**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: QUẢN LÍ THÔNG TIN SẢN PHẨM**

**PHÂN HỆ: QUẢN LÝ DANH MỤC SẢN PHẨM**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1771020385 | Đỗ Quốc Khánh | 18/12/2004 | CNTT 17-04 |

### 

**Hà Nội, năm 2024**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: QUẢN LÍ THÔNG TIN SẢN PHẨM**

**PHÂN HỆ: QUẢN LÍ DANH MỤC SẢN PHẨM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1771020385 | Đỗ Quốc Khánh | 18/12/2004 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2024**

# **LỜI NÓI ĐẦU**

Ngày nay, công nghệ thông tin đóng vai trò không thể thiếu trong mọi lĩnh vực, đặc biệt là quản lý và xử lý dữ liệu. Trong bối cảnh các doanh nghiệp ngày càng mở rộng quy mô và đa dạng hóa sản phẩm, việc quản lý thông tin sản phẩm một cách hiệu quả trở thành yếu tố sống còn để duy trì tính cạnh tranh và tối ưu hóa hoạt động kinh doanh.

Báo cáo này trình bày việc ứng dụng **Lập trình Hướng Đối Tượng (OOP)** trong xây dựng một hệ thống quản lý thông tin sản phẩm, một công cụ quan trọng giúp doanh nghiệp theo dõi, cập nhật, và phân loại sản phẩm một cách khoa học. OOP, với các đặc điểm như tính kế thừa, tính đóng gói, tính trừu tượng, và tính đa hình, mang lại một cách tiếp cận rõ ràng, linh hoạt và dễ mở rộng trong việc phát triển phần mềm.

Trong nội dung báo cáo, chúng tôi sẽ phân tích chi tiết các yêu cầu của hệ thống, áp dụng các nguyên tắc của OOP để mô hình hóa các thành phần chính như sản phẩm, danh mục sản phẩm, và các chức năng quản lý. Ngoài ra, báo cáo cũng minh họa cách lập trình hướng đối tượng giải quyết các vấn đề thực tế trong quản lý thông tin sản phẩm, từ thêm, sửa, xóa sản phẩm đến quản lý kho và tạo báo cáo doanh thu.

Mục tiêu của báo cáo là giúp người đọc hiểu rõ quy trình thiết kế và phát triển hệ thống quản lý sản phẩm, đồng thời nhận thức được tầm quan trọng của OOP trong việc xây dựng các ứng dụng phần mềm hiệu quả. Báo cáo cũng đưa ra các ứng dụng thực tế của hệ thống, từ cửa hàng bán lẻ nhỏ đến nền tảng thương mại điện tử, và đề xuất hướng phát triển mở rộng trong tương lai.

Hy vọng rằng thông qua báo cáo này, người đọc sẽ có cái nhìn tổng quan và sâu sắc hơn về việc ứng dụng OOP trong giải quyết các bài toán quản lý, từ đó áp dụng vào thực tiễn một cách hiệu quả.

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc183293779)

[**CHƯƠNG 1. MÔ TẢ BÀI TOÁN** 7](#_Toc183293780)

[**1.1 Giới thiệu** 7](#_Toc183293781)

[**1.2** **Chức năng chính** 7](#_Toc183293782)

[**1.3** **Yêu cầu phi chức năng** 10](#_Toc183293783)

[**CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG** 13](#_Toc183293784)

[**2.1. Phân tích yêu cầu:** 13](#_Toc183293785)

[**2.2. Thiết kế hệ thống** 16](#_Toc183293786)

[**2.3. Triển khai.** 20](#_Toc183293787)

[**2.4. Vận hành vào bảo trì.** 21](#_Toc183293788)

[**CHƯƠNG 3. KẾT QUẢ THỰC HIỆN** 23](#_Toc183293789)

[**3.1. Công nghệ đã sử dụng.** 23](#_Toc183293790)

[**3.2. Tiến độ thực hiện.** 23](#_Toc183293791)

[**3.3 Kết quả đạt được** 23](#_Toc183293792)

[**KẾT LUẬN** 43](#_Toc183293808)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 44](#_Toc183293809)

# **CHƯƠNG 1. MÔ TẢ BÀI TOÁN**

## **1.1 Giới thiệu**

Quản lý thông tin sản phẩm là một bài toán phổ biến và thiết yếu trong các doanh nghiệp hiện đại, đặc biệt trong các lĩnh vực sản xuất, bán lẻ và thương mại điện tử. Với sự gia tăng không ngừng của số lượng sản phẩm và sự đa dạng trong danh mục, việc quản lý hiệu quả dữ liệu sản phẩm đã trở thành một yếu tố quan trọng, góp phần nâng cao hiệu suất hoạt động và hỗ trợ ra quyết định kinh doanh.

Lập trình Hướng Đối Tượng (OOP) được lựa chọn làm phương pháp tiếp cận chính trong việc xây dựng hệ thống quản lý thông tin sản phẩm. OOP cung cấp một cách tiếp cận trực quan và hiệu quả thông qua việc mô hình hóa các đối tượng trong thế giới thực như sản phẩm, danh mục, kho hàng và giao dịch. Các đặc tính của OOP như tính trừu tượng, tính đóng gói, tính kế thừa, và tính đa hình giúp tăng tính linh hoạt, dễ bảo trì và mở rộng hệ thống trong tương lai.

Báo cáo này tập trung trình bày quá trình xây dựng và triển khai hệ thống quản lý thông tin sản phẩm bằng cách sử dụng OOP. Hệ thống được thiết kế để thực hiện các chức năng cơ bản như:

* Thêm, sửa, xóa và tìm kiếm sản phẩm.
* Quản lý danh mục sản phẩm.
* Theo dõi tồn kho và bán hàng.
* Tạo báo cáo doanh thu và thống kê tồn kho.

Các chức năng này không chỉ giúp doanh nghiệp tối ưu hóa quy trình quản lý mà còn giảm thiểu sai sót khi xử lý dữ liệu. Hệ thống được thiết kế hướng đến các doanh nghiệp vừa và nhỏ, cửa hàng bán lẻ hoặc các cá nhân kinh doanh trực tuyến, mang lại một giải pháp đơn giản nhưng hiệu quả cho bài toán quản lý thông tin sản phẩm.

Báo cáo sẽ cung cấp một cái nhìn tổng quan về bài toán, phân tích các yêu cầu, mô tả hệ thống và trình bày các ứng dụng thực tế, đồng thời làm nổi bật vai trò của OOP trong việc giải quyết bài toán quản lý thông tin sản phẩm.

## **Chức năng chính**

**1. Quản lý thông tin sản phẩm**

* **Thêm sản phẩm mới:**
  + Nhập thông tin chi tiết của sản phẩm bao gồm:
    - **ID sản phẩm**: Mã nhận diện duy nhất.
    - **Tên sản phẩm**: Tên gọi của sản phẩm.
    - **Loại hàng**: Phân loại sản phẩm (ví dụ: Điện tử, Thực phẩm, Thời trang, v.v.).
    - **Số lượng**: Số lượng sản phẩm có sẵn trong kho.
    - **Giá**: Giá bán của sản phẩm.
    - **Xuất xứ**: Nơi sản xuất hoặc nguồn gốc của sản phẩm.
  + Đảm bảo không có sản phẩm trùng ID trong hệ thống.
* **Cập nhật sản phẩm:**
  + Sửa đổi thông tin của sản phẩm khi cần, chẳng hạn:
    - Thay đổi tên, loại hàng, hoặc giá sản phẩm.
    - Cập nhật số lượng khi có biến động kho hàng.
    - Thay đổi thông tin xuất xứ nếu cần.
* **Xóa sản phẩm:**
  + Loại bỏ các sản phẩm không còn kinh doanh hoặc hết hạn sử dụng.

**2. Hiển thị và tìm kiếm sản phẩm**

* **Hiển thị danh sách sản phẩm:**
  + Liệt kê đầy đủ thông tin của tất cả các sản phẩm, bao gồm ID, tên, loại hàng, số lượng, giá, và xuất xứ.
  + Hỗ trợ sắp xếp theo các tiêu chí như giá, số lượng, hoặc loại hàng.
* **Tìm kiếm sản phẩm:**
  + Tìm kiếm sản phẩm theo nhiều tiêu chí, bao gồm:
    - **ID sản phẩm:** Xác định sản phẩm cụ thể.
    - **Tên sản phẩm:** Tìm kiếm theo từ khóa trong tên.
    - **Loại hàng:** Lọc danh sách sản phẩm theo danh mục cụ thể.
    - **Xuất xứ:** Tìm sản phẩm theo nguồn gốc hoặc nơi sản xuất.

**3. Quản lý số lượng hàng tồn**

* **Kiểm tra tồn kho:**
  + Theo dõi số lượng hàng hóa trong kho để đảm bảo luôn có đủ sản phẩm phục vụ nhu cầu khách hàng.
  + Hiển thị danh sách các sản phẩm sắp hết hàng hoặc hết hàng.
* **Cập nhật số lượng:**
  + Tăng số lượng khi nhập thêm hàng.
  + Giảm số lượng khi sản phẩm được bán hoặc xuất kho.

**4. Quản lý giá bán**

* **Cập nhật giá:**
  + Điều chỉnh giá bán của sản phẩm dựa trên các yếu tố như thị trường, chi phí nhập hàng, hoặc chương trình khuyến mãi.
* **Hiển thị sản phẩm theo mức giá:**
  + Lọc danh sách sản phẩm theo giá (ví dụ: từ thấp đến cao hoặc trong khoảng giá nhất định).

**5. Báo cáo và phân tích**

* **Báo cáo tồn kho:**
  + Tổng hợp số lượng sản phẩm còn lại trong kho theo loại hàng, xuất xứ, hoặc các tiêu chí khác.
* **Báo cáo giá trị hàng hóa:**
  + Tính toán tổng giá trị sản phẩm hiện có trong kho dựa trên số lượng và giá bán.
* **Phân tích loại hàng:**
  + Đưa ra thông tin chi tiết về tỷ lệ từng loại hàng trong hệ thống (ví dụ: % sản phẩm thuộc loại Điện tử hoặc Thực phẩm).

## **Yêu cầu phi chức năng**

**1. Hiệu suất (Performance)**

* **Xử lý dữ liệu nhanh chóng:**
  + Hệ thống phải đảm bảo thời gian xử lý các thao tác thêm, sửa, xóa, và tìm kiếm sản phẩm trong vòng dưới 1 giây, ngay cả khi cơ sở dữ liệu chứa hàng nghìn sản phẩm.
* **Khả năng mở rộng:**
  + Hệ thống phải có khả năng hoạt động hiệu quả khi số lượng sản phẩm, danh mục hoặc người dùng tăng đáng kể mà không ảnh hưởng đến hiệu suất.
* **Xử lý đồng thời:**
  + Hỗ trợ nhiều người dùng thao tác cùng lúc trên hệ thống mà không gây xung đột dữ liệu.

**2. Bảo mật (Security)**

* **Xác thực người dùng:**
  + Hệ thống cần đảm bảo chỉ những người dùng có quyền (như quản trị viên hoặc nhân viên) mới được phép thực hiện các chức năng quản lý sản phẩm, kho, và báo cáo.
* **Bảo vệ dữ liệu:**
  + Thông tin sản phẩm và dữ liệu giao dịch phải được mã hóa khi lưu trữ hoặc truyền tải.
* **Kiểm soát truy cập:**
  + Phân quyền rõ ràng giữa các vai trò người dùng:
    - Quản trị viên: Toàn quyền quản lý.
    - Nhân viên: Chỉ có quyền truy cập các chức năng nhất định.

**3. Khả năng sử dụng (Usability)**

* **Giao diện thân thiện:**
  + Hệ thống cần cung cấp giao diện đơn giản, dễ sử dụng, với hướng dẫn rõ ràng cho người dùng ở mọi cấp độ.
* **Hỗ trợ tìm kiếm nâng cao:**
  + Tích hợp chức năng tìm kiếm thông minh với khả năng gợi ý từ khóa hoặc lọc sản phẩm theo nhiều tiêu chí (giá, loại hàng, xuất xứ).
* **Đa ngôn ngữ:**
  + Hệ thống nên hỗ trợ nhiều ngôn ngữ, đặc biệt là ngôn ngữ chính của người dùng mục tiêu.

**4. Khả năng bảo trì và mở rộng (Maintainability and Scalability)**

* **Dễ bảo trì:**
  + Mã nguồn được tổ chức rõ ràng, tuân theo các nguyên tắc lập trình Hướng Đối Tượng, giúp dễ dàng sửa đổi hoặc nâng cấp.
* **Khả năng mở rộng:**
  + Hệ thống cần cho phép tích hợp thêm các tính năng mới như hỗ trợ thanh toán trực tuyến, quản lý khách hàng, hoặc giao dịch trên nhiều kênh.

**5. Tương thích và tính khả dụng (Compatibility and Availability)**

* **Tương thích đa nền tảng:**
  + Hệ thống phải chạy được trên nhiều môi trường khác nhau, bao gồm cả máy tính (Windows, MacOS) và nền tảng web.
* **Khả dụng cao:**
  + Hệ thống phải luôn sẵn sàng hoạt động, với thời gian ngừng hoạt động không quá 1% trong tháng.

**6. An toàn dữ liệu (Data Integrity)**

* **Đảm bảo dữ liệu không bị trùng lặp hoặc mất mát:**
  + Mỗi sản phẩm phải có một ID duy nhất, và các giao dịch phải được lưu trữ an toàn.
* **Sao lưu và khôi phục dữ liệu:**
  + Hệ thống cần tự động sao lưu định kỳ và hỗ trợ khôi phục dữ liệu trong trường hợp xảy ra lỗi hệ thống.

**7. Tuân thủ quy định (Compliance)**

* **Đáp ứng tiêu chuẩn bảo mật:**
  + Hệ thống cần tuân thủ các tiêu chuẩn bảo mật dữ liệu như GDPR (Quy định bảo vệ dữ liệu chung) nếu hoạt động ở thị trường quốc tế.
* **Hỗ trợ tính năng kiểm tra:**
  + Ghi lại toàn bộ lịch sử thao tác trên hệ thống để dễ dàng kiểm tra hoặc truy vết khi có vấn đề.

# **CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG**

## **2.1. Phân tích yêu cầu:**

**\* Xác định khách hàng:**

* **Khách hàng cuối:**
  + **Người mua sắm:** Khách hàng tìm kiếm và mua sản phẩm trên nền tảng thương mại điện tử. Họ quan tâm đến việc tìm kiếm sản phẩm nhanh chóng, xem chi tiết sản phẩm, và thực hiện thanh toán dễ dàng.
  + **Nhà quản lý và nhân viên quản lý sản phẩm:** Những người này chịu trách nhiệm quản lý thông tin sản phẩm, xử lý các đơn hàng, và theo dõi tình trạng tồn kho. Họ cần công cụ giúp nhập liệu sản phẩm, cập nhật giá cả và số lượng, và tạo các báo cáo doanh thu.
* **Yêu cầu thu thập:**  
  Dựa trên yêu cầu của hệ thống, các chức năng cần có bao gồm:
  + **Quản lý sản phẩm:** Thêm, sửa, xóa và tìm kiếm thông tin sản phẩm.
  + **Quản lý danh mục sản phẩm:** Phân loại sản phẩm vào các nhóm hàng hóa.
  + **Báo cáo doanh thu và thống kê tồn kho:** Cung cấp thông tin giúp nhà quản lý nắm bắt tình trạng kho hàng và hiệu quả kinh doanh.

**\* Phân tích yêu cầu:**

**Yêu cầu chức năng đối với từng thuộc tính:**

**ID sản phẩm (Product ID)**

* **Chức năng:**
  + **Đặc điểm:** Mỗi sản phẩm sẽ có một mã ID duy nhất để phân biệt với các sản phẩm khác trong hệ thống.
  + **Yêu cầu:** ID phải là duy nhất cho mỗi sản phẩm. Hệ thống phải kiểm tra tính duy nhất của ID khi thêm sản phẩm mới vào hệ thống.
  + **Chức năng liên quan:**
    - **Thêm sản phẩm mới:** Khi nhập sản phẩm, người dùng sẽ nhập ID. Hệ thống sẽ kiểm tra xem ID đã tồn tại trong cơ sở dữ liệu hay chưa.
    - **Cập nhật sản phẩm:** Khi chỉnh sửa sản phẩm, ID không được thay đổi, vì ID là đặc trưng duy nhất của mỗi sản phẩm.
    - **Xóa sản phẩm:** Xóa sản phẩm dựa trên ID.
    - **Tìm kiếm sản phẩm:** Cho phép tìm kiếm sản phẩm bằng ID.

**Tên sản phẩm (Product Name)**

* **Chức năng:**
  + **Đặc điểm:** Tên sản phẩm là thông tin quan trọng giúp khách hàng nhận diện và tìm kiếm sản phẩm.
  + **Yêu cầu:** Tên sản phẩm phải rõ ràng, dễ hiểu và không được trùng lặp trong cùng một danh mục sản phẩm.
  + **Chức năng liên quan:**
    - **Thêm sản phẩm mới:** Khi thêm sản phẩm, người dùng sẽ nhập tên sản phẩm. Hệ thống cần kiểm tra tính hợp lệ của tên (không quá dài, không chứa ký tự đặc biệt...).
    - **Cập nhật sản phẩm:** Người dùng có thể thay đổi tên sản phẩm khi cần thiết.
    - **Tìm kiếm sản phẩm:** Hệ thống cần hỗ trợ tìm kiếm sản phẩm dựa trên tên.
    - **Hiển thị danh sách sản phẩm:** Danh sách sản phẩm sẽ được hiển thị theo tên hoặc lọc theo từ khóa trong tên sản phẩm.

**Loại hàng (Product Category)**

* **Chức năng:**
  + **Đặc điểm:** Loại hàng giúp phân loại sản phẩm theo nhóm, ví dụ như **Điện tử**, **Thời trang**, **Thực phẩm**, **Mỹ phẩm**.
  + **Yêu cầu:** Mỗi sản phẩm phải được phân loại vào một hoặc nhiều danh mục sản phẩm để người dùng có thể dễ dàng tìm kiếm và phân loại.
  + **Chức năng liên quan:**
    - **Thêm và chỉnh sửa danh mục:** Hệ thống phải hỗ trợ thêm, sửa, và xóa các loại hàng/danh mục sản phẩm.
    - **Lọc sản phẩm theo loại:** Người dùng có thể tìm kiếm hoặc lọc các sản phẩm theo danh mục.
    - **Cập nhật thông tin sản phẩm:** Cho phép thay đổi loại hàng của sản phẩm khi có yêu cầu.

**Số lượng (Quantity)**

* **Chức năng:**
  + **Đặc điểm:** Số lượng sản phẩm hiện có trong kho.
  + **Yêu cầu:** Hệ thống cần theo dõi số lượng sản phẩm tồn kho và tự động cập nhật khi có giao dịch bán hoặc nhập hàng.
  + **Chức năng liên quan:**
    - **Cập nhật số lượng:** Khi có sản phẩm mới nhập vào kho, số lượng cần được cập nhật. Khi sản phẩm được bán, số lượng phải giảm đi.
    - **Cảnh báo tồn kho thấp:** Hệ thống cần có tính năng cảnh báo khi số lượng sản phẩm trong kho đạt mức tối thiểu để nhà quản lý có thể kịp thời nhập hàng.
    - **Hiển thị số lượng:** Số lượng sản phẩm sẽ được hiển thị chi tiết trong danh sách sản phẩm và khi xem thông tin chi tiết của sản phẩm.

**Giá (Price)**

* **Chức năng:**
  + **Đặc điểm:** Giá là yếu tố quyết định khi khách hàng lựa chọn sản phẩm.
  + **Yêu cầu:** Giá cần được nhập đúng và có thể thay đổi tùy theo các chương trình khuyến mãi, thay đổi thị trường, hoặc chi phí sản xuất.
  + **Chức năng liên quan:**
    - **Cập nhật giá:** Hệ thống cho phép cập nhật giá bán của sản phẩm khi có sự thay đổi.
    - **Áp dụng giảm giá:** Hệ thống có thể hỗ trợ giảm giá cho các sản phẩm theo chương trình khuyến mãi.
    - **Hiển thị giá:** Giá của sản phẩm sẽ được hiển thị khi khách hàng duyệt sản phẩm.
    - **Tính toán tổng giá trị đơn hàng:** Giá sẽ được sử dụng để tính toán tổng giá trị của đơn hàng khi khách hàng thanh toán.

**Xuất xứ (Origin)**

* **Chức năng:**
  + **Đặc điểm:** Xuất xứ của sản phẩm giúp người tiêu dùng biết được nguồn gốc của sản phẩm, có thể là từ một quốc gia hoặc một nhà sản xuất nổi tiếng.
  + **Yêu cầu:** Cung cấp thông tin chi tiết về xuất xứ giúp tăng tính minh bạch và tin cậy cho sản phẩm.
  + **Chức năng liên quan:**
    - **Cập nhật thông tin xuất xứ:** Cho phép cập nhật hoặc thay đổi thông tin về xuất xứ của sản phẩm khi có yêu cầu.
    - **Hiển thị xuất xứ:** Xuất xứ sẽ được hiển thị khi khách hàng xem chi tiết sản phẩm.

## **2.2. Thiết kế hệ thống**

**Thiết kế hệ thống sử dụng mô hình MVC (Model-View-Controller):**

**Model:** là phần chứa các đối tượng dữ liệu và logic nghiệp vụ. Nó không liên quan trực tiếp đến giao diện người dùng và không chứa bất kỳ logic điều khiển nào.  
Các lớp như Product, Category, Inventory, và SalesReport sẽ chịu trách nhiệm lưu trữ và xử lý dữ liệu sản phẩm, danh mục, kho hàng, và doanh thu.

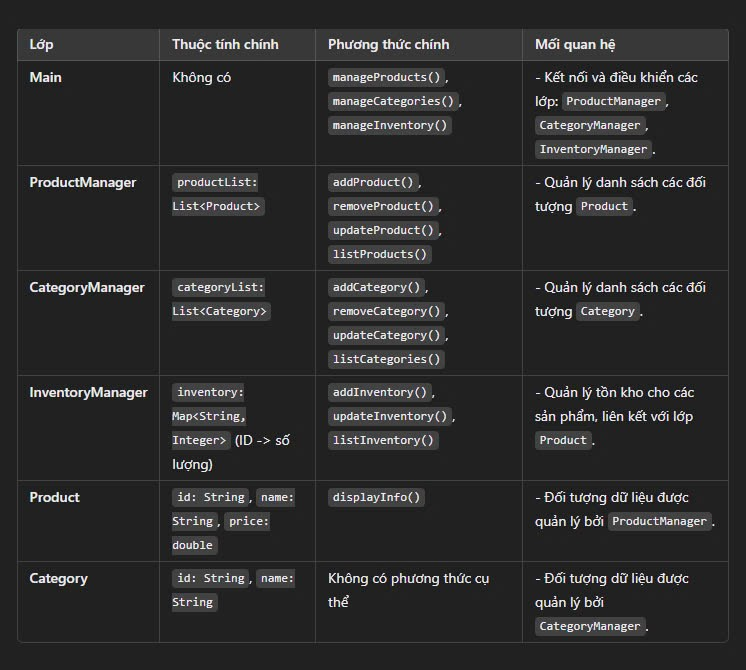
**View:** là phần giao diện người dùng, chịu trách nhiệm hiển thị dữ liệu từ Model và gửi các yêu cầu từ người dùng đến Controller.  
Giao diện người dùng đơn giản với các chức năng chính như thêm, sửa, xóa sản phẩm và báo cáo. Các lớp ProductView, CategoryView, ReportView sẽ đảm nhiệm việc hiển thị dữ liệu.

**Controller:** là phần xử lý logic nghiệp vụ, nhận yêu cầu từ View và thao tác với Model để cập nhật dữ liệu. Nó cũng chịu trách nhiệm quyết định View nào sẽ được hiển thị.

Lớp ProductController hoặc InventoryController sẽ chịu trách nhiệm nhận yêu cầu từ người dùng và điều phối dữ liệu giữa Model và View.

**Mối quan hệ giữa các lớp:**

* **Model** và **View** không trực tiếp liên hệ với nhau. **View** chỉ hiển thị thông tin từ **Model** thông qua **Controller**.
* **Controller** đóng vai trò trung gian giữa **Model** và **View**, nhận yêu cầu từ **View**, thực hiện logic nghiệp vụ và cập nhật **Model**, sau đó yêu cầu **View** hiển thị dữ liệu mới.



**Thiết kế cơ sở dữ liệu (với file .txt):**

* Tạo tệp movies.txt để lưu trữ thông tin về các bộ phim. Các thuộc tính sẽ bao gồm: ID, Tiêu đề, Thể loại, Thời gian, Đạo diễn, và Ngày phát hành.
* Dòng đầu tiên: Tiêu đề cột, mô tả các thuộc tính của bộ phim.
* Các dòng tiếp theo: Dữ liệu về các bộ phim, mỗi bộ phim trên một dòng, các thuộc tính phân tách bằng dấu phẩy.

**Thiết kế giao diện:**

* Hiển thị menu chính với các lựa chọn:

1. Quản lý sản phẩm

2. Quản lý danh mục sản phẩm

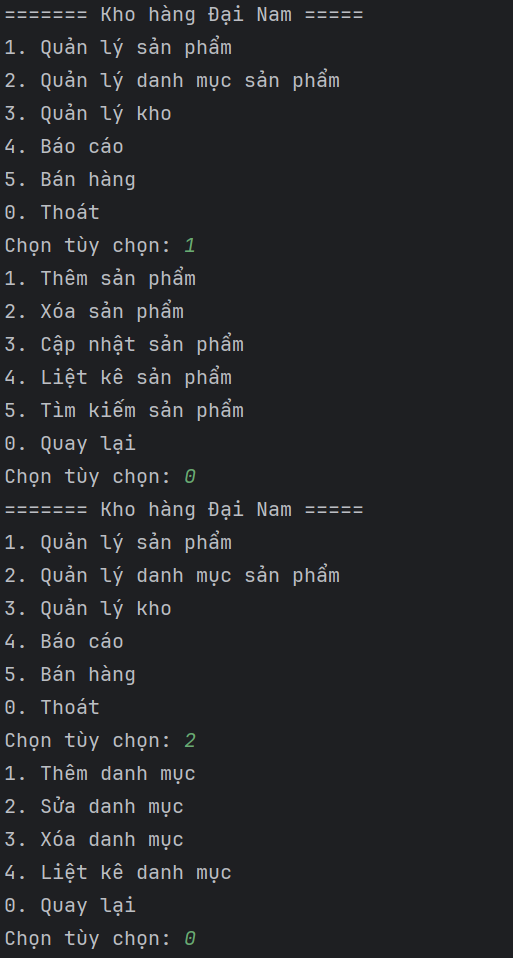
3. Quản lý kho

4. Báo cáo

5. Bán hàng

0. Thoát

* Sử dụng các thông báo rõ ràng để hướng dẫn người dùng nhập liệu và hiển thị kết quả.
* Minh họa



## **2.3. Triển khai.**

* Xây dựng hệ thống:
* Sử dụng ngôn ngữ Java để cài đặt các lớp theo mô hình MVC.
* Hệ thống sẽ lưu trữ và quản lý dữ liệu thông qua các tệp văn bản (.txt) để đọc thông tin, đồng thời xử lý dữ liệu và hiển thị giao diện trên console.
* Chi tiết triển khai:
* Cài đặt các class:
  + Tạo các lớp theo sơ đồ lớp, bao gồm các lớp trong nhóm Model, View, và Controller.
  + Thực hiện các phương thức đọc dữ liệu từ file .txt trong các lớp Model, đảm bảo thông tin được lưu trữ chính xác và có thể khôi phục khi cần.
  + Các lớp View sẽ hiển thị thông tin và nhận đầu vào từ người dùng thông qua giao diện console.
  + Lớp Controller sẽ chịu trách nhiệm điều phối giữa Model và View, xử lý logic nghiệp vụ.
* Xử lý đọc/ghi file:
  + Viết các phương thức đọc dữ liệu từ file .txt để tải dữ liệu ban đầu cho ứng dụng.
  + Đảm bảo tệp được tổ chức hợp lý để dễ dàng quản lý (ví dụ: mỗi tệp lưu một loại dữ liệu như movies.txt).
* Tối ưu hóa xử lý dữ liệu:
  + Xây dựng các lớp tiện ích để xử lý dữ liệu đọc từ file, chẳng hạn như chuyển đổi từ chuỗi (String) sang đối tượng (Object) và ngược lại.
  + Đảm bảo tính toàn vẹn dữ liệu khi xử lý các tác vụ như đặt vé hoặc tính doanh thu.
* Hiển thị giao diện console:
  + Cung cấp menu tương tác để người dùng lựa chọn các chức năng (thêm, xóa, tìm kiếm, đặt vé, xem báo cáo).
  + Hiển thị dữ liệu dưới dạng danh sách dễ đọc, hỗ trợ hiển thị lỗi hoặc thông báo trạng thái khi người dùng thực hiện các thao tác.
* Kiểm thử:
* Viết các test case để kiểm tra các chức năng chính:
  + Đọc dữ liệu từ tệp.
  + Xử lý logic nghiệp vụ (ví dụ: đặt vé, kiểm tra ghế trống).
  + Tính toán doanh thu và số liệu thống kê.
* Đảm bảo ứng dụng hoạt động đúng theo yêu cầu và xử lý được các trường hợp ngoại lệ (ví dụ: file không tồn tại, dữ liệu không hợp lệ).

## **2.4. Vận hành vào bảo trì.**

* **Cài đặt và triển khai:**
* **Hướng dẫn người dùng chạy ứng dụng từ console:**
  + Sau khi đã biên dịch ứng dụng Java, người dùng có thể chạy ứng dụng từ console.
  + Đảm bảo rằng tệp .jar của ứng dụng được tạo thành công và có thể chạy trên hệ thống của người dùng.
* **Hướng dẫn sử dụng tệp .txt:**
  + **Tạo tệp .txt**: Người dùng cần tạo một tệp văn bản để lưu trữ dữ liệu công việc.
  + **Cấu trúc tệp**: Tệp .txt cần có các cột phân tách bằng dấu phẩy (hoặc tab) để lưu trữ thông tin.
  + **Cách chia sẻ quyền truy cập**: Người dùng chỉ cần đảm bảo rằng tệp .txt có thể được đọc từ ứng dụng Java.
* **Cài đặt thêm thư viện (nếu cần):**
  + Nếu ứng dụng sử dụng các thư viện ngoài Java tiêu chuẩn (ví dụ: để xử lý tệp văn bản hoặc các công cụ phụ trợ khác), bạn cần cài đặt các thư viện cần thiết.
  + Các thư viện có thể được cài đặt qua Maven hoặc Gradle, tùy thuộc vào cấu trúc dự án.
* **Bảo trì:**
* **Sửa lỗi phát sinh:**
  + Kiểm tra lỗi ứng dụng khi có sự cố và sửa chữa để đảm bảo ứng dụng hoạt động ổn định.
  + Đảm bảo các tính năng như đọc tệp .txt hoạt động chính xác, không gặp lỗi khi tệp không tồn tại hoặc có lỗi dữ liệu.
* **Cập nhật chức năng mới:**
  + Cập nhật các chức năng ứng dụng nếu có yêu cầu mới, chẳng hạn như thêm mới các loại công việc, tìm kiếm công việc trong tệp, hoặc thêm tính năng báo cáo.
* **Cải thiện hiệu năng:**
  + Đảm bảo rằng ứng dụng có thể xử lý các tệp lớn một cách hiệu quả. Có thể áp dụng các cải tiến như lưu trữ dữ liệu trong bộ nhớ đệm hoặc tối ưu hóa việc đọc/ghi tệp để giảm thiểu độ trễ.

# **CHƯƠNG 3. KẾT QUẢ THỰC HIỆN**

## **3.1. Công nghệ đã sử dụng.**

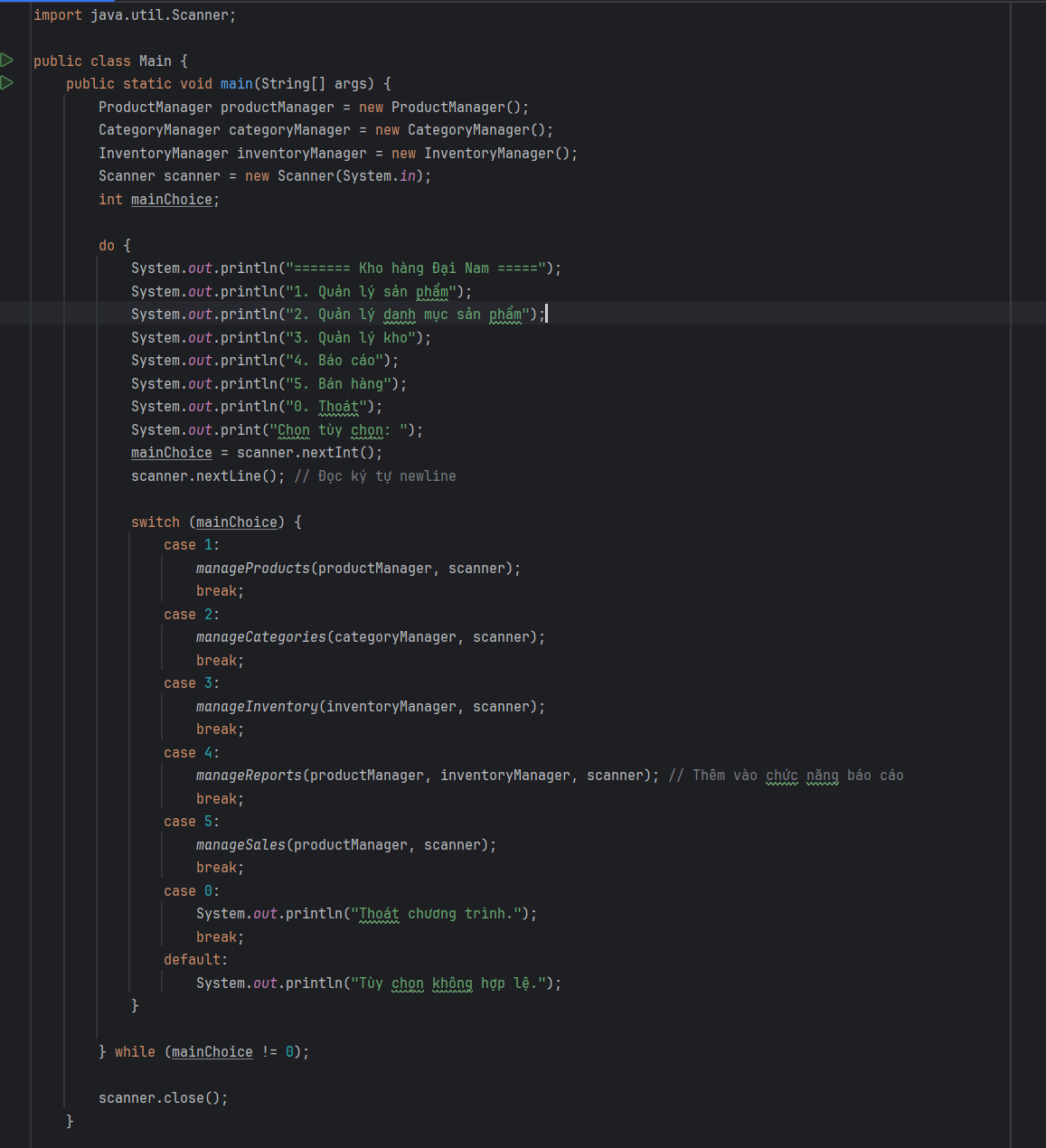
* Ngôn ngữ lập trình: **Java**
* Công cụ: **IntelliJ IDEA**
* Thư viện: Lớp **SimpleDateFormat** trong gói **java.text** được dùng để định dạng và phân tích ngày tháng từ chuỗi theo một mẫu cụ thể, trong khi **ParseException** hỗ trợ xử lý các lỗi xảy ra nếu chuỗi ngày tháng không khớp với định dạng mong muốn. Lớp **Date** trong **java.util** biểu diễn ngày tháng trong Java, còn **ArrayList** và **List** được sử dụng để tạo và quản lý các danh sách động. Cuối cùng, Scanner trong **java.util** được dùng để nhận dữ liệu đầu vào từ người dùng thông qua console.

## **3.2. Tiến độ thực hiện.**

Link github tới dự án: **https://github.com/tyanzuq2811/FIT4007\_PROJECT\_MANAGECINEMA.git**

## **3.3 Kết quả đạt được**

**\* Main.java**

****

**1. Tổng quan chương trình**

* Đây là một chương trình menu-driven (dựa trên menu), nơi người dùng chọn các chức năng từ một menu chính.
* Các chức năng được phân chia rõ ràng thành các nhóm lớn:
  + Quản lý sản phẩm
  + Quản lý danh mục
  + Quản lý kho
  + Báo cáo
  + Bán hàng
* Người dùng có thể thoát chương trình bằng cách chọn 0.

**2. Cấu trúc chính**

Các lớp chính

* ProductManager: Quản lý thông tin và hoạt động liên quan đến sản phẩm.
* CategoryManager: Quản lý danh mục sản phẩm.
* InventoryManager: Quản lý số lượng sản phẩm trong kho.

Vòng lặp chính (do-while)

* Vòng lặp này đảm bảo chương trình tiếp tục hoạt động cho đến khi người dùng chọn thoát (mainChoice = 0).
* Tại mỗi vòng lặp:
  1. Menu chính hiển thị các tùy chọn.
  2. Người dùng nhập lựa chọn.
  3. Chương trình thực hiện hành động tương ứng dựa trên lựa chọn thông qua switch-case.

**3. Các hàm chức năng**

Mỗi tùy chọn trong menu sẽ gọi một hàm quản lý chi tiết. Các hàm này chưa được triển khai trong đoạn mã, nhưng có thể được thiết kế như sau:

(a) manageProducts(ProductManager productManager, Scanner scanner)

* Xử lý các chức năng liên quan đến sản phẩm như:
  + Thêm, xóa, cập nhật thông tin sản phẩm.
  + Liệt kê sản phẩm hiện có.

(b) manageCategories(CategoryManager categoryManager, Scanner scanner)

* Xử lý các chức năng liên quan đến danh mục sản phẩm như:
  + Tạo danh mục mới.
  + Sửa/xóa danh mục.
  + Hiển thị danh sách danh mục.

(c) manageInventory(InventoryManager inventoryManager, Scanner scanner)

* Quản lý số lượng sản phẩm trong kho:
  + Nhập thêm hàng.
  + Xuất bớt hàng.
  + Kiểm tra tồn kho.

(d) manageReports(ProductManager productManager, InventoryManager inventoryManager, Scanner scanner)

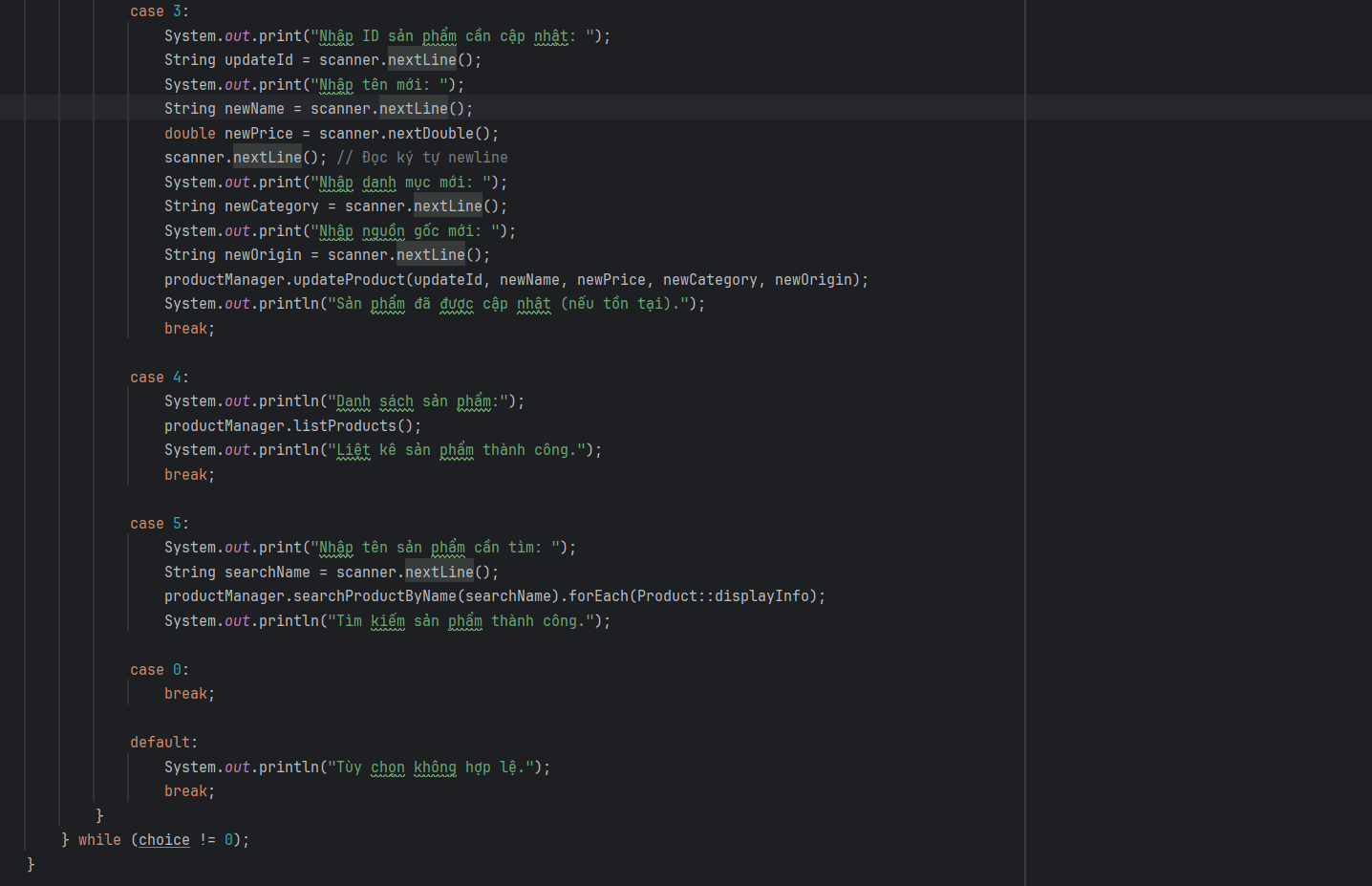
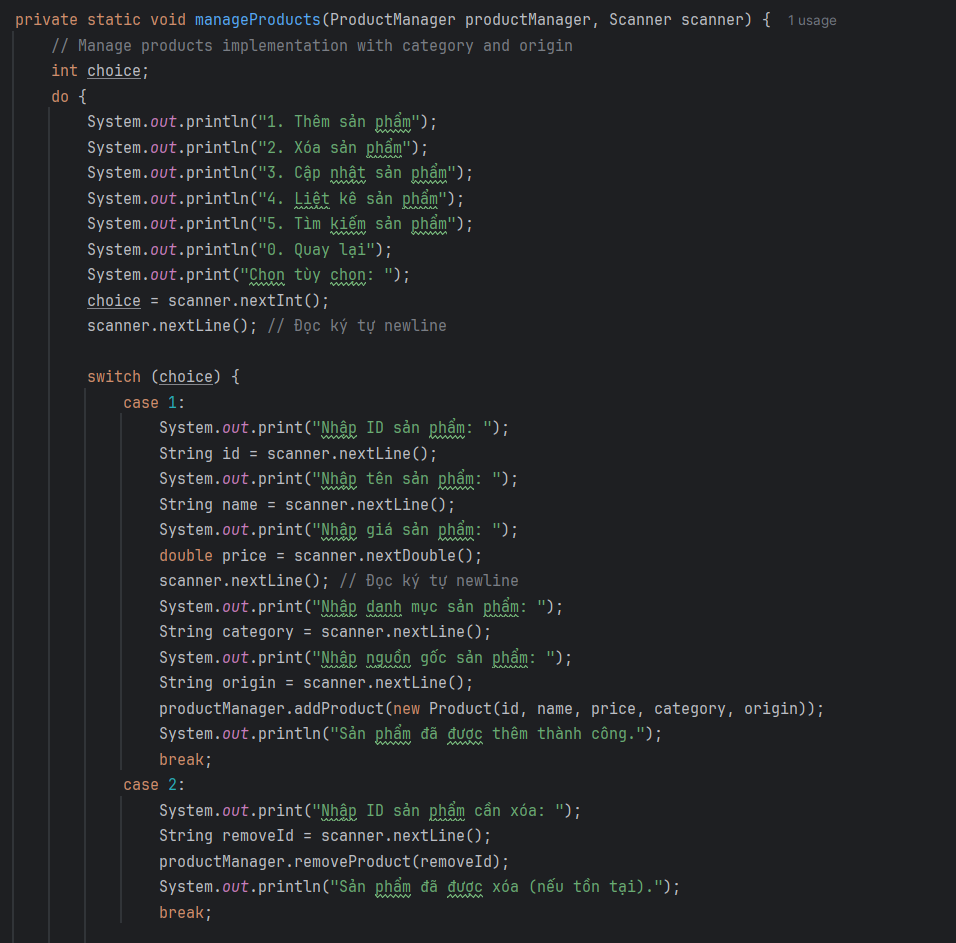
* Tạo báo cáo:
  + Báo cáo doanh số, tồn kho.
  + Thống kê các sản phẩm bán chạy.
  + Phân tích theo danh mục hoặc thời gian.

(e) manageSales(ProductManager productManager, Scanner scanner)

* Quản lý bán hàng:
  + Thực hiện giao dịch bán hàng.
  + Cập nhật tồn kho.
  + Tính toán tổng giá trị đơn hàng.

**4. Điểm nổi bật**

* Modular code: Các chức năng được tách biệt thành các lớp và phương thức riêng biệt, giúp dễ dàng mở rộng và bảo trì.
* Scanner: Được sử dụng để nhập liệu từ người dùng. Việc sử dụng scanner.nextLine() để xử lý ký tự xuống dòng giúp tránh lỗi khi nhập dữ liệu.

****

1. Tổng quan chức năng

* Hàm manageProducts cung cấp menu phụ để người dùng tương tác với các sản phẩm.
* Sử dụng một vòng lặp do-while để đảm bảo menu hiển thị liên tục cho đến khi người dùng chọn quay lại (choice == 0).
* Các chức năng được tổ chức rõ ràng thông qua switch-case, giúp quản lý từng thao tác riêng biệt.

2. Các chức năng chính

(a) Thêm sản phẩm

* Người dùng nhập thông tin sản phẩm bao gồm:
  + ID, tên, giá, danh mục, và nguồn gốc.
* Thông tin này được sử dụng để tạo một đối tượng Product, sau đó được thêm vào qua productManager.addProduct().
* Phản hồi về việc thêm thành công được hiển thị.

(b) Xóa sản phẩm

* Người dùng cung cấp ID của sản phẩm cần xóa.
* productManager.removeProduct() xử lý việc xóa. Nếu sản phẩm tồn tại, nó sẽ bị xóa. Nếu không, không có gì xảy ra.

(c) Cập nhật sản phẩm

* Người dùng nhập ID sản phẩm cần cập nhật, sau đó cung cấp thông tin mới (tên, giá, danh mục, nguồn gốc).
* productManager.updateProduct() thực hiện cập nhật dựa trên ID. Nếu sản phẩm không tồn tại, không có gì xảy ra.

(d) Liệt kê sản phẩm

* Sử dụng productManager.listProducts() để hiển thị tất cả sản phẩm hiện có.
* Đây là một cách nhanh chóng để kiểm tra nội dung kho hàng.

(e) Tìm kiếm sản phẩm

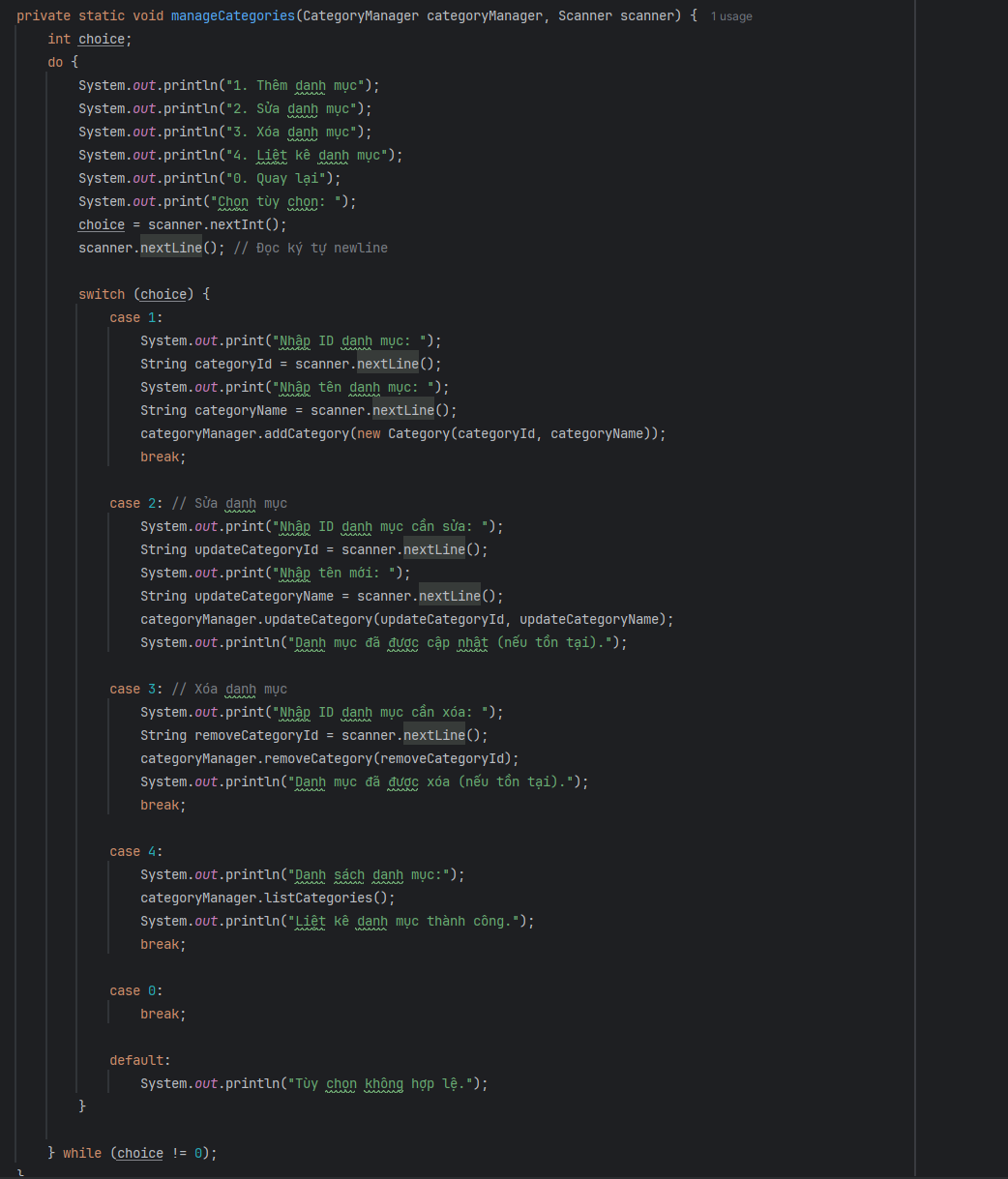
* Người dùng nhập một phần hoặc toàn bộ tên sản phẩm.
* Sử dụng productManager.searchProductByName(), trả về danh sách sản phẩm phù hợp.
* Thông tin chi tiết của từng sản phẩm được hiển thị qua Product::displayInfo.

3. Xử lý nhập liệu

* scanner.nextLine() được sử dụng cẩn thận để xử lý newline thừa sau khi sử dụng nextInt hoặc nextDouble. Điều này tránh lỗi nhảy dòng khi nhập liệu liên tiếp.
* Các lệnh System.out.print hướng dẫn rõ ràng người dùng nhập đúng thông tin.

4. Điểm nổi bật

* Menu đa cấp: Hàm cung cấp một menu con cho việc quản lý sản phẩm, làm cho chương trình dễ dàng sử dụng và duy trì.
* Tách biệt logic: Các chức năng như thêm, xóa, cập nhật được tách riêng biệt trong các phương thức của ProductManager, giúp mã dễ mở rộng.
* Tính năng tìm kiếm: Hỗ trợ tìm kiếm theo tên sản phẩm, trả về danh sách phù hợp.

****

1. Tổng quan chức năng

* Hàm hiển thị một menu phụ, cho phép người dùng thao tác với danh mục sản phẩm.
* Vòng lặp do-while đảm bảo menu hiển thị liên tục cho đến khi người dùng chọn quay lại (choice == 0).
* Các chức năng được xử lý thông qua switch-case, tương tự như trong manageProducts.

2. Các chức năng chính

(a) Thêm danh mục

* Người dùng nhập:
  + ID danh mục: Một chuỗi duy nhất để xác định danh mục.
  + Tên danh mục: Mô tả hoặc tên của danh mục.
* categoryManager.addCategory() thêm danh mục mới vào hệ thống.

(b) Sửa danh mục

* Người dùng nhập ID danh mục cần sửa và tên mới cho danh mục.
* categoryManager.updateCategory() thực hiện cập nhật. Nếu danh mục không tồn tại, không có thay đổi nào được thực hiện.
* Hiển thị thông báo thành công (hoặc lỗi nếu cần cải thiện thêm).

(c) Xóa danh mục

* Người dùng nhập ID danh mục cần xóa.
* categoryManager.removeCategory() thực hiện việc xóa danh mục. Nếu không tồn tại, không có gì xảy ra.
* Hiển thị thông báo phản hồi.

(d) Liệt kê danh mục

* categoryManager.listCategories() hiển thị toàn bộ danh mục hiện có trong hệ thống.
* In thông báo hoàn thành sau khi liệt kê danh mục.

(e) Quay lại

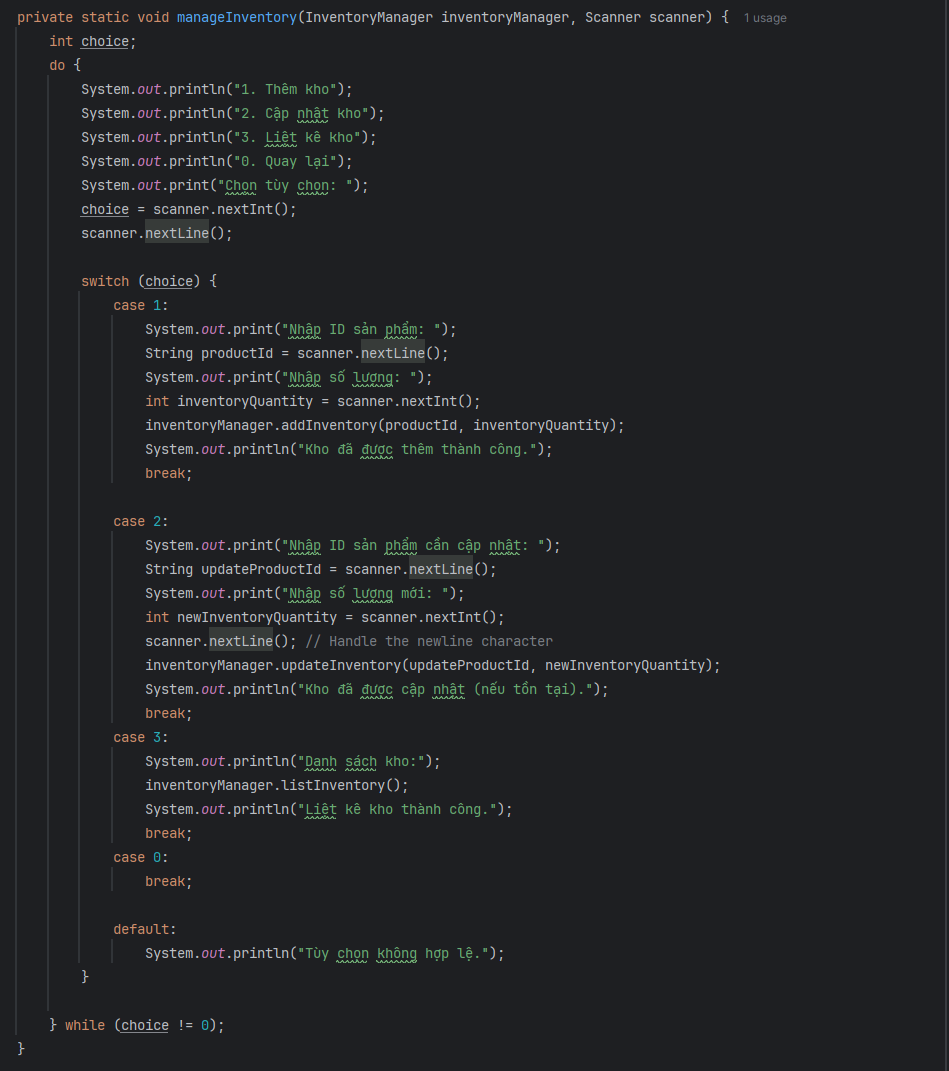
* Khi người dùng nhập 0, vòng lặp thoát và quay lại menu chính.

3. Xử lý nhập liệu

* scanner.nextLine() được sử dụng cẩn thận để đọc ký tự xuống dòng thừa sau khi sử dụng nextInt. Điều này giúp tránh lỗi khi nhập liệu liên tiếp.
* Hướng dẫn người dùng: Các lệnh System.out.print cung cấp chỉ dẫn rõ ràng để người dùng nhập đúng thông tin.

4. Điểm nổi bật

* Tổ chức tốt: Mỗi chức năng được quản lý riêng biệt trong các phương thức của CategoryManager, giúp mã dễ mở rộng và bảo trì.
* Liệt kê danh mục: Chức năng liệt kê cung cấp cái nhìn tổng quan về các danh mục hiện có.
* Thông báo phản hồi: Hiển thị thông báo khi hoàn thành mỗi thao tác giúp người dùng biết trạng thái thao tác.

****

1. Tổng quan chức năng

* Hàm sử dụng menu phụ để cho phép người dùng tương tác với kho hàng.
* Sử dụng vòng lặp do-while để duy trì menu cho đến khi người dùng chọn quay lại (choice == 0).
* switch-case xử lý logic cho từng thao tác cụ thể.

2. Các chức năng chính

(a) Thêm kho

* Người dùng nhập:
  + ID sản phẩm: Xác định sản phẩm muốn thêm vào kho.
  + Số lượng: Số lượng sản phẩm cần thêm.
* inventoryManager.addInventory() xử lý thêm sản phẩm vào kho.
* In thông báo xác nhận thêm thành công.

(b) Cập nhật kho

* Người dùng nhập:
  + ID sản phẩm: Xác định sản phẩm cần cập nhật.
  + Số lượng mới: Giá trị cập nhật số lượng tồn kho.
* inventoryManager.updateInventory() cập nhật số lượng tồn kho cho sản phẩm có ID tương ứng.
* Nếu sản phẩm không tồn tại, thao tác sẽ không có tác dụng. Người dùng cần được thông báo.

(c) Liệt kê kho

* Sử dụng inventoryManager.listInventory() để hiển thị danh sách toàn bộ kho hàng.
* Chức năng này cung cấp cái nhìn tổng quan về sản phẩm và số lượng trong kho.

(d) Quay lại

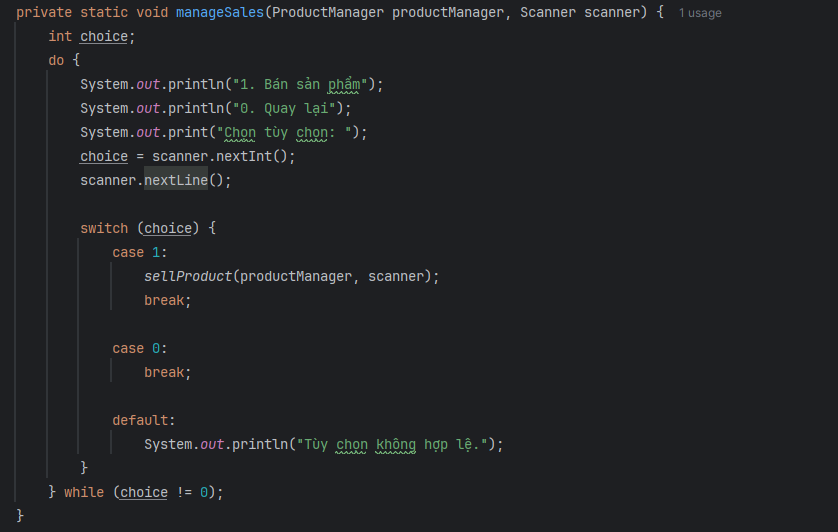
* Khi người dùng nhập 0, vòng lặp kết thúc và quay lại menu chính.

3. Xử lý nhập liệu

* scanner.nextLine() được sử dụng cẩn thận để tránh lỗi nhảy dòng sau khi đọc số nguyên bằng scanner.nextInt().
* Các lệnh System.out.print hướng dẫn rõ ràng giúp người dùng nhập liệu đúng thông tin.

4. Điểm nổi bật

* Tổ chức hợp lý: Các thao tác quản lý kho được xử lý tách biệt thông qua các phương thức trong InventoryManager.
* Chức năng liệt kê: Hỗ trợ quản lý kho hiệu quả bằng cách cung cấp danh sách đầy đủ sản phẩm trong kho.

****

1. Tổng quan chức năng

* Menu phụ có hai lựa chọn:
  + 1: Thực hiện quy trình bán sản phẩm thông qua hàm sellProduct.
  + 0: Quay lại menu chính.
* Sử dụng vòng lặp do-while để hiển thị lại menu sau mỗi thao tác, trừ khi người dùng chọn quay lại.

2. Chi tiết chức năng

(a) Bán sản phẩm

* Chức năng này được gọi thông qua hàm sellProduct.
* Dựa vào productManager, hệ thống có thể thực hiện:
  + Tìm kiếm sản phẩm.
  + Kiểm tra số lượng tồn kho (nếu cần tích hợp với kho).
  + Trừ số lượng sản phẩm khi giao dịch thành công.
  + Xác nhận giao dịch và hiển thị thông tin chi tiết.

(b) Quay lại

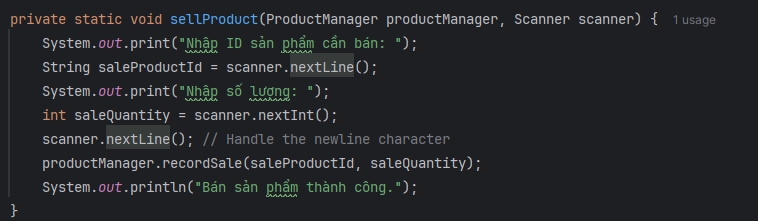
* Khi người dùng chọn 0, vòng lặp kết thúc và hệ thống quay lại menu chính.

(c) Tùy chọn không hợp lệ

* Nếu người dùng nhập tùy chọn không hợp lệ (khác 0 và 1), hệ thống sẽ hiển thị thông báo lỗi.

3. Điểm nổi bật

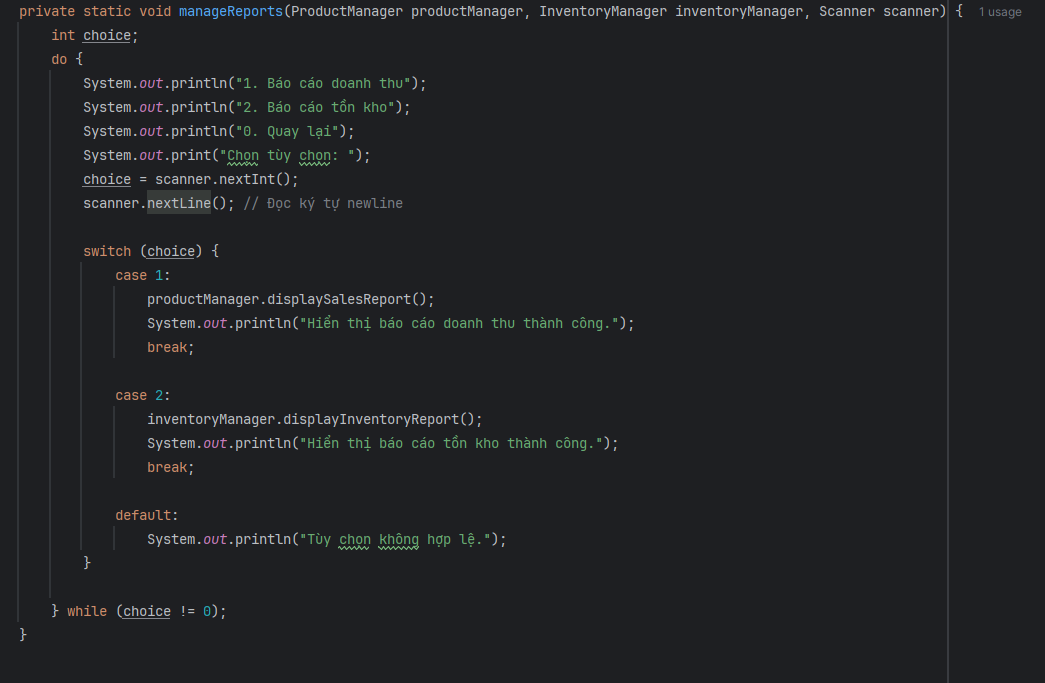
* Thiết kế đơn giản: Menu có ít tùy chọn, giúp người dùng dễ dàng thao tác.
* Phân tách trách nhiệm: Hàm sellProduct được tách riêng để thực hiện logic bán hàng, giúp mã dễ bảo trì và mở rộng.

****

Chức năng:

Hàm này thực hiện các bước sau:

1. Nhập ID sản phẩm:
   * Người dùng nhập ID của sản phẩm cần bán (chuỗi).
2. Nhập số lượng bán:
   * Người dùng nhập số lượng sản phẩm muốn bán (số nguyên).
3. Ghi nhận bán hàng:
   * Gọi hàm recordSale của đối tượng productManager để thực hiện thao tác bán hàng.
4. Phản hồi thành công:
   * Hiển thị thông báo "Bán sản phẩm thành công".

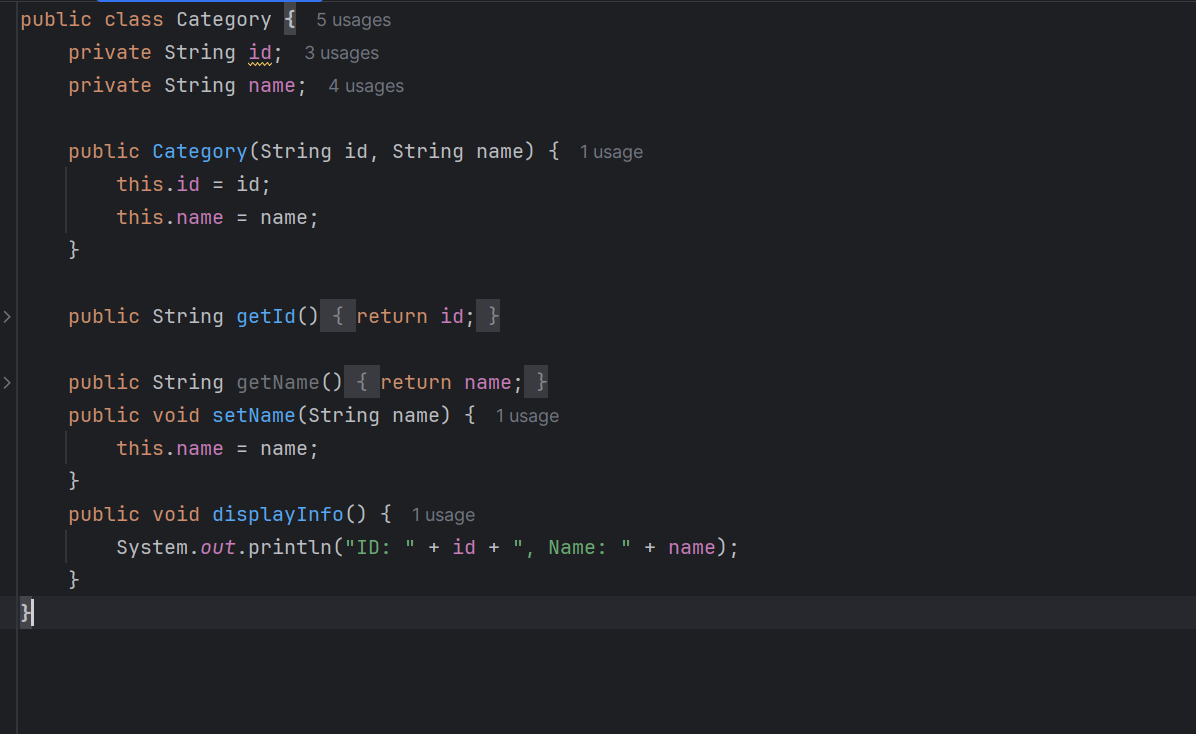
****

Chức năng:

Hàm này hiển thị các báo cáo theo lựa chọn của người dùng:

1. Báo cáo doanh thu:
   * Gọi phương thức displaySalesReport từ đối tượng productManager để hiển thị thông tin doanh thu.
2. Báo cáo tồn kho:
   * Gọi phương thức displayInventoryReport từ đối tượng inventoryManager để hiển thị thông tin hàng tồn kho.
3. Quay lại:
   * Khi chọn "0", hàm thoát khỏi vòng lặp và quay lại menu trước đó.

**\* Category.java**



1. Mô tả chức năng

Lớp Category đại diện cho một danh mục sản phẩm trong hệ thống quản lý kho. Nó chứa thông tin cơ bản về danh mục và cung cấp các phương thức để truy cập, cập nhật và hiển thị thông tin.

2. Cấu trúc và các thành phần

(a) Thuộc tính

1. id:
   * Loại: String.
   * Lưu trữ mã định danh duy nhất của danh mục.
   * Được sử dụng để xác định danh mục trong các hoạt động như thêm, sửa, hoặc xóa.
2. name:
   * Loại: String.
   * Lưu trữ tên của danh mục.

(b) Constructor

public Category(String id, String name)

* Dùng để khởi tạo một đối tượng Category với id và name.
* Giúp đảm bảo rằng mỗi danh mục được tạo ra đều có đầy đủ thông tin ban đầu.

(c) Getter và Setter

1. Getter cho id:
   * Phương thức getId trả về mã danh mục.
   * Chỉ đọc (không có setter) vì id thường không thay đổi sau khi được tạo.
2. Getter và Setter cho name:
   * getName: Trả về tên danh mục.
   * setName: Cập nhật tên danh mục.
   * Việc cung cấp setter cho name là hợp lý vì tên danh mục có thể cần chỉnh sửa.

(d) Hiển thị thông tin

* Phương thức displayInfo:

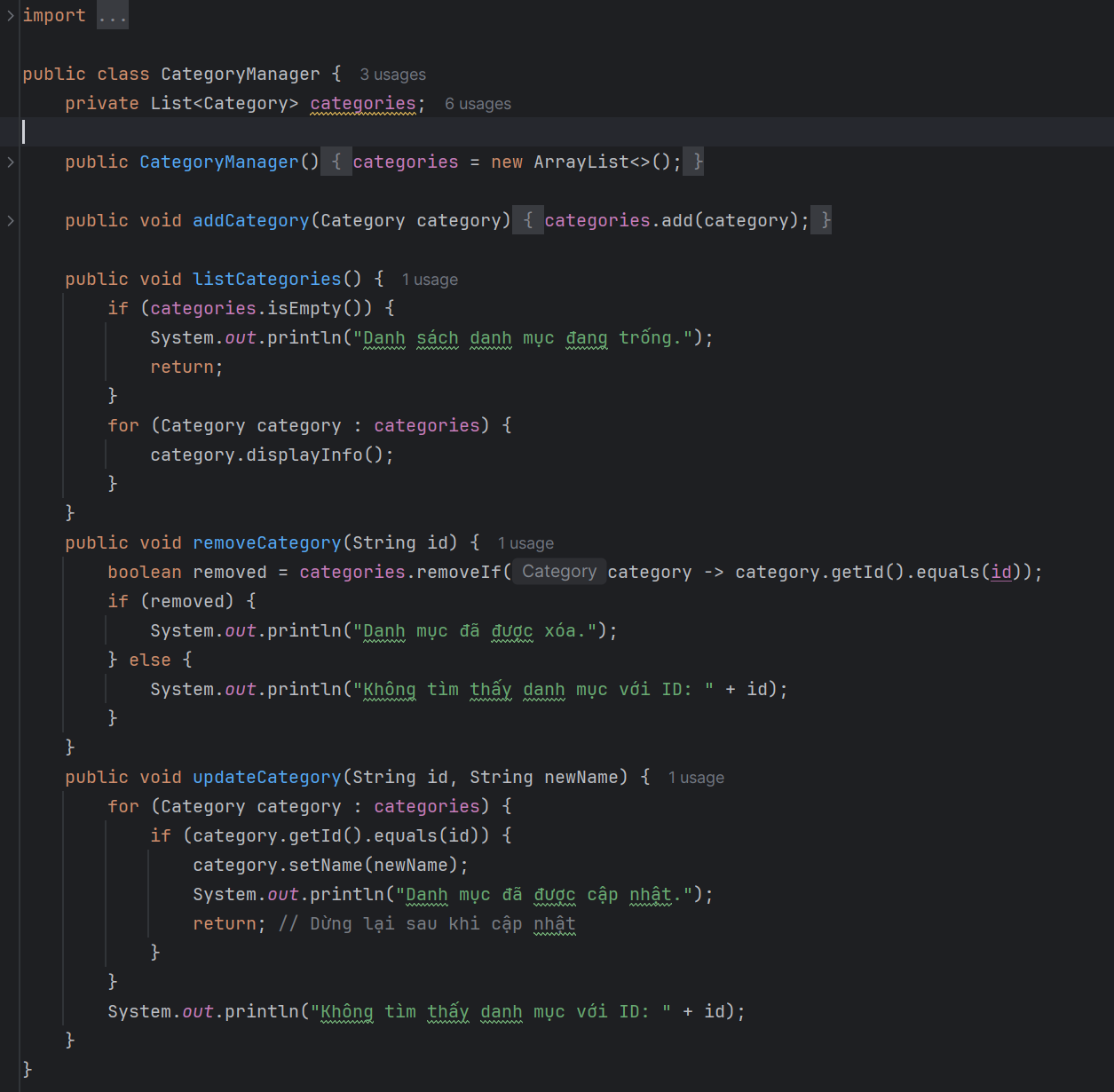
public void displayInfo() {

System.out.println("ID: " + id + ", Name: " + name);

}

* + In thông tin danh mục theo định dạng dễ hiểu.
  + Giúp thuận tiện khi cần liệt kê các danh mục trong chương trình.

\* CategoryManager.java



1. Mô tả chức năng

Lớp CategoryManager chịu trách nhiệm quản lý danh mục trong hệ thống. Nó sử dụng danh sách các đối tượng Category để thực hiện các thao tác như thêm, xóa, cập nhật và liệt kê danh mục.

2. Cấu trúc và các thành phần

(a) Thuộc tính

1. categories:
   * Loại: List<Category>.
   * Sử dụng một ArrayList để lưu trữ danh sách các danh mục.
   * ArrayList cho phép thực hiện các thao tác như thêm, xóa, và tìm kiếm với hiệu năng phù hợp cho danh sách nhỏ đến trung bình.

(b) Constructor

public CategoryManager() {

categories = new ArrayList<>();

}

* Khởi tạo một danh sách trống để lưu các danh mục.
* Đảm bảo rằng đối tượng CategoryManager luôn sẵn sàng để sử dụng ngay sau khi khởi tạo.

(c) Phương thức

1. Thêm danh mục

public void addCategory(Category category)

* + Thêm một đối tượng Category vào danh sách categories.
  + Không kiểm tra trùng lặp ID trước khi thêm.

1. Liệt kê danh mục

public void listCategories()

* + In thông tin tất cả các danh mục trong danh sách.
  + Nếu danh sách trống, hiển thị thông báo: *"Danh sách danh mục đang trống."*.

1. Xóa danh mục

public void removeCategory(String id)

* + Tìm và xóa danh mục dựa trên ID.
  + Sử dụng phương thức removeIf của List để xóa các phần tử thỏa mãn điều kiện:

categories.removeIf(category -> category.getId().equals(id));

* + Thông báo kết quả:
    - Thành công: *"Danh mục đã được xóa."*
    - Thất bại: *"Không tìm thấy danh mục với ID: id."*

1. Cập nhật danh mục

public void updateCategory(String id, String newName)

* + Duyệt qua danh sách categories để tìm danh mục có ID khớp với tham số id.
  + Nếu tìm thấy:
    - Cập nhật tên danh mục (newName) thông qua setter.
    - Hiển thị thông báo: *"Danh mục đã được cập nhật."*
  + Nếu không tìm thấy:
    - Hiển thị thông báo: *"Không tìm thấy danh mục với ID: id."*

# **KẾT LUẬN**

Trong bài tập lớn này, chúng ta đã áp dụng các nguyên lý của lập trình hướng đối tượng (OOP) để xây dựng hệ thống quản lý thông tin sản phẩm cho một nền tảng thương mại điện tử. Hệ thống đã được thiết kế để hỗ trợ các chức năng cơ bản như thêm, sửa, xóa, tìm kiếm và hiển thị thông tin sản phẩm, với sự phân chia rõ ràng giữa các lớp đối tượng như Product, Category, và Inventory.

Thông qua việc sử dụng các kỹ thuật lập trình hướng đối tượng, hệ thống có thể dễ dàng mở rộng và bảo trì trong tương lai. Các đối tượng được tổ chức hợp lý, giúp giảm thiểu sự trùng lặp mã nguồn và tăng tính linh hoạt trong việc phát triển các tính năng mới. Các nguyên lý như kế thừa, đóng gói và đa hình đã được ứng dụng để tối ưu hóa mã nguồn và tăng cường khả năng tái sử dụng.

Ngoài ra, hệ thống còn cho phép người quản trị dễ dàng theo dõi các thông tin liên quan đến sản phẩm như tên, giá cả, số lượng tồn kho và mô tả chi tiết. Việc tổ chức dữ liệu sản phẩm theo danh mục cũng giúp việc tìm kiếm và quản lý trở nên hiệu quả hơn.

Tuy nhiên, trong quá trình thực hiện bài tập này, một số thách thức đã được gặp phải, bao gồm việc đảm bảo tính đồng nhất của dữ liệu trong quá trình thao tác và quản lý hệ thống. Tuy vậy, hệ thống đã đáp ứng được yêu cầu cơ bản và có thể mở rộng để phục vụ cho các yêu cầu phức tạp hơn trong môi trường thương mại điện tử.

Tóm lại, qua bài tập lớn này, chúng ta đã hiểu rõ hơn về cách sử dụng lập trình hướng đối tượng để xây dựng các hệ thống quản lý hiệu quả và bền vững, đặc biệt trong lĩnh vực thương mại điện tử, nơi tính linh hoạt và khả năng mở rộng là yếu tố quan trọng.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

1. Kiều Tuấn Dũng (2024), *Giáo trình Lập trình hướng đối tượng*, Trường Đại học Đại Nam.