**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH PYTHON**

**ĐỀ TÀI: Quản lý học từ vựng tiếng Anh**

**Giảng viên: ThS. Lê Tuấn Anh**

**Sinh viên thực hiện:**

1. **Vũ Thị Ngọc Hân – 1871020216**
2. **Nghiêm Minh Phúc – 1871020457**

**Hà Nội, năm 2025**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH PYTHON**

**ĐỀ TÀI: Quản lý học từ vựng tiếng Anh**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TT | Mã sv | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1871020216 | Vũ Thị Ngọc Hân | 30/10/2006 |  |  |
| 2 | 1871020457 | Nghiêm Minh Phúc | 21/06/2006 |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM THI 1**  **ThS. Lê Tuấn Anh** | **CÁN BỘ CHẤM THI 2** |

### 

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Trong bối cảnh hội nhập quốc tế, việc làm chủ tiếng Anh không chỉ là lợi thế mà còn là nhu cầu thiết yếu trong học tập và nghề nghiệp. Từ vựng là nền tảng cốt lõi giúp người học tự tin giao tiếp và tiếp thu kiến thức. Tuy nhiên, việc quản lý và ghi nhớ hàng trăm, thậm chí hàng ngàn từ tiếng Anh thường gặp khó khăn: thiếu hệ thống, không theo dõi được tiến độ và kết quả ôn tập.

Đề tài “Quản lý học từ vựng tiếng Anh” được triển khai nhằm khắc phục những hạn chế trên thông qua một ứng dụng console đơn giản, linh hoạt, xây dựng trên ngôn ngữ Python. Ứng dụng cho phép:

- Tổ chức từ vựng theo chủ đề, dễ dàng thêm, sửa, xóa từ mới;

- Luyện tập ngẫu nhiên và cập nhật điểm số, ghi lại lịch sử trả lời đúng/sai;

- Thống kê tiến trình theo điểm, hiển thị những từ sai nhiều nhất hoặc có điểm thấp;

- Xuất/nhập dữ liệu dưới định dạng JSON, đảm bảo khả năng mở rộng và chia sẻ.

Toàn bộ chương trình gồm 20 chức năng, vận dụng nhuần nhuyễn cấu trúc điều kiện, vòng lặp, hàm, danh sách, từ điển, đọc/ghi file và thư viện random, json. Qua quá trình phát triển, nhóm chúng em không chỉ củng cố kiến thức lập trình mà còn tích lũy kinh nghiệm thiết kế phần mềm hỗ trợ học tập, hướng tới các giải pháp công nghệ giáo dục trong tương lai.

Chúng em xin gửi lời cảm ơn sâu sắc đến thầy cô và bạn bè đã góp ý, hỗ trợ trong quá trình hoàn thiện đề tài. Mọi đóng góp để nâng cao chất lượng sản phẩm sẽ được ghi nhận và triển khai ở những phiên bản tiếp theo.

Hà Nội, năm 2025

**Nhóm sinh viên thực hiện:**

Vũ Thị Ngọc Hân - 1871020216

Nghiêm Minh Phúc- 1871020457

**MỤC LỤC**

[**CHƯƠNG 1. GIỚI THIỆU VỀ MỘT SỐ KHÁI NIỆM TRONG PYTHON** 5](#_Toc202476598)

[**1.1. Biến và Kiểu Dữ Liệu.** 5](#_Toc202476599)

[***1.1.1 Biến trong Python.*** 5](#_Toc202476600)

[***1.1.2 Kiểu Dữ Liệu Cơ Bản.*** 5](#_Toc202476601)

[***1.1.3 Toán tử và Phép Toán.*** 6](#_Toc202476602)

[**1.2 Cấu Trúc Điều Khiển.** 6](#_Toc202476603)

[***1.2.1 Câu Lệnh Rẽ Nhánh.*** 6](#_Toc202476604)

[***1.2.2 Vòng Lặp.*** 7](#_Toc202476605)

[**1.3 Hàm và Module.** 7](#_Toc202476606)

[***1.3.1 Hàm (Function).*** 7](#_Toc202476607)

[***1.3.2 Module.*** 8](#_Toc202476608)

[**1.4 Xử Lý Tệp Tin (File I/O).** 8](#_Toc202476609)

[**1.5 Xử Lý Ngoại Lệ (Exception Handling)** 9](#_Toc202476610)

[**CHƯƠNG 2. GIỚI THIỆU, MÔ TẢ ĐỀ TÀI** 10](#_Toc202476611)

[**2.1 Bối cảnh và Lý do chọn đề tài.** 10](#_Toc202476612)

[*2.1.1 Bối cảnh nghiên cứu.* 10](#_Toc202476613)

[*2.1.2 Lý do chọn đề tài.* 10](#_Toc202476614)

[**2.2 Mục tiêu và Đối tượng nghiên cứu.** 10](#_Toc202476615)

[*2.2.1 Mục tiêu nghiên cứu.* 10](#_Toc202476616)

[*2.2.2 Đối tượng nghiên cứu.* 10](#_Toc202476617)

[**2.3 Phạm vi nghiên cứu.** 11](#_Toc202476618)

[**2.4 Nội dung và Phương pháp thực hiện.** 11](#_Toc202476619)

[*2.4.1 Phương pháp thu thập yêu cầu.* 11](#_Toc202476620)

[*2.4.2 Phương pháp thiết kế và triển khai.* 11](#_Toc202476621)

[**2.5 Cấu trúc báo cáo.** 11](#_Toc202476622)

[**CHƯƠNG 3. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG** 13](#_Toc202476623)

[**3.1 Tổng quan chức năng chương trình.** 13](#_Toc202476624)

[**3.2 Cấu trúc dữ liệu trong file vocab\_data.json.** 15](#_Toc202476625)

[**3.3 Phân tích mã nguồn chính trong file main.py.** 16](#_Toc202476626)

[*3.3.1 Khởi tạo dữ liệu và lưu trữ.* 16](#_Toc202476627)

[*3.3.2 Các chức năng quản lý từ vựng cơ bản* 16](#_Toc202476628)

[*3.3.4 Các chức năng thống kê và báo cáo.* 17](#_Toc202476629)

[*3.3.5 Giao diện menu chính và điều khiển chương trình.* 19](#_Toc202476630)

[**HƯỚNG PHÁT TRIỂN** 20](#_Toc202476631)

[**KẾT LUẬN** 21](#_Toc202476632)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 22](#_Toc202476633)

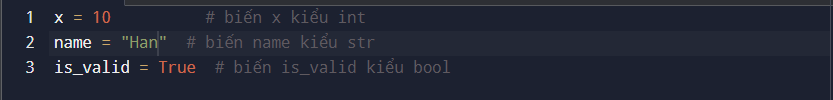
**CHƯƠNG 1. GIỚI THIỆU VỀ MỘT SỐ KHÁI NIỆM TRONG PYTHON**

**1.1. Biến và Kiểu Dữ Liệu.**

***1.1.1 Biến trong Python.***

Trong Python, biến là tên định danh dùng để tham chiếu tới vùng nhớ lưu trữ giá trị. Khác với nhiều ngôn ngữ khác, Python không yêu cầu khai báo trước kiểu dữ liệu khi tạo biến. Khi gán giá trị cho biến, Python sẽ tự động xác định và gán kiểu cho biến dựa trên giá trị đó.

Ví dụ:



**Quy tắc đặt tên biến:**

- Bắt đầu bằng chữ cái (a–z, A–Z) hoặc dấu gạch dưới (\_), không bắt đầu bằng chữ số.

- Chỉ chứa chữ cái, chữ số và dấu gạch dưới, không chứa khoảng trắng hoặc ký tự đặc biệt.

- Phân biệt chữ hoa – chữ thường (case-sensitive).

- Không sử dụng các từ khóa (keywords) của Python như if, for, def,...

***1.1.2 Kiểu Dữ Liệu Cơ Bản.***

Python hỗ trợ các kiểu dữ liệu nguyên thủy và cấu trúc dữ liệu tổng hợp:

**int**: Số nguyên, ví dụ -3, 0, 42.

**float**: Số thực, ví dụ 3.14, -0.001.

**str**: Chuỗi ký tự Unicode, ví dụ "Hello, Python!".

**bool**: Giá trị logic True hoặc False.

**list**: Danh sách có thứ tự, có thể chứa nhiều kiểu khác nhau, ví dụ [1, "a", True].

**tuple**: Bộ giá trị bất biến (immutable), ví dụ (1, 2, 3).

**dict**: Từ điển ánh xạ khóa → giá trị, ví dụ {"a": 1, "b": 2}.

**set**: Tập hợp các phần tử duy nhất, không theo thứ tự, ví dụ {1, 2, 3}.

**Chuyển đổi kiểu dữ liệu:**

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

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

***1.1.3 Toán tử và Phép Toán.***

- Toán tử số học: +, -, \*, /, // (chia lấy phần nguyên), % (chia lấy dư), \*\* (lũy thừa).

- Toán tử so sánh: ==, !=, <, >, <=, >=.

- Toán tử logic: and, or, not.

- Toán tử gán: =, +=, -=, \*=, /=...

Ví dụ:

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, ảnh chụp màn hình

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

**1.2 Cấu Trúc Điều Khiển.**

Cấu trúc điều khiển cho phép chương trình thực thi khác nhau tùy vào điều kiện hoặc thực hiện lặp lại khối lệnh.

***1.2.1 Câu Lệnh Rẽ Nhánh.***

- if – kiểm tra điều kiện, nếu đúng thì thực hiện khối lệnh.

- if...else – nếu điều kiện sai, thực hiện khối lệnh thay thế.

- if...elif...else – kiểm tra nhiều điều kiện theo thứ tự.

Ví dụ: Ảnh có chứa văn bản, ảnh chụp màn hình, Phần mềm đa phương tiện, phần mềm

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

***1.2.2 Vòng Lặp.***

- for – lặp qua các phần tử của đối tượng tuần tự (list, tuple, dict, string, range).

- while – lặp khi điều kiện còn đúng.

Ví dụ for:



Ví dụ while:

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ, Phần mềm đa phương tiện

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

**Điều khiển vòng lặp:**

- break – thoát vòng lặp ngay lập tức.

- continue – bỏ qua phần còn lại của vòng lặp hiện tại và chuyển sang lần lặp tiếp theo.

**1.3 Hàm và Module.**

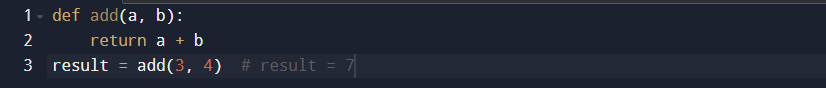
***1.3.1 Hàm (Function).***

Hàm là khối mã được định nghĩa để thực hiện một nhiệm vụ nhất định, có thể tái sử dụng nhiều lần. Sử dụng từ khóa def để khai báo hàm.

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

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

Hàm có thể nhận đối số, trả về giá trị qua từ khóa return.



***1.3.2 Module.***

Module là tệp chứa các hàm, lớp, biến được tái sử dụng. Sử dụng import để gọi module có sẵn hoặc module tự tạo.

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, Phông chữ

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

**1.4 Xử Lý Tệp Tin (File I/O).**

Python hỗ trợ thao tác đọc/ghi tệp dễ dàng qua hàm open().

**Ảnh có chứa văn bản, Phông chữ, Phần mềm đa phương tiện, ảnh chụp màn hình

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

## **1.5 Xử Lý Ngoại Lệ (Exception Handling)**

Sử dụng khối try...except để bắt và xử lý lỗi trong quá trình chạy chương trình, tránh làm chương trình dừng đột ngột.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, Phần mềm đa phương tiện

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

**CHƯƠNG 2. GIỚI THIỆU, MÔ TẢ ĐỀ TÀI**

## **2.1 Bối cảnh và Lý do chọn đề tài.**

### *2.1.1 Bối cảnh nghiên cứu.*

Trong những năm gần đây, việc học tiếng Anh ngày càng trở nên phổ biến, đặc biệt trong bối cảnh hội nhập toàn cầu và cách mạng 4.0. Các phương pháp học truyền thống như sổ tay, flashcard giấy tuy hiệu quả nhưng khó quản lý khối lượng lớn từ vựng, thiếu tính tương tác và không lưu trữ được lịch sử ôn tập. Ứng dụng công nghệ, đặc biệt là ngôn ngữ lập trình Python, mang đến giải pháp số hóa, tự động hóa quá trình quản lý và học từ vựng, giúp người học tiết kiệm thời gian và tối ưu hóa kết quả học tập.

### *2.1.2 Lý do chọn đề tài.*

1, Hiệu quả lưu trữ: JSON giúp lưu trữ dữ liệu có cấu trúc, dễ đọc, dễ chỉnh sửa.

2, Tự động hóa: Python cung cấp thư viện sẵn có (json, random, datetime) để nhanh chóng phát triển tính năng tạo bài tập ngẫu nhiên và thống kê kết quả.

3, Tính mở rộng: Thiết kế module hóa, tách riêng chức năng theo hàm, dễ dàng thêm chức năng mới (ví dụ: giao diện GUI, kết nối CSDL).

4, Nhu cầu thực tế: Nhiều người học tiếng Anh mong muốn công cụ cá nhân hóa, theo dõi tiến trình, cảnh báo từ cần ôn tập lại.

## **2.2 Mục tiêu và Đối tượng nghiên cứu.**

### *2.2.1 Mục tiêu nghiên cứu.*

- Xây dựng ứng dụng console Python với đầy đủ chức năng quản lý từ vựng: thêm, sửa, xóa, tra cứu theo chủ đề.

- Triển khai bài tập ôn luyện ngẫu nhiên, cập nhật điểm và theo dõi lịch sử đúng/sai.

- Cung cấp báo cáo tổng quan: thống kê số từ theo điểm, danh sách từ yếu, đề xuất nhóm từ cần tập trung.

- Đảm bảo tính ổn định, khả năng mở rộng và dễ sử dụng cho người học.

### *2.2.2 Đối tượng nghiên cứu.*

**-** Sinh viên, học sinh, người đi làm tự học tiếng Anh có nhu cầu quản lý và hệ thống hóa từ vựng.

- Những người mới bắt đầu hoặc ở trình độ trung cấp muốn cải thiện vốn từ.

## **2.3 Phạm vi nghiên cứu.**

- Giao diện: Ứng dụng console (chạy trên terminal).

- Ngôn ngữ: Python 3.x, không yêu cầu framework phức tạp.

- Dữ liệu: Lưu trữ cục bộ trong file vocab\_data.json.

- Chủ đề: Hiện tại triển khai 10-15 chủ đề thông dụng (động vật, màu sắc, trường học…), dự kiến mở rộng lên 20 chủ đề.

## **2.4 Nội dung và Phương pháp thực hiện.**

### *2.4.1 Phương pháp thu thập yêu cầu.*

- Phỏng vấn nhóm người học tiếng Anh để xác định nhu cầu chức năng.

- Tham khảo các ứng dụng học từ vựng nổi tiếng (Anki, Quizlet) để rút kinh nghiệm UX.

### *2.4.2 Phương pháp thiết kế và triển khai.*

- Thiết kế cấu trúc dữ liệu: Từ điển Python chứa khóa chủ đề, giá trị là danh sách từ (định dạng JSON).

- Phân chia module: Mỗi chức năng (thêm từ, luyện tập, thống kê…) triển khai thành hàm riêng biệt.

- Kiểm thử: Viết kịch bản test thủ công (test edge cases: từ trùng lặp, nhập sai định dạng điểm…).

- Tài liệu hóa: Ghi chú code, viết hướng dẫn sử dụng file README.md.

## **2.5 Cấu trúc báo cáo.**

Chương 1: Giới thiệu các khái niệm cơ bản trong Python.

Chương 2: Giới thiệu và mô tả đề tài.

Chương 3: Phát triển và triển khai ứng dụng.

Kết Luận

Hướng phát triển

Mục tài liệu tham khảo

**CHƯƠNG 3. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG**

**3.1 Tổng quan chức năng chương trình.**

Ứng dụng quản lý từ vựng được thiết kế dưới dạng menu tương tác, bao gồm nhiều chức năng chính để thêm, sửa, xóa và ôn luyện từ vựng. Chương trình cho phép người dùng nhập liệu chủ đề và từ mới, đồng thời lưu trữ kết quả vào file JSON. Các chức năng chính bao gồm:

***Thêm từ vựng:*** Nhập chủ đề, từ tiếng Anh và nghĩa tiếng Việt. Chương trình tạo một đối tượng từ mới có cấu trúc {"từ": <từ>, "nghĩa": <nghĩa>, "điểm": 0, "lịch sử": []} và thêm vào danh sách của chủ đề tương ứng. Nếu chủ đề chưa tồn tại, nó sẽ được tạo mới (hàm them\_tu\_vung()).

***Thêm chủ đề:*** Nhập tên chủ đề mới và tạo ra một mục chủ đề với danh sách từ rỗng. Nếu tên chủ đề đã có sẵn, báo “Chủ đề đã tồn tại” (hàm them\_chu\_de()).

***Xem từ theo chủ đề:*** Hiển thị danh sách từ vựng theo từng chủ đề. Mỗi chủ đề được in ra rồi liệt kê các từ kèm nghĩa và điểm số của chúng. Ví dụ, chủ đề “Động vật” sẽ liệt kê các từ như “cat – con mèo (Điểm: 2)”.

***Hiển thị tất cả từ:*** Liệt kê toàn bộ từ trong tất cả các chủ đề, mỗi từ hiển thị định dạng từ (chủ đề) – nghĩa | Điểm: X (hàm hien\_thi\_tat\_ca\_tu()).

***Tìm kiếm từ:*** Nhập một từ khóa, chương trình tìm trong cả từ và nghĩa. Nếu tìm thấy, in ra vị trí (chủ đề) và thông tin của từ đó (hàm tim\_kiem\_tu()).

***Sửa thông tin từ:*** Nhập từ cần sửa, tìm kiếm theo trường "từ", sau đó cho phép thay đổi nghĩa. Kết quả được lưu vào file JSON (hàm sua\_thong\_tin\_tu()).

***Xóa từ vựng:*** Nhập từ cần xóa, chương trình tìm kiếm và loại bỏ phần tử đó khỏi danh sách. Dữ liệu sau khi xóa được lưu lại (hàm xoa\_tu\_vung()).

***Luyện tập từ ngẫu nhiên:*** Chọn ngẫu nhiên một từ bất kỳ từ tất cả chủ đề (hàm luyen\_tap\_ngau\_nhien()). Người dùng nhập nghĩa, nếu trả lời đúng thì in “Đúng rồi!”, tăng điểm của từ lên 1 và thêm True vào mảng "lịch sử"; nếu sai thì in đáp án đúng và thêm False vào "lịch sử". Dữ liệu được cập nhật ngay sau mỗi lần ôn luyện.

***Cập nhật điểm ôn luyện:*** Cho phép nhập thủ công điểm mới cho một từ. Chương trình tìm từ tương ứng và gán giá trị mới cho "điểm" rồi lưu file (hàm cap\_nhat\_diem()).

***Hiển thị lịch sử ôn luyện:*** Nhập từ cần xem, chương trình liệt kê mảng "lịch sử" (mảng giá trị True/False) của từ đó để người dùng đánh giá số lần trả lời đúng/sai (hàm hien\_thi\_lich\_su\_on\_luyen()).

Thống kê và báo cáo:

*- Thống kê theo điểm*: Đếm số lượng từ có cùng mức điểm và in kết quả (hàm thong\_ke\_diem\_on()).

*- Từ sai nhiều nhất*: Đếm số lần sai (giá trị False trong "lịch sử") của mỗi từ, sắp xếp giảm dần và in 5 từ có số sai nhiều nhất (hàm hien\_thi\_tu\_sai\_nhieu()).

*- Từ có điểm = 0*: Liệt kê tất cả từ có điểm bằng 0, hiển thị cùng chủ đề và nghĩa (hàm hien\_thi\_tu\_diem\_0()).

*- Từ điểm cao nhất*: Tìm giá trị điểm lớn nhất và in các từ đạt điểm đó (hàm hien\_thi\_diem\_cao\_nhat()).

*- Sắp xếp theo alphabet*: In danh sách tất cả từ theo thứ tự chữ cái tăng dần (hàm sap\_xep\_tu\_theo\_alphabet()).

*- Từ có điểm thấp hơn*: Nhập ngưỡng điểm, in tất cả từ có điểm nhỏ hơn hoặc bằng ngưỡng đó (hàm hien\_thi\_tu\_diem\_thap()).

1. Xuất dữ liệu ra file: Cho phép ghi toàn bộ dữ liệu sang một file JSON khác do người dùng đặt tên (hàm xuat\_du\_lieu\_ra\_file()).
2. Đếm số từ trong mỗi chủ đề: Thống kê số lượng từ cho từng chủ đề và in ra (hàm dem\_so\_tu\_trong\_chu\_de()).
3. Đặt lại điểm tất cả từ: Gán lại điểm = 0 cho mọi từ trong dữ liệu và lưu lại (hàm dat\_lai\_diem\_tat\_ca()).
4. Lưu dữ liệu thủ công và Thoát: Menu có chức năng lưu thủ công dữ liệu hiện tại (luu\_du\_lieu\_vao\_file()) và kết thúc chương trình khi chọn “Thoát” (hàm menu()).

Các tính năng trên được tổ chức rõ ràng trong menu chính của chương trình, với 21 lựa chọn cụ thể từ 1 đến 21. Người dùng nhập lựa chọn tương ứng, chương trình sẽ gọi hàm xử lý tương ứng như đã liệt kê ở trên.

**3.2 Cấu trúc dữ liệu trong file vocab\_data.json.**

File vocab\_data.json lưu trữ toàn bộ dữ liệu từ vựng theo cấu trúc JSON. Ở cấp cao nhất, mỗi khóa (key) là tên một chủ đề (ví dụ "Động vật", "Thời tiết", "Màu sắc", v.v.), và giá trị tương ứng là một mảng (list) các từ vựng thuộc chủ đề đó. Mỗi phần tử trong mảng này là một đối tượng (dict) chứa thông tin chi tiết của một từ. Cấu trúc mẫu của một từ vựng như sau:

* "từ": Từ vựng tiếng Anh (loại string).
* "nghĩa": Nghĩa tiếng Việt của từ (string).
* "điểm": Số điểm (số lần trả lời đúng trong ôn luyện), kiểu số nguyên (int).
* "lịch sử": Mảng các giá trị true/false (bool), lần lượt biểu thị việc ôn luyện mỗi lần có đúng (true) hay sai (false).

Ví dụ, trong mục chủ đề "Động vật" có một từ vựng "cat" được lưu như sau:

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

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

Như vậy, một chủ đề có thể chứa nhiều từ vựng, mỗi từ có đầy đủ trường thông tin như trên. Các ví dụ tương tự được lưu ở các chủ đề khác nhau (ví dụ "Thời tiết", "Màu sắc", "Trường học", v.v.) trong file JSON.

Việc lưu trữ theo cấu trúc này giúp chương trình dễ dàng truy cập và xử lý: chỉ cần đọc file JSON để nạp vào biến vocab\_data (kiểu dict), sau đó tìm theo từ khóa chủ đề hoặc duyệt qua toàn bộ các phần tử để thực hiện chức năng tương ứng.

## **3.3 Phân tích mã nguồn chính trong file main.py.**

Mã nguồn chính main.py được tổ chức gồm các hàm riêng biệt tương ứng với các chức năng của chương trình, và một hàm menu() điều phối luồng chính. Dưới đây là phân tích chi tiết các phần quan trọng:

### *3.3.1 Khởi tạo dữ liệu và lưu trữ.*

Nạp dữ liệu từ file: Ngay khi chạy chương trình, ở cuối file main.py có đoạn mã with open("vocab\_data.json", "r", encoding="utf-8") as f: vocab\_data = json.load(f). Đoạn này dùng module json để đọc toàn bộ nội dung file JSON vào biến vocab\_data (kiểu dict). Tất cả các chức năng tiếp theo đều thao tác trên biến vocab\_data này.

Hàm lưu dữ liệu: Bất kỳ thay đổi nào trong vocab\_data đều được ghi trở lại file bằng hàm luu\_du\_lieu\_vao\_file(). Cụ thể, hàm này mở file "vocab\_data.json" ở chế độ ghi và sử dụng json.dump(vocab\_data, f, ensure\_ascii=False, indent=4). Tham số ensure\_ascii=False cho phép ghi đúng ký tự Unicode (gồm tiếng Việt) và indent=4 để định dạng file cho dễ đọc. Ví dụ, sau khi thêm từ hoặc thay đổi điểm, hàm này được gọi để cập nhật file.

### *3.3.2 Các chức năng quản lý từ vựng cơ bản*

***Hàm them\_tu\_vung():*** Người dùng nhập tên chủ đề (chu\_de), từ mới (tu) và nghĩa (nghia). Chương trình tạo biến tu\_moi là một dict với các trường "từ", "nghĩa", và khởi tạo "điểm": 0, "lịch sử": [] (nghĩa là từ mới chưa có lần luyện tập nào). Nếu chủ đề chưa tồn tại trong vocab\_data, hàm sẽ tạo một mục mới với danh sách rỗng trước khi thêm. Sau đó, tu\_moi được thêm vào mảng của chủ đề và gọi luu\_du\_lieu\_vao\_file() để lưu lại. Cuối cùng in ra thông báo đã thêm từ.

***Hàm them\_chu\_de():*** Cho phép tạo một chủ đề mới. Người dùng nhập tên mới (chu\_de). Nếu tên này đã tồn tại trong vocab\_data, chương trình thông báo lỗi. Ngược lại, hàm thêm khóa mới vào vocab\_data với giá trị là một list rỗng rồi lưu file.

***Hàm xem\_tu\_theo\_chu\_de():*** Duyệt qua các khóa (chủ đề) của vocab\_data và in tiêu đề của mỗi chủ đề, sau đó in lần lượt từng từ trong chủ đề đó. Ví dụ, với chủ đề "Động vật", nó sẽ in == Động vật == rồi in các từ cùng nghĩa và điểm. (Lưu ý: ý đồ của code là lồng hai vòng for, tuy indent trong ví dụ có thể chưa đúng nhưng kết quả mong muốn là liệt kê mọi từ theo từng chủ đề.)

***Hàm hien\_thi\_tat\_ca\_tu():*** In tất cả từ trong tất cả các chủ đề dưới dạng từng dòng: từ (chủ đề) - nghĩa | Điểm: X. Chức năng này dùng hai vòng lặp for để quét qua mọi từ trong vocab\_data.

***Hàm tim\_kiem\_tu():*** Cho phép người dùng nhập một chuỗi tu\_khoa. Chương trình lặp qua tất cả các từ, nếu tu\_khoa là phần con của trường "từ" hoặc "nghĩa" thì in ra thông tin từ đó cùng chủ đề, báo là tìm thấy. Cơ chế tìm kiếm đơn giản, không phân biệt hoa thường.

***Hàm sua\_thong\_tin\_tu():*** Nhập tên từ cần sửa (tu\_khoa), chương trình tìm từ đó trong vocab\_data. Nếu tìm được (so sánh chính xác với "từ"), hàm yêu cầu nhập nghĩa mới và gán vào trường "nghĩa", sau đó lưu file. Nếu không tìm thấy, in thông báo lỗi.

***Hàm xoa\_tu\_vung():*** Tương tự tìm kiếm như trên. Nếu tìm thấy một từ khớp với tu\_khoa, hàm sẽ gọi ds.remove(item) để xóa phần tử đó khỏi danh sách của chủ đề, rồi lưu dữ liệu. Nếu không tìm thấy từ, in thông báo.

***3.3.3 Chức năng luyện tập và cập nhật điểm.***

***Hàm luyen\_tap\_ngau\_nhien():*** Hàm này tạo một danh sách all\_words gồm tất cả cặp (chu\_de, item) từ vocab\_data. Sau đó, dùng random.choice(all\_words) để lấy ngẫu nhiên một từ (và biết luôn chủ đề). Chương trình hỏi: Nghĩa của từ '<từ>' là gì? và đọc câu trả lời. Nếu câu trả lời (sau khi chuyển về chữ thường và bỏ khoảng trắng) trùng với "nghĩa" của từ, in “Đúng rồi!” và tăng điểm của từ lên 1, đồng thời thêm giá trị True vào "lịch sử". Ngược lại, in ra đáp án đúng và thêm False vào "lịch sử". Sau mỗi lần luyện tập, luu\_du\_lieu\_vao\_file() được gọi để lưu lại thay đổi. Cơ chế "lịch sử" là một danh sách ghi lại đúng/sai của từng lần luyện tập, cho phép thống kê sau này.

***Hàm cap\_nhat\_diem():*** Nếu người dùng muốn cập nhật điểm thủ công, hàm này cho phép nhập từ và điểm mới. Tương tự tìm qua các từ, nếu tìm được, gán giá trị mới cho trường "điểm" và lưu file. Nếu không tìm thấy từ đó, thông báo lỗi.

***Hàm hien\_thi\_lich\_su\_on\_luyen():*** Nhập từ cần xem. Nếu tìm được, in mảng "lịch sử" của từ đó (ví dụ [True, False, True]), người dùng có thể tự tính số lần đúng/sai.

### *3.3.4 Các chức năng thống kê và báo cáo.*

Các hàm ở phần này thực hiện các phép tính và in ra thông tin tổng hợp từ vocab\_data:

***Hàm thong\_ke\_diem\_on():*** Tính số lượng từ ứng với mỗi giá trị điểm. Đầu tiên khởi tạo thong\_ke = {}, sau đó duyệt qua tất cả từ để đếm số lần xuất hiện của mỗi điểm và lưu vào từ điển thong\_ke[d] = count. Cuối cùng in ra kết quả đã đếm (theo thứ tự điểm tăng dần).

***Hàm hien\_thi\_tu\_sai\_nhieu():***Đếm số lần sai của mỗi từ. Với mỗi từ, nó lấy item["lịch sử"].count(False) để biết số lần trả lời sai. Các kết quả này được lưu vào danh sách tu\_sai\_nhieu dưới dạng tuple (so\_sai, từ, chủ đề). Sau khi thu thập xong, danh sách được sắp xếp giảm dần theo so\_sai, rồi in 5 phần tử đầu (các từ sai nhiều nhất).

***Hàm hien\_thi\_tu\_diem\_0():*** Duyệt qua tất cả từ, nếu item["điểm"] == 0 thì in ra từ đó kèm nghĩa và chủ đề.

***Hàm hien\_thi\_diem\_cao\_nhat():*** Tìm giá trị điểm lớn nhất bằng cách: max\_diem = max(item["điểm"] for ds in vocab\_data.values() for item in ds). Sau đó, lại duyệt qua các từ một lần nữa, in các từ có "điểm" == max\_diem cùng chủ đề.

***Hàm sap\_xep\_tu\_theo\_alphabet():*** Thu thập tất cả từ vào một list all\_words dạng (từ, chủ đề, nghĩa), rồi dùng sorted(all\_words) để sắp xếp theo thứ tự chữ cái của từ. Cuối cùng in lần lượt từng phần tử đã sắp xếp.

***Hàm hien\_thi\_tu\_diem\_thap():*** Cho phép nhập một ngưỡng điểm nguong. Chương trình in ra các từ có điểm <= nguong cùng thông tin chi tiết. Nếu nhập sai kiểu dữ liệu (ví dụ không phải số), nó sẽ thông báo “Giá trị không hợp lệ.” và thoát hàm (xem khối try/except).

***Hàm xuat\_du\_lieu\_ra\_file():*** Người dùng nhập tên file (phải có phần mở rộng .json). Chương trình ghi toàn bộ vocab\_data vào file mới đó, định dạng giống như lưu bình thường, giúp sao lưu hoặc chia sẻ dữ liệu.

***Hàm dem\_so\_tu\_trong\_chu\_de():*** Duyệt từng chủ đề và in số phần tử (số từ) trong mỗi chủ đề bằng len(ds).

***Hàm dat\_lai\_diem\_tat\_ca():*** Duyệt qua mọi từ và đặt lại item["điểm"] = 0 cho tất cả. Sau đó gọi luu\_du\_lieu\_vao\_file() để lưu kết quả và thông báo hoàn thành.

### *3.3.5 Giao diện menu chính và điều khiển chương trình.*

**Hàm menu():** hiển thị một bảng tùy chọn cho người dùng với 21 mục lựa chọn từ 1 đến 21. Sau khi in menu, chương trình chờ người dùng nhập số chức năng. Đoạn if/elif kiểm tra giá trị choice và gọi hàm tương ứng (ví dụ nhập "1" thì gọi them\_tu\_vung(), "8" gọi luyen\_tap\_ngau\_nhien(), v.v. theo các dòng lệnh). Hai lựa chọn cuối cùng là Lưu dữ liệu thủ công (số 20) và Thoát chương trình (số 21). Khi chọn “Thoát”, chương trình in “Tạm biệt!” và dùng break để thoát vòng lặp, kết thúc chương trình.

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

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

Tóm lại, cấu trúc menu cho phép người dùng điều khiển mọi chức năng đã phân tích ở trên. Vòng lặp while True liên tục hiển thị menu và xử lý lệnh cho đến khi người dùng chọn thoát.

**LINK GITHUB**

https://github.com/phucnghiem3010/phuc2006

# **HƯỚNG PHÁT TRIỂN**

Trong tương lai, chương trình có thể được mở rộng và nâng cấp theo các hướng sau:

1. Xây dựng giao diện đồ họa (GUI): Sử dụng thư viện như Tkinter, PyQt hoặc Kivy để tăng tính thân thiện với người dùng và trải nghiệm trực quan hơn.
2. Phát triển phiên bản web hoặc ứng dụng di động: Triển khai chương trình dưới dạng ứng dụng web (Flask, Django) hoặc mobile app (React Native, Flutter) để người dùng có thể truy cập từ mọi nơi.
3. Tích hợp phát âm và âm thanh: Sử dụng API từ điển hoặc text-to-speech để hỗ trợ người học nghe cách phát âm của từ.
4. Hệ thống bài kiểm tra tự động và báo cáo chi tiết: Tự động tạo bài kiểm tra từ vựng theo từng chủ đề hoặc trình độ, xuất báo cáo quá trình học theo tuần/tháng.
5. Kết nối cơ sở dữ liệu SQL hoặc NoSQL: Lưu trữ dữ liệu tập trung, hỗ trợ nhiều người dùng hoặc phân tích dữ liệu ở quy mô lớn hơn.

**KẾT LUẬN**

Qua quá trình tìm hiểu và phát triển, chương trình quản lý từ vựng tiếng Anh đã được xây dựng hoàn chỉnh với nhiều chức năng hữu ích, hỗ trợ người học trong việc ghi nhớ, ôn luyện và theo dõi quá trình học từ vựng một cách hiệu quả.

Chương trình được viết bằng ngôn ngữ Python, sử dụng cấu trúc dữ liệu linh hoạt như dict, list kết hợp với file JSON để lưu trữ từ vựng theo từng chủ đề. Các chức năng như thêm, sửa, xóa từ; luyện tập ngẫu nhiên; thống kê kết quả; xuất dữ liệu; và nhiều tiện ích khác đã giúp tạo nên một hệ thống hỗ trợ học tập từ vựng mang tính ứng dụng thực tiễn cao.

Thông qua chức năng lịch sử ôn luyện và thống kê điểm, người học có thể đánh giá năng lực, xác định những từ còn yếu để tập trung ôn lại. Giao diện dòng lệnh đơn giản, dễ sử dụng, phù hợp cho cả người mới bắt đầu học lập trình hoặc người học tiếng Anh ở mọi trình độ.

Tuy chương trình còn ở mức cơ bản và chạy trên môi trường console, nhưng đây là nền tảng tốt để tiếp tục phát triển lên phiên bản giao diện đồ họa (GUI) hoặc tích hợp với cơ sở dữ liệu và các công nghệ web để mở rộng quy mô và tính tiện dụng.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Fundamentals of Programming Python, Richard L. Halterman. Southern Adventist University 2019.
2. Nguyễn Ngọc Giang, Phan Xuân Vọng, Nguyễn Quốc Anh, Đường Vào Lập Trình Python, NXB Đại Học Quốc Gia, 2020
3. Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, Inc, 2022
4. Th.S Lê Tuấn Anh, khoa CNTT, Trường Đại học Đại Nam.
5. Thầy Nguyễn Duy Khương, khoa CNTT, Trường Đại học Đại Nam.
6. Real Python. *Reading and Writing JSON to a File in Python*. Truy cập tại: <https://realpython.com/python-json/>
7. Lập trình Python cơ bản – NXB Trẻ, 2021.