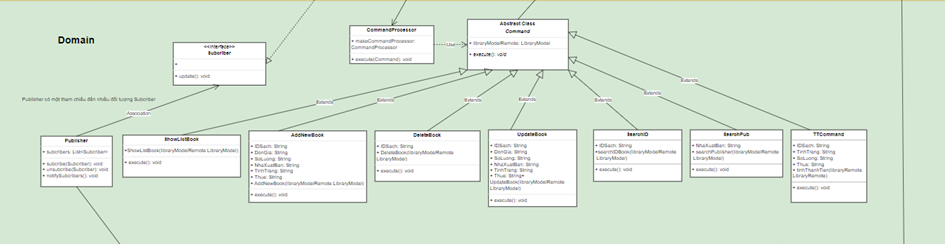
Ở LibraryModel là tầng xử lý nghiệp vụ ở đây sẽ xử lý các thông tin mà MenuController gửi đến để tính toán.

## 2.3. Phân tầng các phần

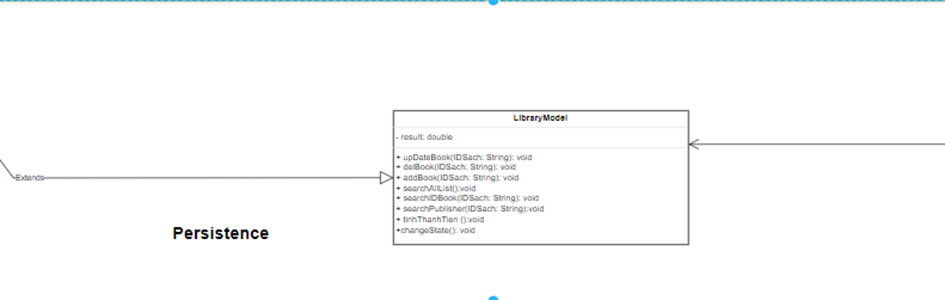
### *2.3.1. Tầng Presentation*

Ở tầng này tập trung chủ yếu vào giao diện đồ họa

### *2.3.2. Tầng Domain*

Ở tầng Domain tập trung vào xử lý nghiệp vụ được cài đặt theo yêu cầu đề bài

### *2.3.3. Tầng Persistence*

Tầng này tập trung vào kết nối, truy xuất dữ liệu trong Database

## 2.4. Mối quan hệ giữa các phần

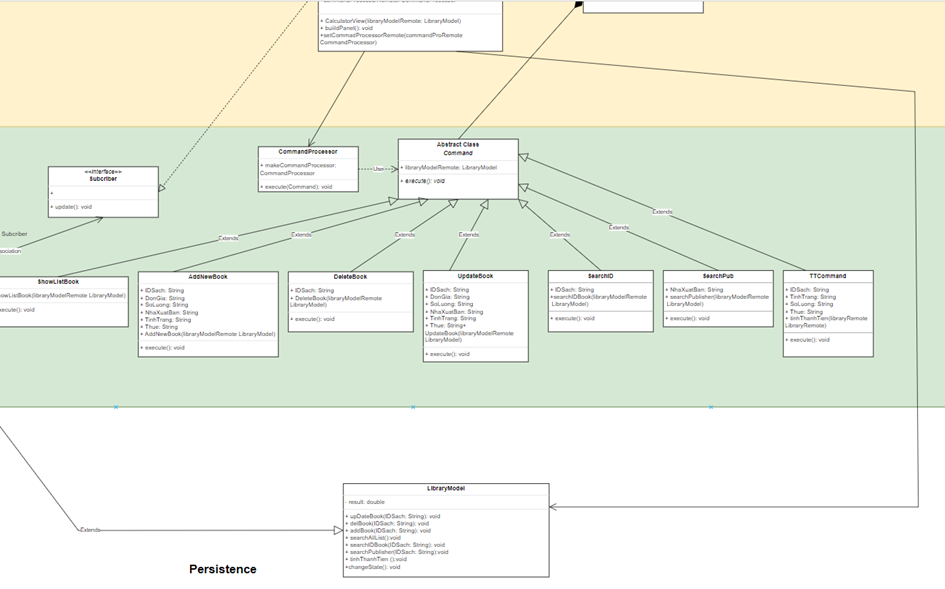
Mối quan hệ giữa LibraryView và MenuController: Composition vi lớp MenuController được tạo bên trong LibraryView

Mối Quan hệ giữa LibraryView và CommandProcessor: Association vì CommandProcessor được tạo trước và LibraryView đã sử dụng hàm setCommadProcessorRemote để truyền remote của CommandProcessor vào.

Mối quan hệ giữa MenuController và Command là: Composition vì đối tượng Command được tạo bên trong MenuController.

Mối quan hệ giữa CommandProcessor và Command là: Dependency vì CommandProcessor không lưu trữ lâu dài remote của Command ở biến đối tượng mà lưu ngắn hạn ở biến cục bộ trong hàm execute().

Mối quan hệ giữa LibraryView và LibraryModel là: Association vì đối tượng LibraryModel được tạo trước LibraryView và LibraryView truyền remote vào bằng constructor



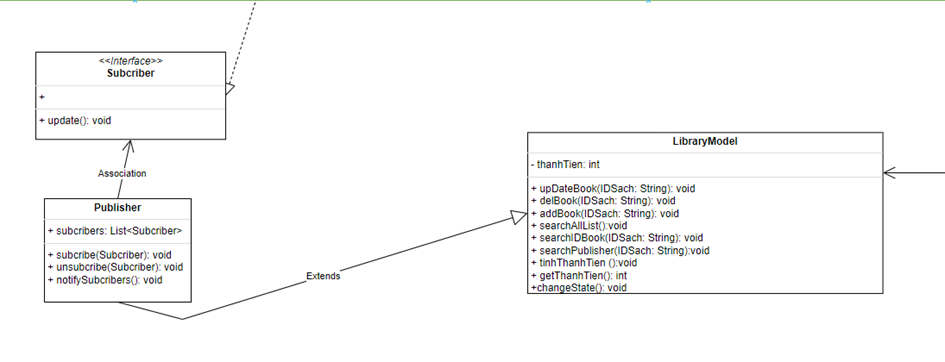
## 2.5. Quy trình các bước khi chạy chương trình

Sau khi nhập dữ liệu và tương tác ở lớp LibraryView, MenuController sẽ lắng nghe sự kiện và xử lý.

Đối tượng CommandProcessor được tạo trước, Class Command được tạo bên trong MenuController sau đó để đưa các nghiệp vụ cho các lớp con như: ShowListBook, AddNewBook, DeleteBook, UpdateBook, SearchID, SearchPub, TTCommand.

Các lớp con được tạo trước sẽ truyên Remote của LibraryModel trong MenuController vào và thực hiện câu lệnh execute trong các lớp con để tính toán sau đó sẽ truyền lại toàn bộ dữ liệu tính toán cùng với remote của LibraryModel vào lớp cha Command thông qua lệnh super().

Để tính toán xử lý thì lớp LibraryModel sẽ nhận được dữ liệu tính toán thông qua các lớp con của Command, các lớp con sẽ gửi số liệu để LibraryModel tính toán.

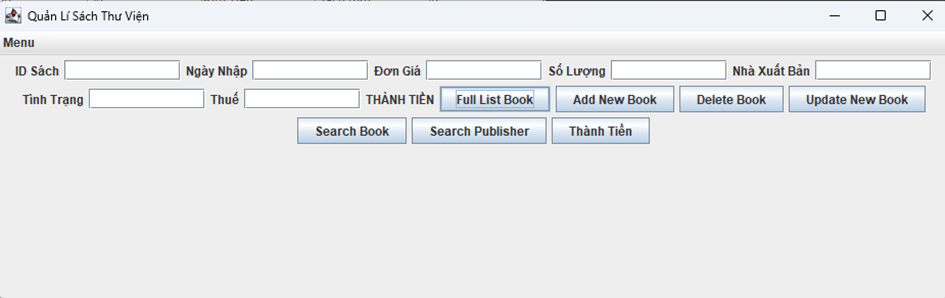
Ở Lớp Library sẽ có các hàm kết nối với database được tạo trong MySql từ đó vừa tính toán, thêm, sửa, xóa và hiển thị dữ liệu trong database.

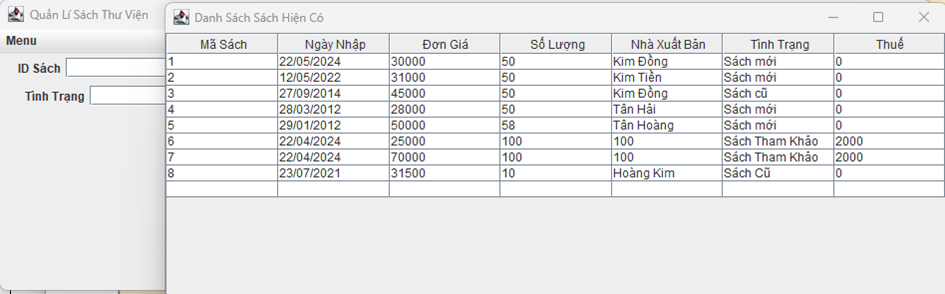
Library Model kế thừa class Publisher, khi tính toán xong thì sẽ có thông báo được truyền vào Publisher, và Publisher sẽ thông báo cho Subcriber thông qua hàm notifySubcriber().

Subcriber trong LibraryView đã được tạo và đăng ký trước với Publisher và trong LibraryView đã overdrive hàm update() của Subcriber vào nên khi Publisher cập nhật kết quả tính toán thì Subcriber sẽ cập nhật và báo kết quả lên hàm update mà LibraryView đã overdrive.

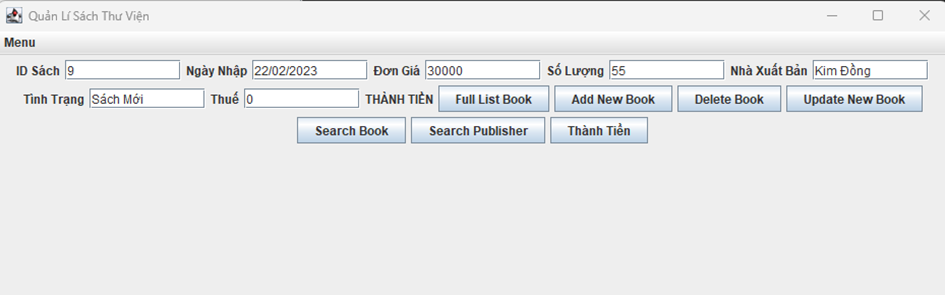
# CHƯƠNG 3: THỰC NGHIỆM, KẾT QUẢ CHƯƠNG TRÌNH

## 3.1. Hiện danh sách các sách hiện đang có

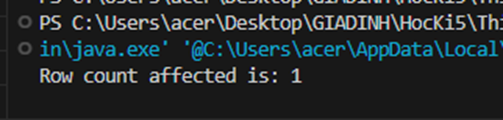
Chọn nút Full List Book

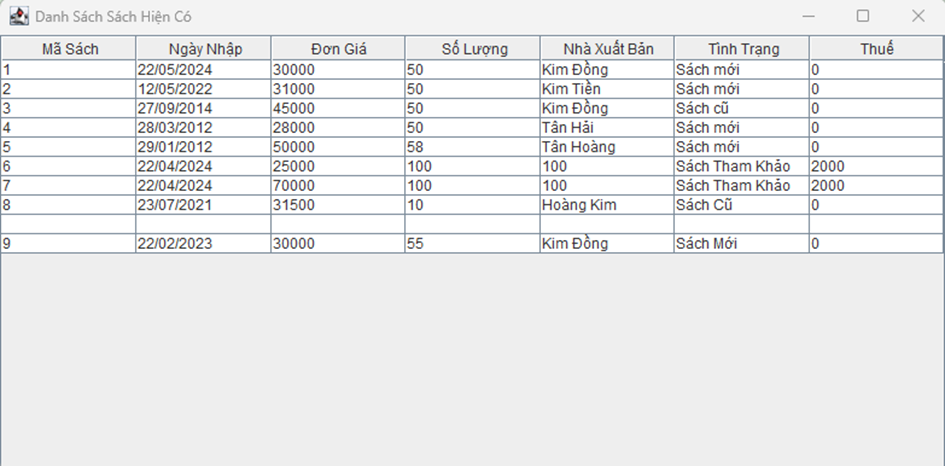
Xuất hiện bảng danh sách với thông tin tất cả sách hiện đang có sẵn

## 3.2. Thêm sách mới vào database

Ta sẽ nhập dữ liệu sách mã “9” có thông tin như sau:

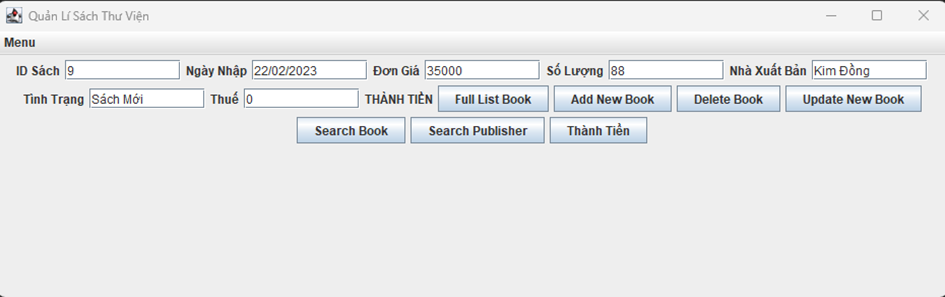
Ấn vào nút “Add New Book” để thêm sách vào database, và màn hình console sẽ hiện thông báo đã có 1 dòng thông tin được thêm



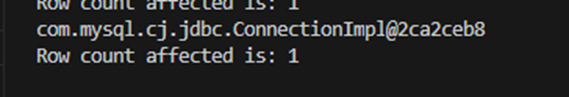
Ta sẽ vào kiểm tra lại danh sách để xem kết quả:

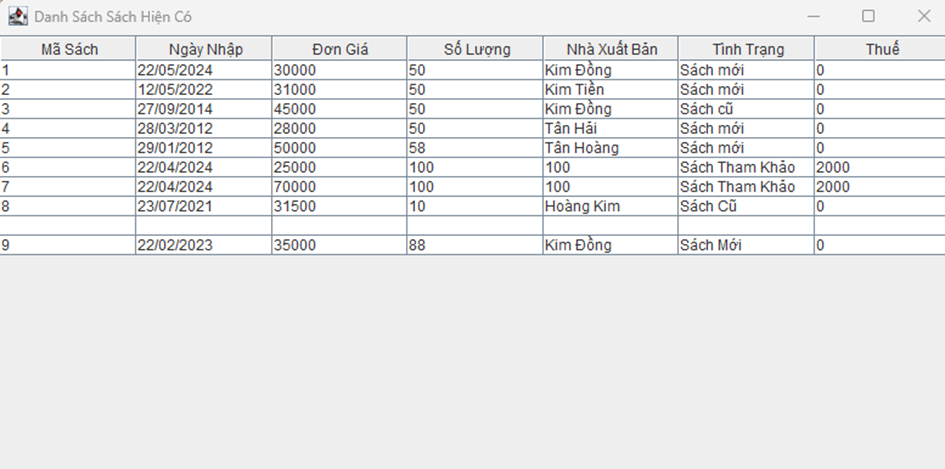
* Vậy là sách có mã “9” đã được thêm vào danh sách

**3.3. Sửa thông tin sách có trong danh sách**

Ta sẽ sửa đơn giá của sách có mã “9” thành 35000 và số lượng thành 88

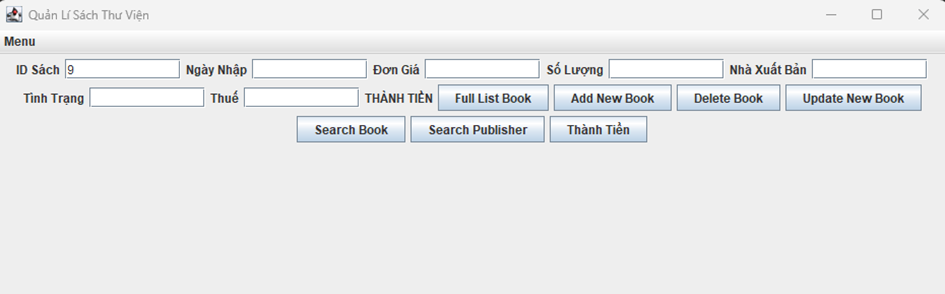
Ta sẽ ấn nút Update New Book để thêm thông tin mới vào, màn hình console vẫn sẽ hiện thông báo



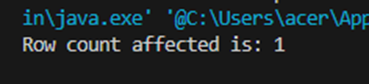
Và vào lại danh sách để kiểm tra lại thông tin

* Thông tin sách đã được thay đổi

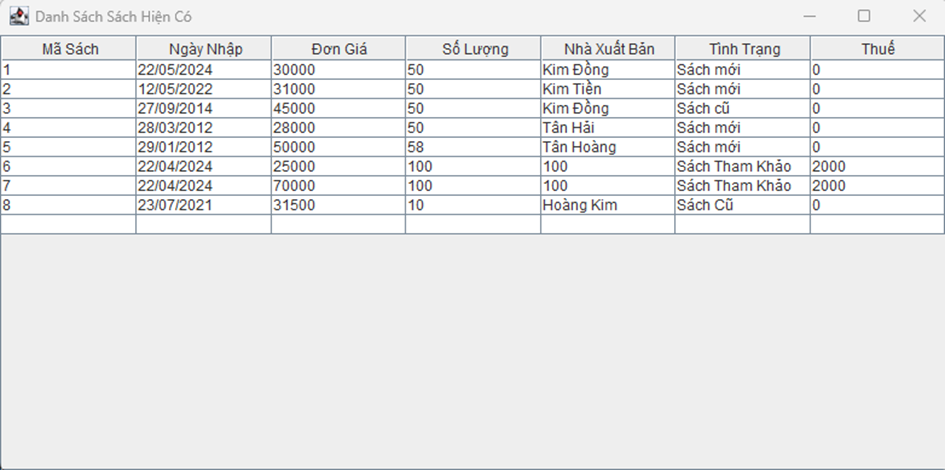
**3.4. Xóa sách đã thêm vào bảng**

Ta sẽ thực hiện xóa sách có mã sách là “9”, ta sẽ nhập trước mã sách vào form trước như sau

Ta sẽ ấn nút DeleteBook để xóa sách đang có trong danh sách, khi xóa thành công màn hình console sẽ hiện thông báo như sau:

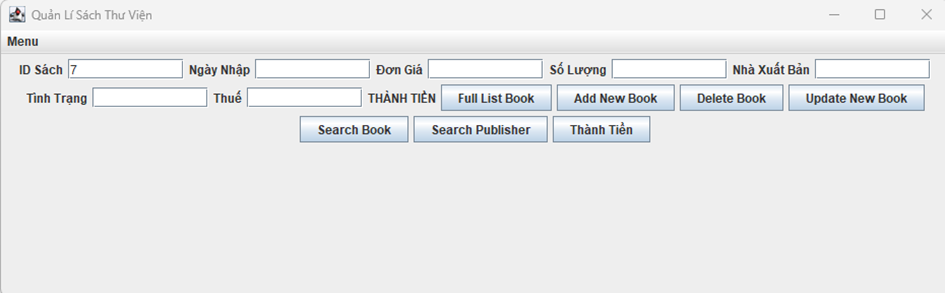


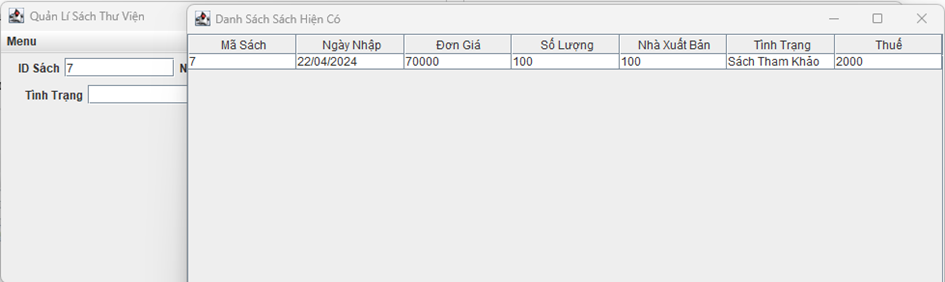
Ta sẽ vào lại danh sách để check lại thông tin



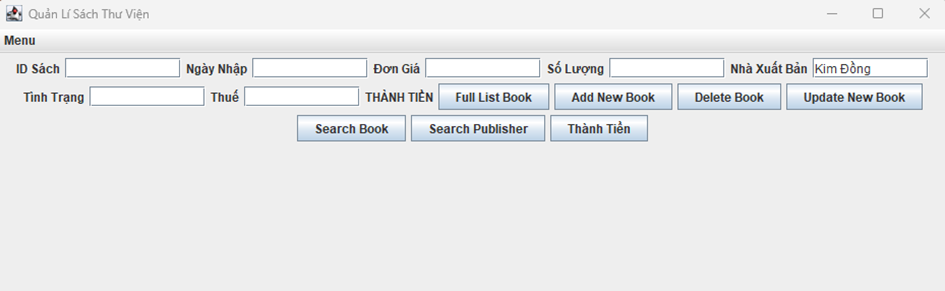
* Sách có mã “9” đã được xóa khỏi danh sách

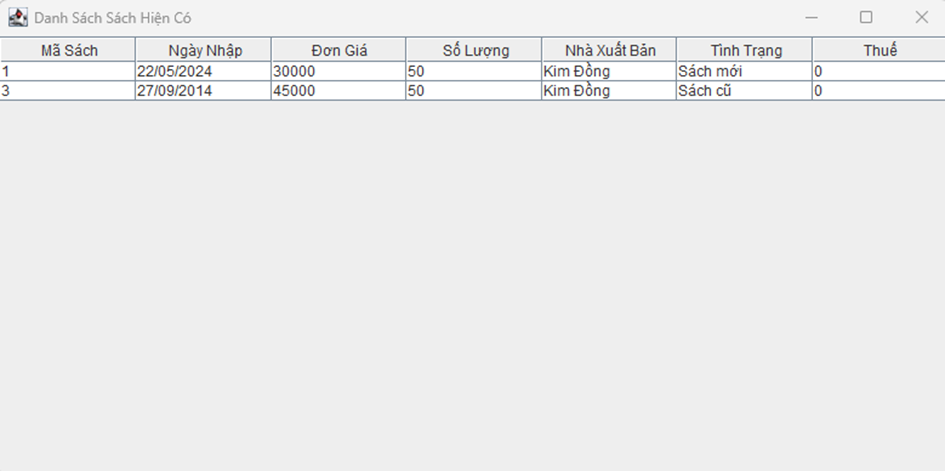
**3.5. Tìm sách theo Mã Sách**

Ta sẽ nhập trước mã sách vào form, ở trường hợp này ta sẽ điền mã sách vào form là “7”, sau đó ấn nút Search Book, màn hình console sẽ hiện lên thông báo khi thêm thành công

Kết quả: Thông tin sách cần tìm sẽ hiện lên

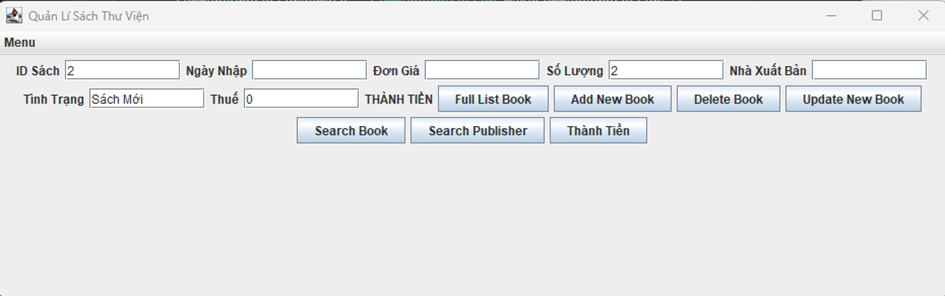
## 3.6. Tìm nhà xuất bản của sách hiện có trong danh sách

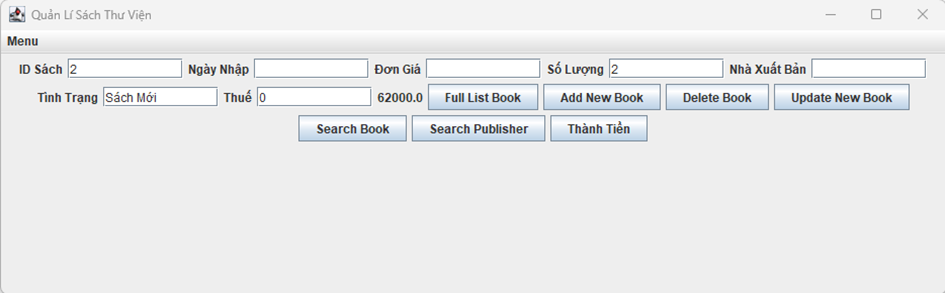
Ta sẽ nhập tên nhà xuất bản vào form trước, sau đó bấm nút Search Publisher để tìm sách có nhà xuất bản cần tìm

Kết quả: Đã tìm được thông tin sách của nhà xuất bản Kim Đồng

## 3.7. Tính thành tiền sách

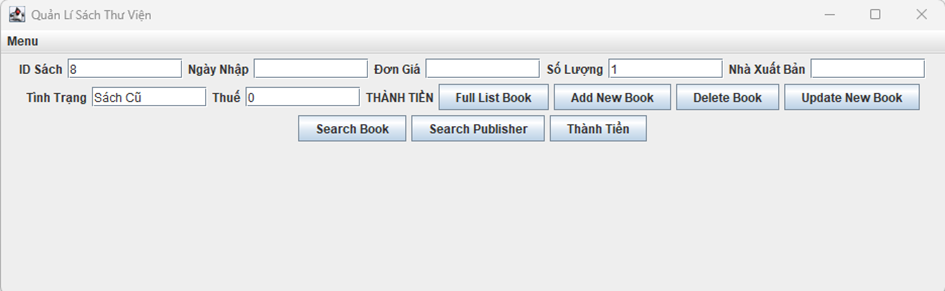
### *3.7.1. Tính thành tiền sách giáo khoa mới*

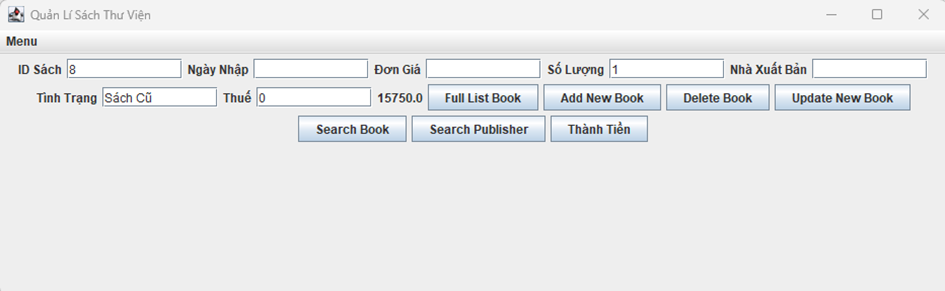
Theo đề bài ta có thành tiền = số lượng \* đơn giá, sẽ nhập vào Form trước các thông tin như mã sách, số lượng, tình trạng sách, thuế và bấm nút Thành Tiền.

Kết quả: kết quả sẽ được hiện trên Label “thành tiền” là 62000.

* Ta có sách có mã “2” có đơn giá là 31000 và số lượng mua là 2 và kết quả là 32000 vậy là kết quả là chính xác

***3.7.2. Tính thành tiền sách giáo khoa cũ***

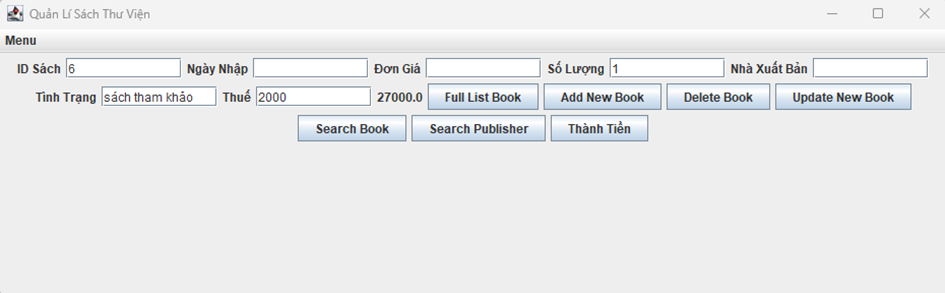
Theo đề bài sách giáo khoa cũ có công thức thành tiền = số lượng \* đơn giá \* 0,5

Trong danh sách ta có sách cũ có mã số 8, số 8 có đơn giá là 31500 vậy khi mua 1 quyển theo công thức đơn giá thi kết quả sẽ là 15750, ta nhập trước các thông tin

Vậy kết quả là 15750 đúng như kết quả mà ta đã tính toán trước

### *3.7.3. Tính thành tiên sách tham khảo*

Theo đề bài, ta có công thức tính thành tiền = số lượng \* đơn giá + thuế, trong danh sách ta có sách có mã “6” là sách tham khảo vậy khi ta mua 1 quyển sách đó với đơn giá là 25000 và khi cộng thêm thuế thì kết quả sẽ là 27000.

Ta cũng sẽ nhập trước các thông tin vào form và có kết quả như sau:

* Kết quả là 27000 đúng. Vậy kết quả là chính xác.

**CHƯƠNG 4: KẾT LUẬN VÀ ĐÁNH GIÁ**

**4.1. Kết quả đạt được**

Phần mềm đã thực hiện được các chức năng như thêm, sửa, xóa, tìm kiếm, tính thành tiền theo từng loại sách.

Sử dụng Jtable để hiển thị thông tin dữ liệu trong database

Sử dụng kết nối với database để truy xuất sửa chữa thông tin dữ liệu.

Tính toán các thực hiện các nghiệp vụ theo yêu cầu đề bài

## 4.2. Cần cải thiện

Cần chỉnh sửa giao diện để phù hợp với người dùng hơn

Cân thêm các chức năng thống kê giá tiền các loại sách, tính trung bình giá các loại sách

Áp dụng đa hình vào bài để thực hiện nghiệp vụ của từng loại sách

Nên tách các chức năng riêng ra không nên để chung hết các nghiệm vụ lại với nhau.