**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH PYTHON**

**ĐỀ TÀI: CHƯƠNG TRÌNH QUẢN LÝ SẢN PHẨM ĐIỆN TỬ**

**Giáo viên hướng dẫn: Đặng Khánh Trung**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 1871020667 | Nguyễn Viết Hoàng Vũ | CNTT 18-09 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH PYTHON**

**ĐỀ TÀI CHƯƠNG TRÌNH QUẢN LÝ SẢN PHẨM ĐIỆN TỬ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1871020667 | Nguyễn Viết Hoàng Vũ | 21/07/2006 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Trong bối cảnh nền kinh tế ngày càng phát triển mạnh mẽ, nhu cầu mua sắm và tiêu dùng các sản phẩm công nghệ như điện thoại, máy tính, phụ kiện điện tử ngày càng tăng cao. Điều này đặt ra yêu cầu cho các cửa hàng, doanh nghiệp kinh doanh trong lĩnh vực này phải có một hệ thống quản lý sản phẩm hiệu quả, chính xác và khoa học nhằm đáp ứng nhu cầu của khách hàng, đồng thời nâng cao hiệu quả hoạt động kinh doanh.

Việc quản lý sản phẩm không chỉ đơn thuần là theo dõi số lượng hàng hóa tồn kho, mà còn bao gồm việc cập nhật giá bán, phân loại mặt hàng, và đảm bảo thông tin sản phẩm được lưu trữ một cách đầy đủ, dễ truy xuất. Một hệ thống quản lý tốt sẽ giúp doanh nghiệp giảm thiểu rủi ro thất thoát hàng hóa, kiểm soát được nguồn cung, và nâng cao năng suất làm việc.

Xuất phát từ thực tế đó, em đã chọn đề tài “Quản lý các sản phẩm (như điện thoại, máy tính, phụ kiện)” với mục tiêu xây dựng một hệ thống đơn giản, dễ sử dụng để hỗ trợ công tác quản lý sản phẩm. Đề tài sẽ tập trung vào việc lưu trữ, xử lý các thông tin cơ bản của sản phẩm như: mã sản phẩm, tên sản phẩm, số lượng tồn kho và giá bán hiện tại. Thông qua đề tài này, em mong muốn áp dụng được những kiến thức đã học vào thực tiễn, đồng thời rèn luyện kỹ năng phân tích và xây dựng hệ thống quản lý.

Em xin chân thành cảm ơn thầy/cô hướng dẫn và các anh/chị đã hỗ trợ, góp ý trong quá trình thực hiện đề tài.

**Mục lục**

|  |  |
| --- | --- |
| Lời nói đầu | **3** |
| Mục tiêu | **5** |
| Giới thiệu đề tài | **6** |
| Phân tích chương trình | **7** |

|  |  |  |
| --- | --- | --- |
|  | *Khai báo danh sách sản phẩm* | **7** |
|  | *Chức năng hiển thị menu* | **7** |
|  | *Chức năng thêm sản phẩm* | **8** |
|  | *Chức năng hiển thị danh sách* | **9** |
|  | *Chức năng tìm kiếm sản phẩm theo mã* | **10** |
|  | *Chức năng cập nhật sản phẩm* | **11** |
|  | *Chức năng xóa sản phẩm* | **12** |
|  | *Sắp xếp theo giá* | **13** |
|  | *Tìm sản phẩm giá cao nhất* | **15** |
|  | *Tìm sản phẩm giá thấp nhất* | **16** |
|  | *Tìm sản phàm theo tên* | **17** |
|  | *Lọc sản phẩm* | **18** |
|  | *Lưu danh sách vào file* | **20** |
|  | *Đọc danh sách từ file* | **20** |
|  | *Hiển thị sản phẩm lưu trong file* | **22** |
|  | *Thống kê kho hàng* | **24** |
|  | *Vòng lặp chính* | **26** |

|  |  |
| --- | --- |
| Tổng kết | **29** |

|  |  |  |
| --- | --- | --- |
|  | *Tổng kết những điều đã làm được* | **29** |
|  | *Hướng phát triển của đề tài* | **29** |
|  | *Kết luận* | **30** |

**I. Mục tiêu**

Đề tài “Quản lý các sản phẩm (như điện thoại, máy tính, phụ kiện)” được thực hiện với các mục tiêu cụ thể sau:

1. Xây dựng hệ thống quản lý sản phẩm đơn giản, dễ sử dụng, hỗ trợ người quản lý theo dõi và cập nhật thông tin các sản phẩm như điện thoại, máy tính, phụ kiện một cách hiệu quả.
2. Lưu trữ và xử lý các thông tin quan trọng của sản phẩm, bao gồm: mã sản phẩm, tên sản phẩm, số lượng tồn kho và giá bán hiện tại.
3. Hỗ trợ việc kiểm soát hàng tồn kho, giúp người quản lý dễ dàng biết được số lượng sản phẩm còn lại, từ đó đưa ra quyết định nhập hàng hoặc điều chỉnh kinh doanh kịp thời.
4. Giảm thiểu sai sót trong quá trình quản lý thủ công, nâng cao độ chính xác và hiệu quả trong công việc quản lý sản phẩm.
5. Ứng dụng kiến thức lập trình và cơ sở dữ liệu đã học để thiết kế và xây dựng một phần mềm quản lý có tính thực tiễn, làm nền tảng cho các hệ thống quản lý quy mô lớn hơn trong tương lai.

**II. Giới thiệu về đề tài**

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, các sản phẩm điện tử như điện thoại thông minh, máy tính, và phụ kiện công nghệ đã trở thành những mặt hàng thiết yếu trong đời sống hàng ngày. Kéo theo đó là sự mở rộng nhanh chóng của các cửa hàng kinh doanh mặt hàng công nghệ. Việc quản lý số lượng lớn sản phẩm với thông tin đa dạng đòi hỏi một hệ thống quản lý hiệu quả và chính xác.

Tuy nhiên, trên thực tế, nhiều cơ sở kinh doanh vẫn đang sử dụng phương pháp quản lý thủ công như ghi chép bằng sổ sách hoặc bảng tính Excel, dẫn đến nhiều khó khăn trong việc kiểm soát số lượng hàng hóa, giá bán, hoặc khi cần truy xuất thông tin. Những hạn chế này có thể gây ra tình trạng thiếu hụt, dư thừa hàng hóa, thất thoát tài sản, hoặc chậm trễ trong việc ra quyết định kinh doanh.

Xuất phát từ thực tiễn đó, đề tài “Quản lý các sản phẩm (như điện thoại, máy tính, phụ kiện)” được lựa chọn nhằm xây dựng một hệ thống quản lý đơn giản, có khả năng lưu trữ và cập nhật thông tin sản phẩm một cách nhanh chóng, chính xác. Hệ thống sẽ hỗ trợ người quản lý theo dõi các thông tin cơ bản như: mã sản phẩm, tên sản phẩm, số lượng tồn và giá bán hiện tại, từ đó phục vụ hiệu quả cho hoạt động kinh doanh.

Đề tài không chỉ mang ý nghĩa thực tiễn mà còn giúp người thực hiện rèn luyện kỹ năng lập trình, tư duy hệ thống và khả năng áp dụng kiến thức công nghệ thông tin vào giải quyết các bài toán thực tế.

**III. Phân tích chương trình**

***1*. *Khai báo danh sách sản phẩm***

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ, đồng hồ

Nội dung do AI tạo ra có thể không chính xác.

* **Chức năng:** Dòng này tạo một **danh sách rỗng** có tên là san\_pham\_list. Đây là **biến lưu trữ trung tâm** chứa toàn bộ thông tin về các sản phẩm trong chương trình quản lý kho.
* **Vai trò:** Là nơi lưu trữ tạm thời các sản phẩm được nhập vào trong quá trình chạy chương trình. Mỗi phần tử trong san\_pham\_list là một dictionary chứa thông tin về một sản phẩm
* **Cách hoạt động:** 
  + **Khởi tạo ban đầu:** Danh sách rỗng khi chương trình bắt đầu.
  + **Khi thêm sản phẩm:** Mỗi lần gọi them\_san\_pham(), một dictionary sản phẩm mới sẽ được append() vào danh sách.
  + **Khi đọc từ file:** Nếu gọi doc\_tu\_file(), danh sách này sẽ được nạp dữ liệu từ file CSV.
  + **Khi kết thúc chương trình:** Dữ liệu trong danh sách sẽ mất (nếu chưa lưu vào file), vì nó chỉ tồn tại trong bộ nhớ tạm.

***2. Chức năng hiển thị menu***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* **Chức năng:** Hàm hien\_thi\_menu() có nhiệm vụ hiển thị danh sách các chức năng của chương trình quản lý sản phẩm ra màn hình. Đây là một giao diện văn bản để người dùng biết mình có thể làm gì trong chương trình.
* **Vai trò:**
  + Đóng vai trò như **bảng điều khiển chính (main menu)** cho người dùng.
  + Cung cấp **hướng dẫn lựa chọn** để người dùng dễ dàng thực hiện các thao tác như: thêm, tìm, sửa, xóa, lưu, thống kê, v.v.
  + Là điểm khởi đầu của mỗi vòng lặp trong chương trình để đợi người dùng nhập lệnh.
* **Cách hoạt động:** 
  + Khi được gọi, hàm này in ra màn hình danh sách các tùy chọn từ 1 đến 14, và 0 để thoát chương trình.
  + Sau khi menu hiển thị, chương trình đợi người dùng nhập số tương ứng với chức năng muốn sử dụng.
  + Sự lựa chọn này sẽ được xử lý trong vòng lặp chính while True bằng chuỗi if-elif để gọi hàm tương ứng.
* **Kết quả:**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.**

***3. Chức năng thêm sản phẩm***

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

* **Chức năng:** Hàm them\_san\_pham() dùng để **thêm một sản phẩm mới vào danh sách quản lý (san\_pham\_list)**. Nó yêu cầu người dùng nhập đầy đủ thông tin của sản phẩm như mã, tên, số lượng, và giá bán.
* **Vai trò:**
  + Cho phép người dùng bổ sung sản phẩm mới vào hệ thống quản lý.
  + Kiểm tra dữ liệu đầu vào nhằm đảm bảo tính hợp lệ trước khi lưu trữ.
  + Ngăn chặn mã sản phẩm trùng lặp, tránh lỗi trùng khóa khi xử lý dữ liệu.
* **Cách thức hoạt động:**
* **Nhập mã sản phẩm** từ bàn phím:
  + Nếu để trống, thông báo lỗi và dừng thao tác.
  + Nếu mã đã tồn tại trong san\_pham\_list, thông báo lỗi trùng mã.
* **Nhập tên sản phẩm**:
  + Nếu để trống, thông báo lỗi và dừng.
* **Nhập số lượng tồn**:
  + Kiểm tra dữ liệu có phải là số nguyên.
  + Nếu không hợp lệ hoặc âm, thông báo lỗi và dừng.
* **Nhập giá bán hiện tại**:
  + Kiểm tra dữ liệu có phải là số thực.
  + Nếu âm hoặc sai định dạng, thông báo lỗi và dừng.
* **Thêm vào danh sách** nếu tất cả thông tin hợp lệ:

san\_pham\_list.append({"ma": ma, "ten": ten, "so\_luong": so\_luong, "gia": gia})

* **In ra thông báo thành công**.
* **Kết quả:**

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Nội dung do AI tạo ra có thể không chính xác.

***4. Chức năng hiển thị danh sách***

***Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Hàm hien\_thi\_danh\_sach() có nhiệm vụ **hiển thị toàn bộ sản phẩm** đang có trong danh sách san\_pham\_list dưới dạng bảng, gồm các thông tin: mã sản phẩm, tên sản phẩm, số lượng tồn và giá bán.
* **Vai trò:**
  + Cung cấp **cái nhìn tổng quan** về toàn bộ danh sách sản phẩm hiện có.
  + Hỗ trợ người quản lý **kiểm tra nhanh** thông tin từng sản phẩm.
  + Là cơ sở để thực hiện các thao tác tiếp theo như cập nhật, tìm kiếm hoặc thống kê.
* **Cách hoạt động:** 
  + **Kiểm tra danh sách sản phẩm**:
    - Nếu san\_pham\_list rỗng → in thông báo: "Danh sách sản phẩm trống." và kết thúc hàm.
  + **Hiển thị tiêu đề cột** theo định dạng bảng:
    - print("\n{:<10} {:<25} {:<15} {:<15}".format("Mã SP", "Tên sản phẩm", "Số lượng", "Giá bán (VND)"))
  + **Duyệt qua từng sản phẩm trong danh sách** và in ra thông tin chi tiết:
    - Sử dụng định dạng căn lề đẹp mắt để bảng dễ nhìn.
    - Giá bán được định dạng có dấu , phân tách hàng nghìn.
* **Kết quả:**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

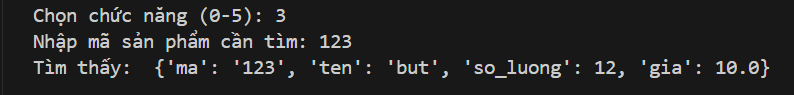
Nội dung do AI tạo ra có thể không chính xác.

***5. Chức năng tìm kiếm sản phẩm theo mã***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* **Chức năng:** Hàm tim\_san\_pham() có chức năng **tìm kiếm sản phẩm theo mã** do người dùng nhập. Nếu tìm thấy, chương trình sẽ in ra thông tin chi tiết của sản phẩm đó.
* **Vai trò :**
  + Hỗ trợ người quản lý **tra cứu nhanh sản phẩm** theo mã định danh.
  + Giúp người dùng tiết kiệm thời gian thay vì phải tìm thủ công trong toàn bộ danh sách.
  + Tăng tính tương tác, tiện lợi khi thao tác với hệ thống quản lý sản phẩm.
* **Kết quả:**



***6. Chức năng cập nhật sản phẩm***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* **Chức năng:** Hàm cap\_nhat\_san\_pham() cho phép cập nhật lại thông tin số lượng và giá bán của một sản phẩm đã có trong danh sách, dựa trên mã sản phẩm mà người dùng nhập vào.
* **Vai trò:**
  + Cho phép người quản lý **điều chỉnh thông tin sản phẩm kịp thời** khi có thay đổi về số lượng tồn kho hoặc giá bán.
  + Đảm bảo dữ liệu trong hệ thống luôn được **cập nhật chính xác**, phản ánh đúng thực tế của kho hàng.
  + Là một phần quan trọng trong chu trình bảo trì và vận hành hệ thống quản lý sản phẩm.
* **Cách Hoạt động:**

Người dùng được yêu cầu nhập **mã sản phẩm cần cập nhật**.

Chương trình sử dụng vòng lặp for để duyệt qua danh sách san\_pham\_list:

* + Nếu tìm thấy sản phẩm có mã trùng khớp:
  + Yêu cầu người dùng nhập **số lượng mới** và **giá bán mới**.
  + Cập nhật lại các giá trị trong từ điển sản phẩm tương ứng.
  + In thông báo "Đã cập nhật sản phẩm" và kết thúc hàm bằng return.

Nếu không tìm thấy sản phẩm sau khi duyệt hết danh sách → in thông báo "Không tìm thấy sản phẩm.".

* **Kết quả:**

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.**

***7. Chức năng xóa sản phẩm***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* **Chức năng:** Hàm xoa\_san\_pham() cho phép người dùng xóa một sản phẩm khỏi danh sách quản lý, dựa trên mã sản phẩm được nhập vào. Nếu tìm thấy, người dùng sẽ được hỏi xác nhận trước khi tiến hành xóa.
* **Vai trò:**
  + Giúp người quản lý loại bỏ sản phẩm không còn tồn tại, đã ngừng kinh doanh hoặc bị nhập nhầm ra khỏi danh sách.
  + Đảm bảo tính chính xác và gọn gàng của dữ liệu kho hàng.
  + Hạn chế rối loạn và lỗi phát sinh từ các sản phẩm không còn phù hợp.
* **Cách hoạt động:**

Chương trình yêu cầu người dùng nhập mã sản phẩm cần xóa.

Duyệt qua danh sách san\_pham\_list:

* + Nếu tìm thấy sản phẩm có mã trùng khớp:
  + In ra thông tin sản phẩm được tìm thấy.
  + Yêu cầu người dùng xác nhận (y/n) trước khi xóa.
  + Nếu người dùng chọn "y":
  + Gọi san\_pham\_list.remove(sp) để xóa sản phẩm.
  + In thông báo "Đã xóa sản phẩm."
  + Nếu "n":
  + In "Hủy xóa sản phẩm."
  + Kết thúc hàm bằng return.

Nếu không tìm thấy sản phẩm → in ra: "Không tìm thấy sản phẩm."

* **Kết quả:**

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Nội dung do AI tạo ra có thể không chính xác.

***8. Sắp xếp theo giá***

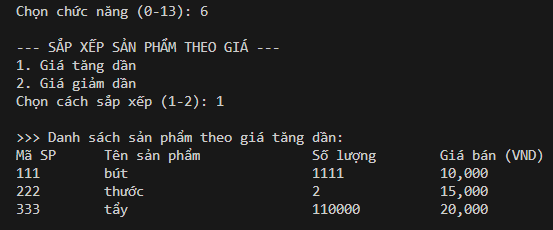
***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Sắp xếp danh sách sản phẩm theo giá bán
* **Vai trò:** Hàm sap\_xep\_theo\_gia() có vai trò giúp người dùng dễ dàng quan sát, so sánh và phân tích các sản phẩm trong kho theo tiêu chí **giá bán**, từ đó phục vụ các mục đích như:
  + Xác định mặt hàng giá thấp để khuyến mãi.
  + Xác định mặt hàng giá cao để kiểm tra tồn kho hoặc đầu tư.
  + Dễ dàng đưa ra chiến lược kinh doanh theo nhóm giá.

Chức năng này rất quan trọng trong các hệ thống quản lý kho hàng, bán lẻ, hoặc thương mại điện tử.

* **Cách hoạt động:**
  + **Kiểm tra danh sách trống**  
    Nếu san\_pham\_list rỗng, chương trình thông báo và kết thúc hàm, tránh lỗi khi sắp xếp danh sách trống.
  + **Hiển thị tùy chọn sắp xếp**  
    Người dùng được yêu cầu chọn một trong hai tùy chọn:
    - **1**: Sắp xếp theo giá tăng dần (rẻ đến đắt)
    - **2**: Sắp xếp theo giá giảm dần (đắt đến rẻ)
  + **Xử lý lựa chọn**
    - Nếu lựa chọn hợp lệ (1 hoặc 2), chương trình sẽ dùng hàm sorted() để tạo một bản sao danh sách đã được sắp xếp theo trường "gia" (giá bán).
    - Biến reverse trong hàm sorted() được dùng để quyết định hướng sắp xếp:
      * reverse=False: tăng dần
      * reverse=True: giảm dần
  + **Hiển thị kết quả**
    - Sau khi sắp xếp, chương trình sẽ hiển thị bảng sản phẩm đã được định dạng đẹp mắt, bao gồm mã, tên, số lượng, và giá bán.
* **Kết quả:**



***9. Tìm sản phẩm giá cao nhất***

***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Tìm sản phẩm có giá cao nhất.
* **Vai trò:** Hàm san\_pham\_gia\_cao\_nhat() có vai trò hỗ trợ người dùng xác định **sản phẩm đang được bán với giá cao nhất** trong danh sách sản phẩm hiện có. Việc xác định sản phẩm giá cao nhất mang lại các lợi ích sau:
  + Giúp quản lý dễ dàng nhận diện sản phẩm cao cấp trong kho.
  + Hỗ trợ phân tích thị trường và phân khúc giá.
  + Là cơ sở để triển khai chương trình marketing hoặc khuyến mãi cho nhóm hàng cao cấp.
* **Cách hoạt động:**

**Bước 1: Kiểm tra danh sách sản phẩm**

* Nếu danh sách san\_pham\_list đang rỗng (không có sản phẩm nào), chương trình in ra thông báo "Danh sách sản phẩm trống." và kết thúc hàm bằng return.

**Bước 2: Tìm sản phẩm có giá cao nhất**

* Hàm sử dụng hàm tích hợp sẵn max() kết hợp với biểu thức key=lambda x: x["gia"] để so sánh giá ("gia") của các sản phẩm trong danh sách.
* Kết quả là một đối tượng sản phẩm (sp\_max) có giá trị "gia" lớn nhất trong toàn bộ danh sách.

**Bước 3: Hiển thị thông tin sản phẩm**

* Sau khi tìm được sản phẩm có giá cao nhất, chương trình hiển thị thông tin của sản phẩm đó: mã sản phẩm, tên sản phẩm và giá bán (được định dạng có dấu phẩy cho dễ đọc).

**Kết quả:**

***Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Nội dung do AI tạo ra có thể không chính xác.***

***10. Tìm sản phẩm giá thấp nhất***

***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Tìm sản phẩm có giá thấp nhất.
* **Vai trò:** Hàm san\_pham\_gia\_thap\_nhat() đảm nhiệm vai trò tìm kiếm và hiển thị sản phẩm có giá bán thấp nhất trong danh sách sản phẩm hiện có. Đây là một chức năng quan trọng trong quản lý kho vì:
  + Giúp nhận diện sản phẩm phổ thông, giá rẻ.
  + Hỗ trợ phân tích cấu trúc giá để điều chỉnh hoặc đề xuất chương trình khuyến mãi.
  + Cung cấp thông tin hữu ích trong việc tư vấn bán hàng hoặc phân khúc khách hàng
* **Cách hoạt động:**
  + **Bước 1: Kiểm tra dữ liệu đầu vào**
    - Trước tiên, hàm kiểm tra xem danh sách sản phẩm san\_pham\_list có dữ liệu hay không.
    - Nếu danh sách trống, chương trình sẽ in ra thông báo "Danh sách sản phẩm trống." và kết thúc hàm bằng return.
  + **Bước 2: Tìm sản phẩm có giá thấp nhất**
    - Hàm sử dụng hàm tích hợp min() cùng với tham số key=lambda x: x["gia"] để so sánh thuộc tính "gia" của các sản phẩm.
    - Kết quả là một sản phẩm có giá trị nhỏ nhất (thấp nhất) trong danh sách — lưu vào biến sp\_min.
  + **Bước 3: Hiển thị kết quả**
    - Thông tin sản phẩm có giá thấp nhất sẽ được in ra màn hình bao gồm mã sản phẩm, tên sản phẩm và giá bán (được định dạng có dấu phẩy phân cách hàng nghìn).
* **Kết quả:**

***Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Nội dung do AI tạo ra có thể không chính xác.***

***11. Tìm sản phẩm theo tên***

***Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Tìm kiếm sản phẩm theo tên.
* **Vai trò:** Hàm tim\_kiem\_theo\_ten() có vai trò cho phép người dùng tìm kiếm một hoặc nhiều sản phẩm trong danh sách dựa trên từ khóa nhập vào là **tên sản phẩm**. Đây là chức năng quan trọng trong chương trình quản lý sản phẩm, nhằm:
  + Giúp người quản lý tra cứu nhanh sản phẩm theo tên.
  + Hỗ trợ tìm kiếm một phần hoặc toàn bộ tên sản phẩm (tìm kiếm gần đúng).
  + Nâng cao trải nghiệm người dùng, đặc biệt khi danh sách sản phẩm lớn.
* **Cách hoạt động:**

**Bước 1: Nhập từ khóa cần tìm**

Người dùng được yêu cầu nhập tên sản phẩm cần tìm kiếm. Hàm sử dụng phương thức .strip().lower() để loại bỏ khoảng trắng và chuyển tất cả ký tự về chữ thường nhằm đảm bảo việc so sánh không phân biệt chữ hoa hay thường.

**Bước 2: Kiểm tra tính hợp lệ**

Nếu người dùng không nhập bất kỳ ký tự nào (input rỗng), chương trình sẽ hiển thị thông báo lỗi: "Tên sản phẩm không được để trống." và dừng việc tìm kiếm.

**Bước 3: Tìm kiếm sản phẩm phù hợp**

Chương trình sử dụng danh sách hiểu (list comprehension) để lọc ra các sản phẩm có tên chứa chuỗi ký tự mà người dùng nhập vào. Việc tìm kiếm là tìm kiếm gần đúng (không cần khớp hoàn toàn tên).

**Bước 4: Hiển thị kết quả**

Nếu tìm thấy sản phẩm phù hợp: hiển thị số lượng kết quả và in thông tin chi tiết của từng sản phẩm, bao gồm mã sản phẩm, tên, số lượng và giá bán.Nếu không tìm thấy kết quả nào: thông báo "Không tìm thấy sản phẩm nào phù hợp."

* **Kết quả:**

***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

***12. Lọc sản phẩm***

***Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Lọc sản phẩm theo tiêu chí cụ thể
* **Vai trò:** Hàm loc\_san\_pham() có vai trò cho phép người dùng **lọc ra những sản phẩm trong danh sách** dựa trên một trong hai tiêu chí:
  + Giá sản phẩm lớn hơn một mức nhất định do người dùng nhập vào.
  + Số lượng sản phẩm thấp hơn một mức nhất định do người dùng chỉ định.

Tính năng lọc này giúp người dùng **tập trung vào nhóm sản phẩm có đặc điểm cụ thể**, phục vụ các mục tiêu như:

* + Xác định nhóm sản phẩm đắt tiền.
  + Phát hiện những mặt hàng có số lượng tồn kho thấp để kịp thời bổ sung.
  + Hỗ trợ ra quyết định nhanh chóng trong quản lý hàng hóa.
* **Cách Hoạt động:**

**Bước 1: Hiển thị lựa chọn tiêu chí lọc**

* + Người dùng được yêu cầu chọn tiêu chí lọc:
  + Nhập **1** để lọc các sản phẩm có **giá lớn hơn** mức giá tối thiểu.
  + Nhập **2** để lọc các sản phẩm có **số lượng thấp hơn** mức tối đa.

**Bước 2: Nhập ngưỡng lọc**

* + Tùy vào lựa chọn ở bước trước, người dùng được yêu cầu nhập:
  + Một mức giá tối thiểu (kiểu float).
  + Hoặc một số lượng tối đa (kiểu int).

**Bước 3: Thực hiện lọc dữ liệu**

* + Sử dụng **list comprehension**, chương trình lọc các sản phẩm phù hợp điều kiện:
  + Với lựa chọn 1: lấy các sản phẩm có gia > muc\_gia.
  + Với lựa chọn 2: lấy các sản phẩm có so\_luong < muc\_so\_luong.

**Bước 4: Hiển thị kết quả**

* + Nếu có sản phẩm phù hợp: hiển thị danh sách các sản phẩm kèm thông tin chi tiết (mã, tên, số lượng, giá).
  + Nếu không có sản phẩm nào phù hợp: thông báo “Không có sản phẩm nào phù hợp.”
* **Kết quả:**

***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

***13. Lưu vào file***

***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Lưu danh sách sản phẩm vào tệp CSV.
* **Vai trò:** Hàm luu\_vao\_file() có chức năng **ghi toàn bộ danh sách sản phẩm hiện có vào một tệp CSV** có tên là san\_pham.csv. Việc lưu dữ liệu ra file có vai trò rất quan trọng trong các chương trình quản lý vì:
  + Đảm bảo **dữ liệu được lưu trữ lâu dài**, không bị mất khi chương trình kết thúc.
  + Cho phép người dùng **xem, chỉnh sửa, hoặc chia sẻ dữ liệu** bằng các phần mềm như Microsoft Excel hoặc Google Sheets.
  + Hỗ trợ tính năng **sao lưu** và **khôi phục** dữ liệu khi cần thiết.
* **Cách hoạt động:**
* **Bước 1: Mở file CSV để ghi**
  + Sử dụng with open() để mở hoặc tạo tệp san\_pham.csv ở chế độ ghi ("w").
  + Đối số newline="" giúp tránh tạo dòng trắng thừa trong file CSV.
  + Bộ mã hóa utf-8 đảm bảo hỗ trợ tiếng Việt không bị lỗi ký tự.
* **Bước 2: Ghi dữ liệu bằng csv.DictWriter**
  + Dùng csv.DictWriter để ghi dữ liệu từ danh sách san\_pham\_list vốn chứa các phần tử là dictionary.
  + Xác định các trường (fieldnames) là: "ma", "ten", "so\_luong", "gia".
  + Ghi dòng tiêu đề (header) bằng writeheader().
  + Ghi từng dòng dữ liệu sản phẩm bằng writerows(san\_pham\_list).
* **Bước 3: Xử lý lỗi ngoại lệ**
  + Nếu trong quá trình ghi file xảy ra lỗi (ví dụ: quyền ghi file bị từ chối, ổ đĩa đầy,...), chương trình sẽ bắt lỗi và thông báo chi tiết bằng except Exception as e.
* **Kết quả:**

******

***14. Đọc từ file***

***Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Đọc dữ liệu sản phẩm từ tệp CSV vào chương trình
* **Vai trò:** Hàm doc\_tu\_file() có chức năng **đọc dữ liệu từ tệp CSV** (cụ thể là tệp san\_pham.csv) và **nạp vào danh sách san\_pham\_list** đang được sử dụng trong chương trình. Đây là một chức năng cực kỳ quan trọng nhằm:
  + **Phục hồi lại dữ liệu sản phẩm** đã được lưu trữ trước đó.
  + Cho phép chương trình **tiếp tục sử dụng dữ liệu cũ** mà không cần nhập lại thủ công.
  + **Tăng tính linh hoạt** và chuyên nghiệp của hệ thống khi có thể ghi nhớ trạng thái làm việc qua các phiên khác nhau.
* **Cách hoạt động:**

**Bước 1: Mở file để đọc**

* + Dùng with open("san\_pham.csv", mode="r", encoding="utf-8") để mở file ở chế độ đọc với mã hóa tiếng Việt.
  + Nếu tệp không tồn tại, sẽ sinh ra ngoại lệ FileNotFoundError.

**Bước 2: Đọc nội dung bằng csv.DictReader**

* + Dùng csv.DictReader(file) để đọc từng dòng dữ liệu từ file CSV và chuyển thành từng dictionary (từ điển).

**Bước 3: Làm rỗng danh sách sản phẩm**

* + Trước khi nạp dữ liệu mới, chương trình xóa hết nội dung cũ bằng san\_pham\_list.clear() để tránh trùng lặp.

**Bước 4: Chuyển kiểu dữ liệu và thêm vào danh sách**

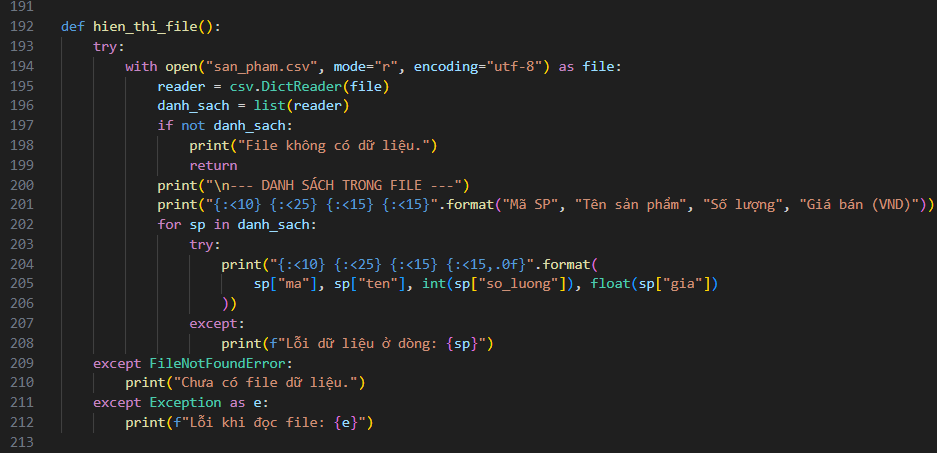
* + Với mỗi dòng đọc được, hàm sẽ:
  + Ép kiểu so\_luong về int
  + Ép kiểu gia về float
  + Thêm dòng dữ liệu hợp lệ vào san\_pham\_list
  + Nếu dòng nào có lỗi về kiểu dữ liệu (ví dụ: chữ trong cột số lượng), dòng đó sẽ bị bỏ qua và hiển thị cảnh báo.

**Bước 5: Xử lý ngoại lệ**

* + Nếu file không tồn tại: thông báo chưa có dữ liệu (FileNotFoundError)
  + Nếu có lỗi khác (như lỗi đọc tệp), sẽ thông báo lỗi chi tiết cho người dùng (except Exception as e)
* **Kết quả:**

******

***15. Hiển thị danh sách lưu trong file***

******

* **Chức năng:** Hiển thị danh sách sản phẩm từ file CSV ra màn hình
* **Vai trò:** Hàm hien\_thi\_file() đảm nhận nhiệm vụ **đọc và hiển thị toàn bộ nội dung sản phẩm từ tệp san\_pham.csv lên màn hình** theo định dạng rõ ràng. Vai trò chính bao gồm:
  + **Giúp người dùng xem nhanh nội dung dữ liệu đang lưu trong file.**
  + Đảm bảo tính minh bạch, hỗ trợ kiểm tra và xác nhận dữ liệu.
  + **Phục vụ cho mục đích thống kê, báo cáo hoặc kiểm tra nhanh** khi không cần thao tác trực tiếp với danh sách trong bộ nhớ.
* **Cách Hoạt động:**

**Bước 1: Mở tệp CSV để đọc**

* + Sử dụng câu lệnh with open("san\_pham.csv", mode="r", encoding="utf-8") để mở tệp ở chế độ đọc.
  + Tệp được đọc với chuẩn mã hóa UTF-8 để đảm bảo hiển thị đúng tiếng Việt.

**Bước 2: Đọc dữ liệu từ file**

* + Sử dụng csv.DictReader để đọc các dòng trong file và chuyển thành danh sách các từ điển (dictionary).
  + Chuyển kết quả thành danh sách danh\_sach = list(reader) để tiện xử lý.

**Bước 3: Kiểm tra dữ liệu**

* + Nếu danh sách rỗng, in ra thông báo File không có dữ liệu. và kết thúc hàm.

**Bước 4: Hiển thị dữ liệu**

* + In tiêu đề bảng bao gồm: Mã SP, Tên sản phẩm, Số lượng, Giá bán.
  + Duyệt qua từng sản phẩm trong danh sách và in ra thông tin định dạng theo hàng cột.

**Bước 5: Xử lý lỗi từng dòng**

* + Nếu có lỗi trong việc chuyển kiểu dữ liệu (ví dụ so\_luong không phải số), dòng đó sẽ bị bỏ qua và thông báo lỗi chi tiết được in ra (Lỗi dữ liệu ở dòng: {sp}).

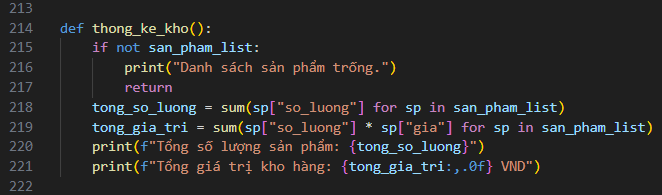
**Bước 6: Xử lý ngoại lệ**

* + Nếu tệp không tồn tại, in thông báo Chưa có file dữ liệu.
  + Nếu có lỗi khác, in nội dung chi tiết của lỗi (Lỗi khi đọc file: ...)
* **Kết quả:**

***Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.***

***16. Thống kê kho hàng***

******

* **Chức năng:** Thống kê tổng số lượng và giá trị kho hàng.
* **Vai trò:** Hàm thong\_ke\_kho() có vai trò thực hiện **thống kê nhanh toàn bộ dữ liệu sản phẩm trong kho**, bao gồm:
  + **Tổng số lượng sản phẩm**: Giúp người dùng biết có bao nhiêu đơn vị hàng hóa đang tồn kho.
  + **Tổng giá trị kho**: Tính toán toàn bộ giá trị hàng hóa dựa trên số lượng và giá bán của từng sản phẩm.
  + Vai trò này đặc biệt quan trọng đối với:
  + Việc **quản lý hàng tồn**.
  + **Lập kế hoạch nhập hàng hoặc bán hàng.**
  + **Báo cáo tài chính**, kiểm kê tài sản định kỳ.
* **Cách hoạt động:**

**Bước 1: Kiểm tra dữ liệu**

* + Nếu danh sách sản phẩm (san\_pham\_list) trống, hàm in thông báo "Danh sách sản phẩm trống." và dừng lại.
  + Điều này đảm bảo không thực hiện tính toán trên dữ liệu rỗng, tránh lỗi logic.

**Bước 2: Tính tổng số lượng sản phẩm**

* + Sử dụng hàm sum() kết hợp với vòng lặp duyệt qua danh sách san\_pham\_list, lấy giá trị so\_luong của từng sản phẩm.
  + tong\_so\_luong = sum(sp["so\_luong"] for sp in san\_pham\_list)

**Bước 3: Tính tổng giá trị kho**

* + Duyệt qua từng sản phẩm, nhân số lượng với giá bán để lấy tổng giá trị.
  + Câu lệnh:
  + tong\_gia\_tri = sum(sp["so\_luong"] \* sp["gia"] for sp in san\_pham\_list)

**Bước 4: Hiển thị kết quả**

* + In ra tổng số lượng sản phẩm và tổng giá trị kho hàng theo định dạng tiền tệ Việt Nam (sử dụng :,.0f để có dấu phẩy phân cách hàng nghìn).
* **Kết quả:**

***Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, thuật in máy

Nội dung do AI tạo ra có thể không chính xác.***

***17. Vòng lặp chính***

***Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.***

* **Chức năng:** Điều khiển chương trình bằng menu.
* **Vai trò:** Đoạn mã trên đóng vai trò là **vòng lặp điều khiển chính** của toàn bộ chương trình quản lý sản phẩm. Cụ thể, nó:
  + Hiển thị menu chức năng cho người dùng.
  + Nhận lựa chọn từ bàn phím.
  + Gọi đúng hàm xử lý tương ứng với lựa chọn đó.
  + Giữ cho chương trình luôn hoạt động cho đến khi người dùng chọn thoát (0).
  + Đây là **trung tâm điều phối** mọi hoạt động trong chương trình. Nếu không có phần này, các chức năng riêng lẻ sẽ không thể vận hành đồng bộ hoặc theo quy trình tương tác với người dùng.
* **Cách hoạt động:**

**Bước 1: Hiển thị menu**

* Hàm hien\_thi\_menu() được gọi để in ra danh sách các chức năng có thể sử dụng (ví dụ: thêm sản phẩm, sắp xếp, tìm kiếm...).

**Bước 2: Nhận lựa chọn từ người dùng**

* Người dùng được yêu cầu nhập một số từ bàn phím đại diện cho chức năng muốn sử dụng (từ 0 đến 14).
* Câu lệnh:

lua\_chon = input("Chọn chức năng (0-13): ")

**Bước 3: Kiểm tra và thực hiện chức năng**

* Đoạn mã sử dụng chuỗi các lệnh if...elif... để **so sánh lựa chọn của người dùng với các chức năng cụ thể**.

Ví dụ:

* Nếu nhập "1" thì gọi hàm them\_san\_pham().
* Nếu nhập "7" thì gọi hàm san\_pham\_gia\_cao\_nhat().

**Bước 4: Xử lý thoát chương trình**

* Nếu người dùng nhập "0", chương trình in thông báo **"Thoát chương trình."** và kết thúc vòng lặp bằng break.

**Bước 5: Xử lý sai lệch đầu vào**

* Nếu người dùng nhập một giá trị không hợp lệ (không nằm trong danh sách "0" đến "14"), chương trình sẽ in:

print("Lựa chọn không hợp lệ. Vui lòng chọn lại.")

* Giúp cải thiện trải nghiệm người dùng và đảm bảo chương trình không bị lỗi.

**IV. Tổng kết**

**1. Tổng kết những điều đã làm được**

Qua quá trình thực hiện đề tài “**Xây dựng chương trình quản lý sản phẩm**”, em đã hoàn thành được các nội dung chính sau:

* Thiết kế chương trình có giao diện dòng lệnh dễ sử dụng, giúp người dùng thao tác quản lý sản phẩm một cách thuận tiện.
* Tổ chức mã nguồn theo hướng cấu trúc, chia rõ các chức năng thành từng hàm, đảm bảo tính rõ ràng và dễ bảo trì.
* Cài đặt đầy đủ các chức năng cơ bản trong quản lý sản phẩm:
  + Thêm, sửa, xóa sản phẩm.
  + Tìm kiếm sản phẩm theo mã và tên.
  + Sắp xếp sản phẩm theo giá tăng dần hoặc giảm dần.
  + Tìm sản phẩm có giá cao nhất và thấp nhất.
  + Lọc sản phẩm theo giá hoặc số lượng.
  + Thống kê tổng số lượng và tổng giá trị sản phẩm tồn kho.
* Xử lý đầu vào cẩn thận, bao gồm kiểm tra dữ liệu rỗng, trùng mã, kiểu dữ liệu hợp lệ và các trường hợp ngoại lệ.
* Cài đặt chức năng đọc và ghi dữ liệu vào file .csv, đảm bảo người dùng có thể lưu trữ dữ liệu lâu dài và sử dụng lại trong các lần chạy tiếp theo.

Thông qua đề tài, em đã củng cố kiến thức về xử lý danh sách, hàm, kiểm tra điều kiện, đọc/ghi file CSV, và xử lý lỗi trong Python.

**2. Hướng phát triển của đề tài**

Trong tương lai, đề tài có thể được phát triển theo các hướng sau:

* **Sử dụng lập trình hướng đối tượng (OOP):**  
  Xây dựng các lớp như SanPham, QuanLySanPham giúp chương trình gọn gàng, mở rộng dễ dàng và phù hợp với các dự án lớn.
* **Phát triển giao diện người dùng (GUI):**  
  Áp dụng thư viện như Tkinter hoặc PyQt để tạo giao diện đồ họa, giúp người dùng dễ thao tác hơn so với dòng lệnh.
* **Kết nối với cơ sở dữ liệu:**  
  Sử dụng SQLite hoặc MySQL để quản lý dữ liệu ở quy mô lớn, đảm bảo hiệu quả và tính nhất quán cao hơn so với lưu file CSV.
* **Thêm các chức năng nâng cao:**
  + Tìm kiếm theo khoảng giá hoặc theo danh mục sản phẩm.
  + Thống kê nâng cao và xuất biểu đồ.
  + Hệ thống đăng nhập và phân quyền người dùng.
  + Tự động lưu sau mỗi thay đổi để tránh mất dữ liệu.
  + Hỗ trợ xuất báo cáo ra file PDF hoặc Excel.

**3. Kết luận**

Đề tài “**Quản lý sản phẩm**” là một ứng dụng nhỏ nhưng có tính thực tiễn rất cao, phù hợp với nhiều mô hình bán hàng, quản lý kho và cửa hàng nhỏ lẻ. Trong quá trình xây dựng chương trình, em đã có cơ hội vận dụng tổng hợp nhiều kiến thức đã học trong môn Lập trình Python như: thao tác với danh sách, xử lý chuỗi, kiểm tra điều kiện, định dạng hiển thị, sử dụng thư viện csv, và đặc biệt là xử lý lỗi để nâng cao tính ổn định của chương trình.

Ngoài ra, em cũng học được cách xây dựng cấu trúc chương trình theo hướng rõ ràng, dễ bảo trì, tách biệt các chức năng thành các hàm riêng biệt và đảm bảo khả năng mở rộng trong tương lai. Việc sử dụng file .csv để lưu trữ và đọc dữ liệu giúp chương trình có khả năng hoạt động lâu dài, giữ được thông tin trong nhiều lần chạy mà không cần nhập lại từ đầu.

Bên cạnh những kỹ thuật đã áp dụng, quá trình thực hiện đề tài cũng giúp em hiểu rõ hơn về quy trình xây dựng một phần mềm thực tế — từ việc xác định yêu cầu, thiết kế chức năng, đến cài đặt, kiểm thử và hoàn thiện sản phẩm. Đây là một trải nghiệm quý báu, giúp em rèn luyện tư duy lập trình logic và cẩn thận hơn trong quá trình viết mã.

Tuy vẫn còn một số hạn chế do chưa áp dụng các kỹ thuật nâng cao như lập trình hướng đối tượng hay giao diện đồ họa, nhưng em tin rằng sản phẩm hiện tại đã đáp ứng tốt các yêu cầu cơ bản của đề tài. Trong thời gian tới, nếu có cơ hội, em sẽ tiếp tục nâng cấp chương trình theo các hướng đã đề ra, nhằm hoàn thiện hơn cả về giao diện lẫn chức năng.

Qua bài tập lớn này, em không chỉ củng cố kỹ năng lập trình mà còn rèn luyện được tính kiên nhẫn, sự tỉ mỉ và kỹ năng giải quyết vấn đề – những yếu tố rất quan trọng trong ngành công nghệ thông tin.

**Mã nguồn chương trình**

# Nguyễn Viết Hoàng Vũ-BTL

san\_pham\_list = []

def hien\_thi\_menu():

    print("\n--- QUẢN LÝ SẢN PHẨM ---")

    print("1. Thêm sản phẩm")

    print("2. Hiển thị danh sách sản phẩm")

    print("3. Tìm sản phẩm theo mã")

    print("4. Cập nhật sản phẩm")

    print("5. Xóa sản phẩm")

    print("6. Sắp xếp theo giá")

    print("7. Tìm sản phẩm có giá cao nhất")

    print("8. Tìm sản phẩm có giá thấp nhất")

    print("9. Tìm sản phẩm theo tên")

    print("10. Lọc sản phẩm")

    print("11. Lưu danh sách vào file")

    print("12. Đọc danh sách từ file")

    print("13. Hiển thị sản phẩm lưu trong file")

    print("14. Thống kê kho hàng")

    print("0. Thoát")

def them\_san\_pham():

    ma = input("Nhập mã sản phẩm: ").strip()

    if not ma:

        return print("Lỗi: Mã sản phẩm không được để trống.")

    if any(sp["ma"] == ma for sp in san\_pham\_list):

        return print("Lỗi: Mã sản phẩm đã tồn tại.")

    ten = input("Nhập tên sản phẩm: ").strip()

    if not ten:

        return print("Lỗi: Tên sản phẩm không được để trống.")

    try:

        so\_luong = int(input("Nhập số lượng tồn: ").strip())

        if so\_luong < 0:

            return print("Lỗi: Số lượng không được âm.")

    except ValueError:

        return print("Lỗi: Số lượng phải là số nguyên.")

    try:

        gia = float(input("Nhập giá bán hiện tại: ").strip())

        if gia < 0:

            return print("Lỗi: Giá không được âm.")

    except ValueError:

        return print("Lỗi: Giá phải là số thực.")

    san\_pham\_list.append({"ma": ma, "ten": ten, "so\_luong": so\_luong, "gia": gia})

    print("Thêm sản phẩm thành công.")

def hien\_thi\_danh\_sach():

    if not san\_pham\_list:

        print("Danh sách sản phẩm trống.")

        return

    print("\n{:<10} {:<25} {:<15} {:<15}".format("Mã SP", "Tên sản phẩm", "Số lượng", "Giá bán (VND)"))

    for sp in san\_pham\_list:

        print("{:<10} {:<25} {:<15} {:<15,.0f}".format(sp["ma"], sp["ten"], sp["so\_luong"], sp["gia"]))

def tim\_san\_pham():

    ma = input("Nhập mã sản phẩm cần tìm: ")

    for sp in san\_pham\_list:

        if sp["ma"] == ma:

            print("Tìm thấy: ", sp)

            return

    print("Không tìm thấy sản phẩm.")

def cap\_nhat\_san\_pham():

    ma = input("Nhập mã sản phẩm cần cập nhật: ")

    for sp in san\_pham\_list:

        if sp["ma"] == ma:

            sp["so\_luong"] = int(input("Nhập số lượng mới: "))

            sp["gia"] = float(input("Nhập giá bán mới: "))

            print("Đã cập nhật sản phẩm.")

            return

    print("Không tìm thấy sản phẩm.")

def xoa\_san\_pham():

    ma = input("Nhập mã sản phẩm cần xóa: ")

    for sp in san\_pham\_list:

        if sp["ma"] == ma:

            print(f"Đã tìm thấy sản phẩm: {sp['ten']} (SL: {sp['so\_luong']}, Giá: {sp['gia']:,.0f} VND)")

            xac\_nhan = input("Bạn có chắc muốn xóa sản phẩm này? (y/n): ").lower()

            if xac\_nhan == "y":

                san\_pham\_list.remove(sp)

                print("Đã xóa sản phẩm.")

            else:

                print("Hủy xóa sản phẩm.")

            return

    print("Không tìm thấy sản phẩm.")

def sap\_xep\_theo\_gia():

    if not san\_pham\_list:

        print("Danh sách sản phẩm trống.")

        return

    print("\n--- SẮP XẾP SẢN PHẨM THEO GIÁ ---")

    print("1. Giá tăng dần")

    print("2. Giá giảm dần")

    lua\_chon = input("Chọn cách sắp xếp (1-2): ").strip()

    if lua\_chon == "1":

        sorted\_list = sorted(san\_pham\_list, key=lambda x: x["gia"])

        print("\n>>> Danh sách sản phẩm theo giá tăng dần:")

    elif lua\_chon == "2":

        sorted\_list = sorted(san\_pham\_list, key=lambda x: x["gia"], reverse=True)

        print("\n>>> Danh sách sản phẩm theo giá giảm dần:")

    else:

        print(" Lựa chọn không hợp lệ. Vui lòng chọn 1 hoặc 2.")

        return

    print("{:<10} {:<25} {:<15} {:<15}".format("Mã SP", "Tên sản phẩm", "Số lượng", "Giá bán (VND)"))

    for sp in sorted\_list:

        print("{:<10} {:<25} {:<15} {:<15,.0f}".format(sp["ma"], sp["ten"], sp["so\_luong"], sp["gia"]))

def san\_pham\_gia\_cao\_nhat():

    if not san\_pham\_list:

        print("Danh sách sản phẩm trống.")

        return

    sp\_max = max(san\_pham\_list, key=lambda x: x["gia"])

    print("Sản phẩm có giá cao nhất:")

    print(f"Mã: {sp\_max['ma']}, Tên: {sp\_max['ten']}, Giá: {sp\_max['gia']:,.0f} VND")

def san\_pham\_gia\_thap\_nhat():

    if not san\_pham\_list:

        print("Danh sách sản phẩm trống.")

        return

    sp\_min = min(san\_pham\_list, key=lambda x: x["gia"])

    print("Sản phẩm có giá thấp nhất:")

    print(f"Mã: {sp\_min['ma']}, Tên: {sp\_min['ten']}, Giá: {sp\_min['gia']:,.0f} VND")

def tim\_kiem\_theo\_ten():

    ten\_can\_tim = input("Nhập tên sản phẩm cần tìm: ").strip().lower()

    if not ten\_can\_tim:

        print(" Tên sản phẩm không được để trống.")

        return

    ket\_qua = [sp for sp in san\_pham\_list if ten\_can\_tim in sp["ten"].lower()]

    if ket\_qua:

        print(f"\n Tìm thấy {len(ket\_qua)} sản phẩm phù hợp:")

        print("{:<10} {:<25} {:<15} {:<15}".format("Mã SP", "Tên sản phẩm", "Số lượng", "Giá bán (VND)"))

        for sp in ket\_qua:

            print("{:<10} {:<25} {:<15} {:<15,.0f}".format(sp["ma"], sp["ten"], sp["so\_luong"], sp["gia"]))

    else:

        print(" Không tìm thấy sản phẩm nào phù hợp.")

def loc\_san\_pham():

    print("\n--- LỌC SẢN PHẨM ---")

    print("1. Giá lớn hơn một mức")

    print("2. Số lượng thấp hơn một mức")

    lua\_chon = input("Chọn tiêu chí lọc (1-2): ")

    if lua\_chon == "1":

        muc\_gia = float(input("Nhập mức giá tối thiểu: "))

        ket\_qua = [sp for sp in san\_pham\_list if sp["gia"] > muc\_gia]

    elif lua\_chon == "2":

        muc\_so\_luong = int(input("Nhập số lượng tối đa: "))

        ket\_qua = [sp for sp in san\_pham\_list if sp["so\_luong"] < muc\_so\_luong]

    else:

        print("Lựa chọn không hợp lệ.")

        return

    if ket\_qua:

        print("\nKết quả lọc:")

        print("{:<10} {:<25} {:<15} {:<15}".format("Mã SP", "Tên sản phẩm", "Số lượng", "Giá bán (VND)"))

        for sp in ket\_qua:

            print("{:<10} {:<25} {:<15} {:<15,.0f}".format(sp["ma"], sp["ten"], sp["so\_luong"], sp["gia"]))

    else:

        print("Không có sản phẩm nào phù hợp.")

import csv

def luu\_vao\_file():

    try:

        with open("san\_pham.csv", mode="w", newline="", encoding="utf-8") as file:

            writer = csv.DictWriter(file, fieldnames=["ma", "ten", "so\_luong", "gia"])

            writer.writeheader()

            writer.writerows(san\_pham\_list)

        print(" Đã lưu danh sách sản phẩm vào file 'san\_pham.csv'.")

    except Exception as e:

        print(f" Lỗi khi lưu file: {e}")

def doc\_tu\_file():

    try:

        with open("san\_pham.csv", mode="r", encoding="utf-8") as file:

            reader = csv.DictReader(file)

            san\_pham\_list.clear()

            for row in reader:

                try:

                    row["so\_luong"] = int(row["so\_luong"])

                    row["gia"] = float(row["gia"])

                    san\_pham\_list.append(row)

                except ValueError:

                    print(f" Bỏ qua dòng lỗi dữ liệu: {row}")

        print(f" Đã đọc {len(san\_pham\_list)} sản phẩm từ file.")

    except FileNotFoundError:

        print(" Chưa có file dữ liệu.")

    except Exception as e:

        print(f" Lỗi khi đọc file: {e}")

def hien\_thi\_file():

    try:

        with open("san\_pham.csv", mode="r", encoding="utf-8") as file:

            reader = csv.DictReader(file)

            danh\_sach = list(reader)

            if not danh\_sach:

                print("File không có dữ liệu.")

                return

            print("\n--- DANH SÁCH TRONG FILE ---")

            print("{:<10} {:<25} {:<15} {:<15}".format("Mã SP", "Tên sản phẩm", "Số lượng", "Giá bán (VND)"))

            for sp in danh\_sach:

                try:

                    print("{:<10} {:<25} {:<15} {:<15,.0f}".format(

                        sp["ma"], sp["ten"], int(sp["so\_luong"]), float(sp["gia"])

                    ))

                except:

                    print(f"Lỗi dữ liệu ở dòng: {sp}")

    except FileNotFoundError:

        print("Chưa có file dữ liệu.")

    except Exception as e:

        print(f"Lỗi khi đọc file: {e}")

def thong\_ke\_kho():

    if not san\_pham\_list:

        print("Danh sách sản phẩm trống.")

        return

    tong\_so\_luong = sum(sp["so\_luong"] for sp in san\_pham\_list)

    tong\_gia\_tri = sum(sp["so\_luong"] \* sp["gia"] for sp in san\_pham\_list)

    print(f"Tổng số lượng sản phẩm: {tong\_so\_luong}")

    print(f"Tổng giá trị kho hàng: {tong\_gia\_tri:,.0f} VND")

# Vòng lặp chính

while True:

    hien\_thi\_menu()

    lua\_chon = input("Chọn chức năng (0-13): ")

    if lua\_chon == "1":

        them\_san\_pham()

    elif lua\_chon == "2":

        hien\_thi\_danh\_sach()

    elif lua\_chon == "3":

        tim\_san\_pham()

    elif lua\_chon == "4":

        cap\_nhat\_san\_pham()

    elif lua\_chon == "5":

        xoa\_san\_pham()

    elif lua\_chon == "6":

        sap\_xep\_theo\_gia()

    elif lua\_chon == "7":

        san\_pham\_gia\_cao\_nhat()

    elif lua\_chon == "8":

        san\_pham\_gia\_thap\_nhat()

    elif lua\_chon == "9":

        tim\_kiem\_theo\_ten()

    elif lua\_chon == "10":

        loc\_san\_pham()

    elif lua\_chon == "11":

        luu\_vao\_file()

    elif lua\_chon == "12":

        doc\_tu\_file()

    elif lua\_chon == "13":

        hien\_thi\_file()

    elif lua\_chon == "14":

        thong\_ke\_kho()

    elif lua\_chon == "0":

        print("Thoát chương trình.")

        break

    else:

        print("Lựa chọn không hợp lệ. Vui lòng chọn lại.")