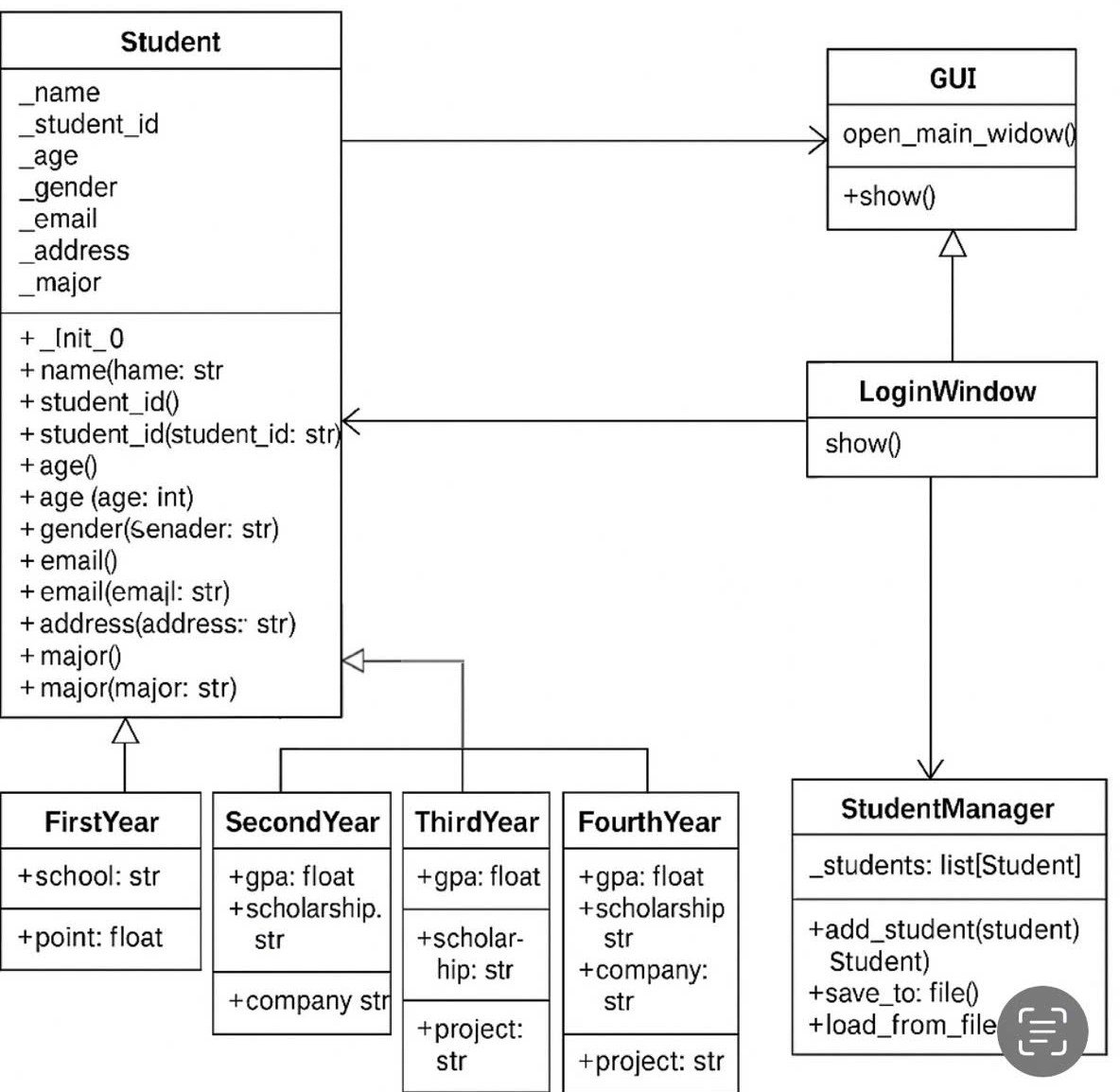
BÁO CÁO BÀI TẬP NHÓM

ĐỀ TÀI: QUẢN LÝ HỒ SƠ HỌC SINH

# Phân tích yêu cầu, ý nghĩa, ứng dụng

* 1. Yêu cầu bài toán:
* Mỗi học sinh cần lưu trữ: họ tên, tuổi, quê quán, lớp.
* Chức năng cơ bản:
* Thêm học sinh mới.
* Hiển thị học sinh.
  1. Ý nghĩa:
* Giảm thiểu thao tác thủ công trong quản lý hồ sơ.
* Tăng tính chính xác và hiệu quả trong tra cứu, thống kê.
* Tạo nền tảng cho việc nâng cấp hệ thống quản lý lớn hơn (kết nối cơ sở dữ liệu, xuất báo cáo...).
  1. Ứng dụng:
* Sử dụng cho văn phòng trường học, giáo viên chủ nhiệm, bộ phận quản lý học vụ.

# Thiết kế UML, UI/UX, sơ đồ nguyên lý



# Phát triển phần mềm ứng dụng

* 1. Mục tiêu phần mềm

Phần mềm "Quản lý hồ sơ học sinh" được xây dựng nhằm hỗ trợ các trường Trung học phổ thông trong việc lưu trữ, quản lý và truy xuất thông tin học sinh theo từng khối lớp (từ lớp 10 đến lớp 12). Mỗi học sinh được mô tả bằng một hồ sơ đầy đủ, bao gồm các thông tin cá nhân, kết quả học tập và các hoạt động học tập bổ sung (nếu có).Thông tin cá nhân (họ tên, tuổi, giới tính, địa chỉ,...)

* 1. Tính năng chính:

Phần mềm bao gồm các chức năng cơ bản như sau:

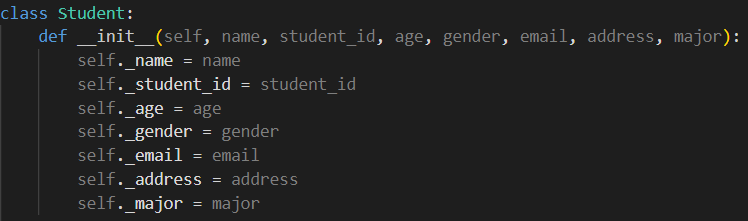
* **Thêm học sinh mới**: Nhập thông tin học sinh và phân loại theo từng khối lớp (lớp 10 đến 12).
* **Hiển thị danh sách học sinh**: Xem danh sách học sinh đã lưu kèm đầy đủ thông tin.
* **Xóa học sinh**: Cho phép xóa học sinh khỏi hệ thống theo mã số học sinh.
* **Lưu dữ liệu ra tệp JSON**: Tự động lưu danh sách học sinh khi có thay đổi.
* **Tải dữ liệu từ tệp JSON**: Tự động nạp danh sách học sinh khi khởi động lại chương trình.
  1. Kiến trúc chương trình

|  |  |
| --- | --- |
| **Lớp** | **Mô tả** |
| Student | Lớp cha, lưu thông tin cơ bản của học sinh |
| FirstYear | Sinh viên năm nhất, có thông tin về trường THCS và điểm đầu vào |
| SecondYear | Sinh viên năm 2, có thông tin GPA và học bổng |
| ThirdYear | Sinh viên năm 3, có thông tin công ty thực tập |
| FourthYear | Sinh viên chuẩn bị tốt nghiệp, có đề tài nghiên cứu |
| StudentManager | Lớp quản lý danh sách sinh viên và các thao tác thêm/xóa/lưu/truy xuất |

# Kỹ thuật Lập trình Hướng Đối Tượng

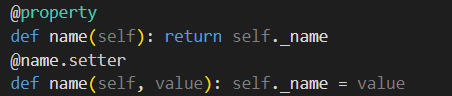
* 1. Tính Trừu Tượng (Abstraction)

Trong dự án này, nhóm đã áp dụng tính trừu tượng để xây dựng các lớp và đối tượng sinh viên. Cụ thể, lớp `Student` đại diện cho những thông tin chung của tất cả các sinh viên như: tên, mã sinh viên, tuổi, giới tính, email, địa chỉ và chuyên ngành. Các lớp con `FirstYear`, `SecondYear`, `ThirdYear`, `FourthYear` sẽ mở rộng thông tin của từng loại sinh viên theo từng năm học.



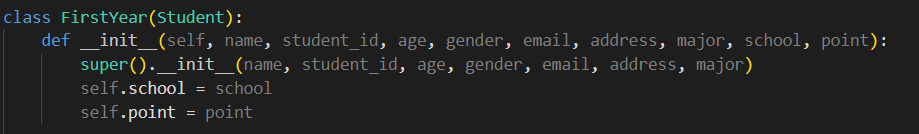
* 1. Tính Đóng Gói (Encapsulation) và Sử Dụng Getter/Setter Đúng Chuẩn

Nhóm đã áp dụng đóng gói bằng cách khai báo các thuộc tính của lớp Student và các lớp con ở mức private (dấu gạch dưới \_), truy cập thông qua các phương thức getter và setter chuẩn. Cách làm này giúp bảo vệ dữ liệu, kiểm soát việc thay đổi giá trị và giữ vững logic xử lý bên trong lớp.



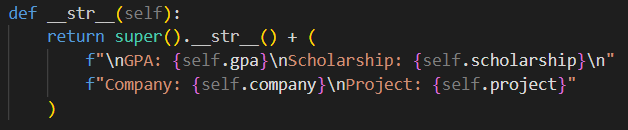
* 1. Tính Kế Thừa (Inheritance)

Lớp `Student` là lớp cha (base class) chứa các thuộc tính cơ bản của một sinh viên. Các lớp con như `FirstYear`, `SecondYear`, `ThirdYear`, `FourthYear` kế thừa từ lớp `Student` và mở rộng thêm các thuộc tính đặc thù của từng năm học, như điểm đầu vào, GPA, học bổng, công ty thực tập và đề tài nghiên cứu.



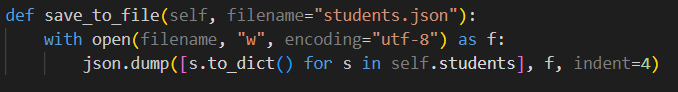
* 1. Tính Đa Hình (Polymorphism)

Nhóm đã áp dụng đa hình thông qua việc ghi đè (override) các phương thức \_\_str\_\_() và to\_dict() trong các lớp con. Việc này cho phép mỗi lớp con thay đổi cách hiển thị và xuất dữ liệu sao cho phù hợp với từng năm học, đảm bảo dữ liệu được định dạng đúng và hành vi phương thức phù hợp với từng loại đối tượng sinh viên.



* 1. Tuần Tự Hóa và Giải Tuần Tự (Serialization / Deserialization)

Nhóm đã sử dụng tuần tự hóa (serialization) và giải tuần tự hóa (deserialization) để lưu trữ và đọc dữ liệu sinh viên từ file JSON. Phương thức `save\_to\_file()` tuần tự hóa dữ liệu và ghi vào file, trong khi phương thức `load\_from\_file()` đọc dữ liệu từ file và giải tuần tự hóa lại thành đối tượng sinh viên.



* 1. Lưu Trữ: Lưu File

Dự án sử dụng file JSON để lưu trữ dữ liệu sinh viên. Phương thức `save\_to\_file()` ghi dữ liệu sinh viên dưới dạng JSON, trong khi phương thức `load\_from\_file()` đọc lại dữ liệu từ file để tái tạo các đối tượng sinh viên.

### Ưu điểm:

* Lưu trữ đơn giản, dễ dàng kiểm tra và chỉnh sửa dữ liệu bằng tay.
* Không cần hệ thống cơ sở dữ liệu phức tạp.

### Nhược điểm:

* Không thích hợp cho dữ liệu lớn hoặc yêu cầu truy vấn phức tạp.
* Thiếu các tính năng như bảo mật, ràng buộc khóa, hoặc backup tự động như trong cơ sở dữ liệu.
  1. Thuật Toán và Thư Viện Sử Dụng
* Thư viện sử dụng: `json` (thư viện chuẩn của Python) để tuần tự hóa và giải tuần tự hóa dữ liệu.
* Thuật toán: Không sử dụng thuật toán phức tạp, chủ yếu thao tác với danh sách, tìm kiếm theo ID sinh viên và ghi/đọc file JSON.
  1. Nền Tảng Phát Triển Phần Mềm
* Ngôn ngữ phát triển: Python
* IDE: VS Code.
* Lưu trữ: File JSON
* Tương thích hệ điều hành: Chạy tốt trên mọi hệ điều hành (Windows, macOS, Linux).

# Kiểm thử, ưu nhược điểm và phát triển tương lai

* 1. **Khả năng hoạt động của chương trình.**

**Chương trình có khả năng hoạt động khá tốt với các chức năng chính như:**

* Quản lý thông tin sinh viên: Cho phép thêm, xóa, và lưu trữ thông tin sinh viên theo các năm học khác nhau.
* Lưu trữ dữ liệu: Dữ liệu của sinh viên được lưu trữ trong tệp JSON, dễ dàng đọc và ghi lại.
* Các lớp kế thừa: Lớp Student được kế thừa để tạo ra các lớp con như FirstYear, SecondYear, ThirdYear, và FourthYear giúp tái sử dụng mã và dễ dàng mở rộng.
  1. **Ưu điểm.**
* Dễ hiểu và đơn giản: Code dễ đọc và dễ duy trì, sử dụng OOP hiệu quả.
* Tính kế thừa và đóng gói tốt: Cấu trúc lớp và các phương thức getter/setter bảo vệ dữ liệu.
* Lưu trữ dữ liệu đơn giản: Dữ liệu sinh viên được lưu trong file JSON.
  1. **Nhược điểm.**
* Hiệu suất kém với số lượng lớn: Việc lưu trữ dữ liệu trong file JSON sẽ gặp vấn đề khi số lượng sinh viên lớn, gây chậm trễ trong quá trình đọc và ghi dữ liệu.
* Thiếu giao diện người dùng (UI): Phần mềm chỉ có thể hoạt động qua dòng lệnh, thiếu giao diện người dùng giúp nâng cao trải nghiệm người dùng.
* Không hỗ trợ đa người dùng: Hệ thống chỉ hỗ trợ một người dùng duy nhất, không có khả năng sử dụng đồng thời cho nhiều người.
* Chưa có kiểm thử đầy đủ: Chưa có các kiểm thử tích hợp và kiểm thử hệ thống, cần bổ sung kiểm tra các tính năng và xử lý lỗi tiềm ẩn.
  1. **Kế hoạch phát triển trong tương lai.**
* Cải tiến lưu trữ dữ liệu: Chuyển sang cơ sở dữ liệu thay vì sử dụng file JSON để nâng cao hiệu suất, đảm bảo khả năng mở rộng khi số lượng sinh viên tăng lên.
* Xây dựng giao diện người dùng (UI): Thiết kế giao diện đồ họa giúp người dùng dễ dàng tương tác với phần mềm, thay vì chỉ sử dụng dòng lệnh.
* Hỗ trợ đa người dùng: Tích hợp tính năng đa người dùng, cho phép nhiều người truy cập và sử dụng hệ thống đồng thời.
* Kiểm thử hệ thống: Thực hiện các kiểm thử tích hợp và kiểm thử hệ thống để đảm bảo các tính năng hoạt động ổn định và không có lỗi trong quá trình sử dụng.

1. **Mô tả mã nguồn mở**
   1. **Giao Diện**

* Cửa sổ đăng nhập được chia thành hai phần:
* **Vùng trái**: Hiển thị logo ứng dụng, căn giữa trong cửa sổ.
* **Vùng phải**: Chứa các trường nhập liệu cho tên người dùng và mật khẩu, cùng với các nút bấm.
* Cửa sổ có kích thước 900x500 và không thể thay đổi kích thước.
  1. **Chức năng**
* **Tên người dùng và Mật khẩu**: Người dùng nhập thông tin vào các trường tương ứng.
* **Nút "Login"**: Khi nhấn, hệ thống kiểm tra thông tin (tên người dùng = "admin", mật khẩu = "123"). Nếu đúng, cửa sổ đăng nhập sẽ đóng và mở giao diện chính. Nếu sai, thông báo lỗi sẽ xuất hiện.
* **Nút "Quên mật khẩu?"**: Hiển thị thông báo yêu cầu người dùng liên hệ với quản trị viên.
  1. **Thư Viện Sử Dụng**
* **Tkinter**: Thư viện chính để xây dựng giao diện người dùng (GUI).
* **PIL (Python Imaging Library)**: Được sử dụng để tải và hiển thị logo ứng dụng.

# Phân công công việc nhóm

|  |  |
| --- | --- |
| Họ và tên | Nhiệm vụ |
| Nguyễn Hữu Huy Hoàng | Phân chia công việc + Lập trình chính |
| Hoàng Viết Hiếu | Lập trình viên phụ + Tài liệu hướng dẫn và video |
| Vũ Ngọc Minh | Viết báo cáo + Kiểm thử |
| Bùi Mạnh Trung | Chuẩn bị slide + Trình bày kết quả dự án |

1. **Tiến độ thực hiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Người phụ trách** | **Hạn chót** |
| 1 | Chọn đề tài & thành lập nhóm | Tất cả thành viên | 14/04/2025 |
| 2 | Thiết kế lớp cơ sở & các lớp con | Nguyễn Hữu Huy Hoàng, Hoàng Viết Hiếu | 16/04/2025 |
| 3 | Cài đặt getter/setter và phương thức | Nguyễn Hữu Huy Hoàng, Hoàng Viết Hiếu | 17/04/2025 |
| 4 | Xây dựng lưu/trả dữ liệu JSON | Nguyễn Hữu Huy Hoàng | 18/04/2025 |
| 5 | Viết unit test cho các lớp & phương thức | Hoàng Viết Hiếu | 19/04/2025 |
| 6 | Kiểm thử tích hợp | Vũ Ngọc Minh | 20/04/2025 |
| 7 | Chuẩn bị slide & kịch bản thuyết trình | Bùi Mạnh Trung | 21/04/2025 |
| 8 | Đánh giá, rà soát & bàn giao nội bộ | Toàn bộ thành viên | 23/04/2025 |