1. **Tính cấp thiết của đề tài ( Thắng )**

Trong bối cảnh giáo dục đại học hiện đại, nhu cầu quản lý thời khóa biểu của sinh viên và giảng viên một cách hiệu quả, chính xác đã trở thành một yêu cầu thiết yếu. Với số lượng học phần ngày càng tăng cùng với lịch giảng dạy phức tạp, các trường đại học gặp nhiều thách thức trong việc xây dựng và quản lý thời khóa biểu. Việc này đòi hỏi không chỉ khả năng sắp xếp hợp lý mà còn phải đảm bảo tính linh hoạt và thuận tiện trong việc truy cập và điều chỉnh. Nếu không có hệ thống quản lý tự động, nguy cơ trùng lặp lịch học, sai sót trong phân bổ lớp học hoặc thậm chí là bỏ sót học phần là rất lớn, từ đó ảnh hưởng trực tiếp đến trải nghiệm học tập của sinh viên và giảng viên.

Hệ thống quản lý thời khóa biểu tự động hóa ra đời nhằm khắc phục các khó khăn trên. Khác với phương pháp quản lý thủ công truyền thống, hệ thống này cho phép trường học nhập liệu, điều chỉnh và xuất thời khóa biểu một cách nhanh chóng. Điều này giúp giảm thiểu đáng kể thời gian xử lý, đồng thời giảm thiểu khả năng xảy ra sai sót trong quá trình sắp xếp thời khóa biểu cho các lớp học và các phòng học. Đối với sinh viên và giảng viên, việc có một hệ thống truy cập trực tuyến giúp họ dễ dàng tra cứu lịch trình cá nhân, thuận tiện trong việc lên kế hoạch học tập, giảng dạy và sắp xếp thời gian một cách hiệu quả. Điều này đặc biệt quan trọng khi các học phần có thể thay đổi và cần cập nhật kịp thời.

Ngoài ra, việc xây dựng hệ thống này theo phương pháp lập trình hướng đối tượng (OOP) còn mang lại nhiều lợi ích kỹ thuật. Bằng cách tổ chức dữ liệu theo cấu trúc các lớp và đối tượng, hệ thống đạt được tính linh hoạt, dễ mở rộng và dễ bảo trì. Các lớp đối tượng trong hệ thống có thể đại diện cho các thực thể như sinh viên, giảng viên, học phần, và thời khóa biểu. Điều này giúp việc cập nhật hoặc thay đổi thông tin trở nên đơn giản hơn. Khi có sự thay đổi về cấu trúc hoặc yêu cầu của hệ thống, chỉ cần thay đổi một số lớp nhất định thay vì phải sửa đổi toàn bộ mã nguồn. Chính nhờ ưu điểm này, phương pháp lập trình hướng đối tượng đảm bảo hệ thống có thể phát triển lâu dài, thích ứng với các yêu cầu mới trong tương lai.

Hơn nữa, đề tài này còn có giá trị học thuật đáng kể. Đối với sinh viên, việc tham gia vào quá trình thiết kế và xây dựng một hệ thống theo phương pháp lập trình hướng đối tượng là cơ hội để áp dụng lý thuyết vào thực tiễn, giúp họ hiểu sâu hơn về các nguyên lý OOP như kế thừa, đa hình, đóng gói và trừu tượng hóa. Kết quả đạt được từ dự án không chỉ dừng lại ở việc hoàn thiện hệ thống quản lý thời khóa biểu, mà còn là nền tảng để sinh viên có thể tiếp tục phát triển các kỹ năng lập trình và thiết kế hệ thống sau này.

1. **Mục đích của đề tài ( Thắng )**

Đề tài "Xây dựng hệ thống quản lý thời khóa biểu theo phương pháp lập trình hướng đối tượng" nhằm giải quyết các vấn đề trong việc tổ chức và quản lý lịch học một cách hiệu quả, khoa học và có tính linh hoạt cao. Dưới đây là các mục tiêu chính mà đề tài hướng đến:

1. Tối ưu hóa quản lý thời khóa biểu  
   Hệ thống này giúp giảm tải công việc quản lý thời khóa biểu cho các phòng ban trong trường đại học. Với khối lượng lớn các học phần và lịch học đa dạng, hệ thống quản lý tự động hóa sẽ hỗ trợ sắp xếp, quản lý và kiểm tra thông tin thời khóa biểu nhanh chóng. Nhờ đó, nhân viên phụ trách có thể nhập liệu, thay đổi và truy xuất thông tin một cách chính xác và hiệu quả hơn so với việc quản lý thủ công.
2. Tăng cường khả năng tra cứu và truy cập thông tin  
   Một mục đích quan trọng của hệ thống là cung cấp khả năng truy cập thông tin thời khóa biểu dễ dàng cho sinh viên và giảng viên. Thay vì phải đến văn phòng hoặc tra cứu qua các bảng thông báo truyền thống, sinh viên và giảng viên có thể xem lịch học và lịch dạy của mình một cách thuận tiện, mọi lúc mọi nơi. Điều này giúp họ dễ dàng lên kế hoạch cá nhân, đảm bảo không bỏ lỡ các tiết học hoặc các buổi giảng dạy.
3. Giảm thiểu sai sót và xung đột trong lịch học  
   Một trong những vấn đề phổ biến trong quản lý thời khóa biểu là tình trạng xung đột thời gian giữa các môn học hoặc giảng viên. Hệ thống được thiết kế để kiểm tra và phát hiện các xung đột thời gian, từ đó ngăn chặn việc sinh viên bị trùng lịch học hoặc giảng viên phải đảm nhận nhiều lớp trong cùng một khung giờ. Nhờ vậy, hệ thống không chỉ đảm bảo tính chính xác mà còn cải thiện trải nghiệm học tập và giảng dạy.
4. Ứng dụng phương pháp lập trình hướng đối tượng (OOP) để nâng cao khả năng mở rộng và bảo trì  
   Sử dụng phương pháp lập trình hướng đối tượng giúp hệ thống có cấu trúc rõ ràng và dễ mở rộng. Các lớp đối tượng như SinhVien, GiangVien, HocPhan, ThoiKhoaBieu được tổ chức một cách khoa học, dễ bảo trì và phát triển. Khi có yêu cầu thay đổi, chẳng hạn như bổ sung thêm chức năng, nâng cấp hệ thống hay điều chỉnh cấu trúc dữ liệu, các thay đổi chỉ cần thực hiện ở một số lớp liên quan, giúp tiết kiệm thời gian và công sức bảo trì.
5. Giúp sinh viên và giảng viên làm quen với hệ thống số hóa trong môi trường giáo dục  
   Việc triển khai một hệ thống quản lý thời khóa biểu điện tử không chỉ hỗ trợ nhà trường mà còn giúp sinh viên và giảng viên làm quen với việc sử dụng các công cụ công nghệ trong học tập và giảng dạy. Điều này phù hợp với xu hướng chuyển đổi số trong giáo dục, giúp các bên liên quan thích nghi với những thay đổi công nghệ và tận dụng được lợi ích của hệ thống số hóa.
6. Phát triển kỹ năng lập trình và tư duy hệ thống cho sinh viên  
   Đối với sinh viên thực hiện dự án, đề tài là cơ hội tốt để áp dụng kiến thức lý thuyết về lập trình hướng đối tượng vào thực tiễn. Thông qua quá trình xây dựng hệ thống, sinh viên sẽ hiểu rõ hơn về các nguyên lý như kế thừa, đa hình, đóng gói và trừu tượng hóa. Họ sẽ được rèn luyện tư duy hệ thống và có cơ hội học hỏi các kỹ năng thiết kế, phân tích và lập trình, giúp phát triển năng lực cá nhân cho những dự án thực tế sau này.
7. **Khảo sát đề tài (Vũ)**

Hiểu biết về OOP: Khoảng 90% người tham gia đều có kiến thức cơ bản về các nguyên lý OOP như kế thừa, đóng gói, đa hình và trừu tượng. Tuy nhiên, một số ít người tham gia chưa thực sự hiểu rõ cách áp dụng các nguyên lý này trong bối cảnh hệ thống thời khóa biểu.

Kinh nghiệm phát triển phần mềm OOP: 60% người tham gia đã từng phát triển phần mềm sử dụng OOP, nhưng chỉ có 40% có kinh nghiệm trực tiếp với các hệ thống quản lý thời khóa biểu.

Thiết kế các lớp đối tượng: Một số lớp đối tượng được đề xuất cho hệ thống thời khóa biểu bao gồm:

* Lớp Môn Học: Chứa thông tin về môn học, thời gian, giảng viên và phòng học.
* Lớp Giảng Viên: Quản lý thông tin về giảng viên, lịch dạy và các môn học phụ trách.
* Lớp Sinh Viên: Quản lý lịch học của sinh viên, môn học đã đăng ký và điểm số.
* Lớp Thời Gian: Quản lý thông tin về các khung giờ học và lịch học chung.

Ứng dụng OOP vào quản lý thời khóa biểu: Phần lớn người tham gia nhận thấy rằng OOP có thể giúp dễ dàng mở rộng hệ thống, đặc biệt khi cần thêm các tính năng mới như tích hợp với lịch cá nhân, thông báo lịch học thay đổi, hoặc phân tích dữ liệu thời khóa biểu.

1. **Mô tả yêu cầu đề tài (Vũ)**

Hệ thống quản lý thời khóa biểu đóng vai trò quan trọng trong việc tổ chức các hoạt động giảng dạy và học tập của sinh viên, giảng viên, và các phòng học trong các cơ sở giáo dục. Việc xây dựng hệ thống thời khóa biểu hiệu quả, dễ dàng mở rộng và bảo trì là một thách thức lớn. Áp dụng phương pháp Lập trình Hướng Đối Tượng (OOP) vào thiết kế hệ thống có thể giúp giải quyết các vấn đề này bằng cách tạo ra các đối tượng (objects) mô phỏng các thực thể trong hệ thống như giảng viên, sinh viên, môn học, phòng học, thời gian, từ đó giúp tối ưu hóa các thao tác quản lý.

**Chương 3: Giai đoạn xây dựng : Phân tích**

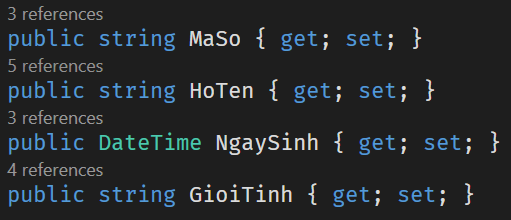
1. **Class Người ( Thắng, Vũ )**

Lớp Nguoi đóng vai trò là một lớp cha trừu tượng, cung cấp các thuộc tính và phương thức cơ bản cho các đối tượng con như SinhVien và GiangVien. Lớp này thể hiện các thông tin chung mà bất kỳ cá nhân nào (dù là sinh viên hay giảng viên) cũng sẽ có, chẳng hạn như tên, tuổi, địa chỉ, và số điện thoại. Nhờ áp dụng tính kế thừa trong lập trình hướng đối tượng, lớp Nguoi giúp giảm thiểu sự lặp lại mã nguồn khi các thuộc tính và phương thức chung được định nghĩa tại đây và có thể sử dụng lại ở các lớp con.

Cấu trúc và Thuộc tính chính của Lớp Nguoi

Lớp Nguoi có các thuộc tính cơ bản sau:

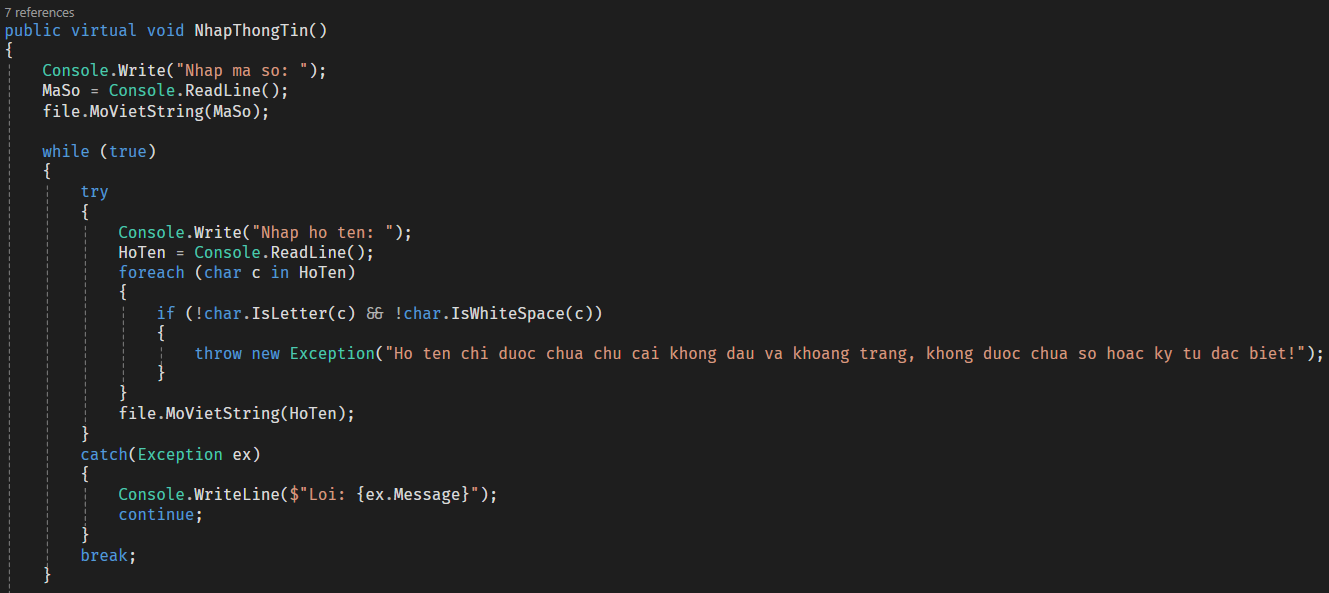
* HoTen (string): Lưu trữ tên của người dùng, là thông tin chung cho cả sinh viên và giảng viên.
* MaSo (int): Biểu thị độ tuổi của người dùng.
* NgaySinh (DateTime): Địa chỉ nơi ở của người dùng.
* GioiTinh (string): Số điện thoại liên hệ của người dùng.

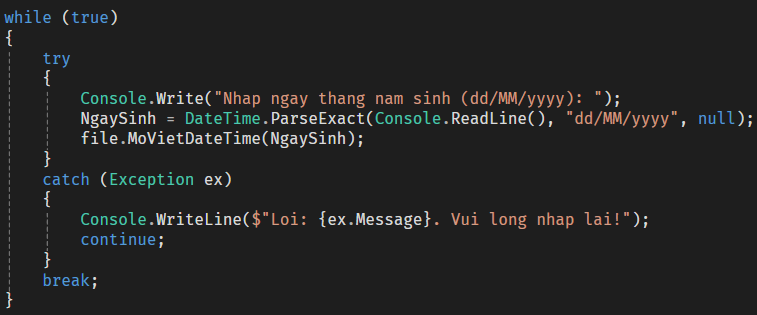


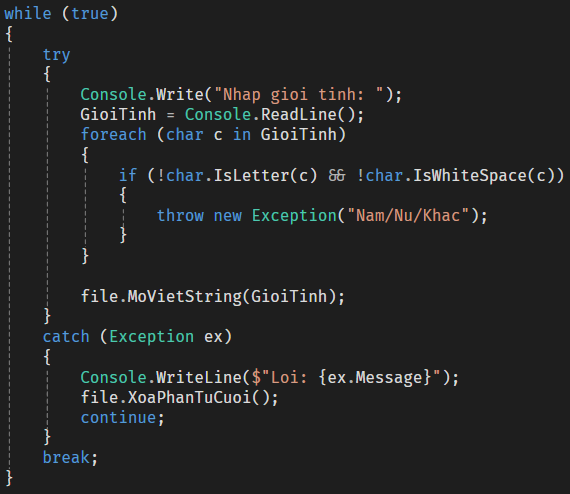
Các Phương thức chính của Lớp Nguoi

Lớp Nguoi chứa một số phương thức quan trọng để xử lý thông tin cá nhân của người dùng. Dưới đây là các phương thức cơ bản:

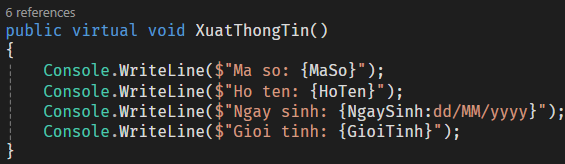
1. **NhapThongTin()**
   * Phương thức NhapThongTin() cho phép người dùng nhập thông tin cá nhân cơ bản như tên, mã số, ngày sinh, và giới tính.
   * Phương thức này thường được gọi khi người dùng khởi tạo một đối tượng sinh viên hoặc giảng viên và muốn lưu trữ thông tin cá nhân của họ vào hệ thống.
   * Cơ chế hoạt động: Người dùng sẽ được yêu cầu nhập từng thông tin, và dữ liệu sẽ được lưu trữ vào các thuộc tính tương ứng.







1. **XuatThongTin()**
   * Phương thức XuatThongTin() được sử dụng để hiển thị các thông tin cá nhân đã được nhập cho người dùng, giúp kiểm tra và xác nhận các thông tin đã lưu.
   * Cơ chế hoạt động: Phương thức này in ra màn hình các thông tin cơ bản của người dùng, bao gồm tên, mã số, ngày sinh, và giới tính. Đây là phương thức giúp người dùng dễ dàng xem lại thông tin cá nhân đã lưu trữ.

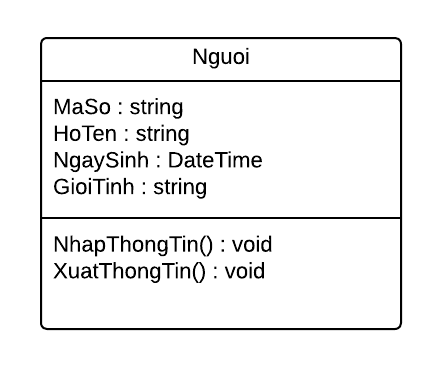


**Phân tích OOP trong Lớp Nguoi**

* Tính Kế thừa: Nguoi là lớp cha của các lớp SinhVien và GiangVien. Tất cả các thuộc tính và phương thức chung của Nguoi đều có thể được tái sử dụng trong các lớp con mà không cần định nghĩa lại.
* Tính Đóng gói: Các thuộc tính của Nguoi được khai báo với quyền truy cập thích hợp để đảm bảo chỉ các phương thức của lớp (hoặc lớp con) mới có thể truy cập và thay đổi dữ liệu.
* Tính Trừu tượng: Nguoi có thể được thiết kế dưới dạng lớp trừu tượng (abstract) nếu có các phương thức cần được triển khai cụ thể ở các lớp con, chẳng hạn như XuatThongTin() được tùy biến riêng cho SinhVien hoặc GiangVien.

**Ảnh minh họa (Sơ đồ và Mã nguồn)**

**(DIAGRAM)**



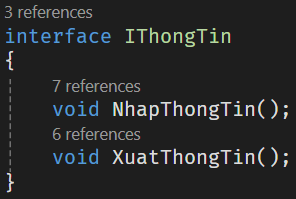
1. **Class Interface liên quan đến Người (Thắng, Vũ)**

IThongTin cung cấp nền tảng để phát triển các lớp quản lý thông tin của giảng viên và sinh viên một cách linh hoạt và có tổ chức.

Giao diện: IThongTin được sử dụng để khai báo các phương thức mà bất kỳ lớp nào triển khai (implement) giao diện này đều phải định nghĩa.

Phương thức khai báo:

* NhapThongTin(): Một phương thức không trả về giá trị, dự kiến để thu thập thông tin từ người dùng.
* XuatThongTin(): Một phương thức không trả về giá trị, dự kiến để hiển thị thông tin đã được nhập



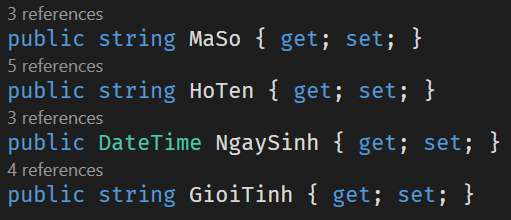
1. **Class Sinh Viên ( Thắng )**

Lớp SinhVien là một lớp con của Nguoi, sử dụng các thuộc tính cơ bản từ lớp cha và bổ sung các thuộc tính, phương thức đặc thù cho sinh viên. Điều này cho phép lớp SinhVien quản lý thông tin sinh viên chi tiết hơn, bao gồm mã lớp và khóa học, phục vụ mục tiêu xây dựng hệ thống quản lý thời khóa biểu.

Cấu trúc và Thuộc tính của Lớp SinhVien

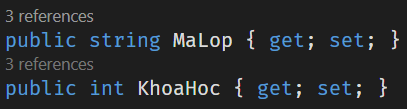
Lớp SinhVien kế thừa các thuộc tính từ lớp Nguoi, bao gồm:

* MaSo (string): Mã số định danh duy nhất của mỗi cá nhân.
* HoTen (string): Tên đầy đủ của cá nhân.
* NgaySinh (DateTime): Ngày sinh của cá nhân.
* GioiTinh (string): Giới tính của cá nhân.



Ngoài các thuộc tính kế thừa, lớp SinhVien có thêm các thuộc tính riêng như sau:

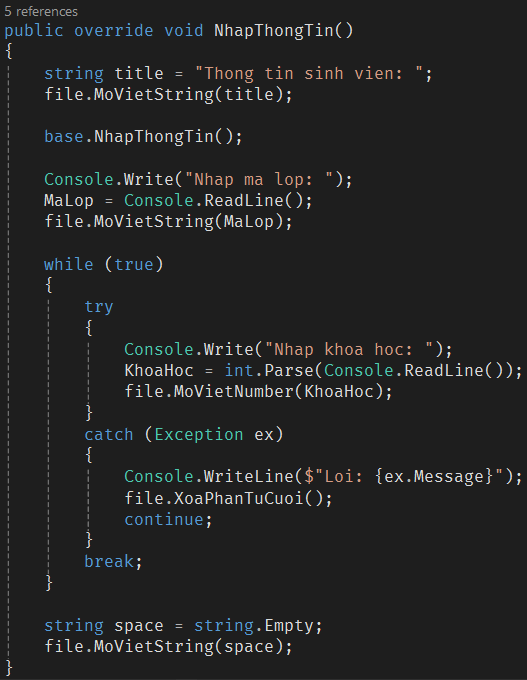
* MaLop (string): Mã lớp của sinh viên, giúp phân loại sinh viên theo lớp học.
* KhoaHoc (int): Biểu thị khóa học của sinh viên (ví dụ: khóa học 2023).



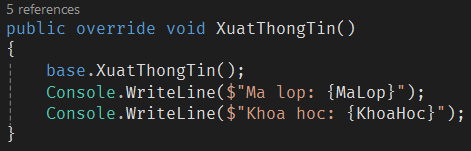
Các Phương thức của Lớp SinhVien

Lớp SinhVien có một số phương thức chính, trong đó các phương thức NhapThongTin() và XuatThongTin() được ghi đè từ lớp cha Nguoi để đáp ứng nhu cầu quản lý thông tin riêng biệt cho sinh viên.

1. **NhapThongTin()**
   * Phương thức NhapThongTin() ghi đè (override) từ Nguoi, cho phép nhập thông tin sinh viên bao gồm các thuộc tính kế thừa và các thuộc tính riêng như mã lớp và khóa học.
   * Cơ chế hoạt động: Sau khi gọi phương thức NhapThongTin() của lớp cha để nhập thông tin cơ bản, hệ thống sẽ yêu cầu nhập các thông tin bổ sung như mã lớp và khóa học.



1. **XuatThongTin()**
   * Phương thức XuatThongTin() ghi đè từ Nguoi, dùng để hiển thị thông tin sinh viên bao gồm cả các thuộc tính cơ bản và các thông tin riêng như mã lớp và khóa học.
   * Cơ chế hoạt động: Phương thức này hiển thị thông tin kế thừa từ Nguoi trước, sau đó bổ sung các thông tin đặc thù của SinhVien.

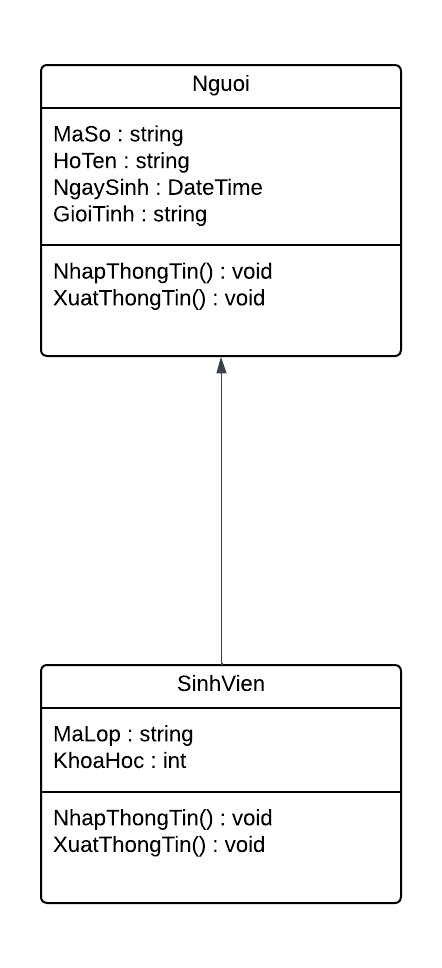


**Phân tích OOP trong Lớp SinhVien**

* Tính Kế thừa: SinhVien kế thừa các thuộc tính và phương thức từ lớp cha Nguoi, tránh lặp lại mã nguồn và giúp dễ bảo trì.
* Tính Đa hình: SinhVien ghi đè các phương thức từ Nguoi, cho phép điều chỉnh phương thức nhập và xuất thông tin để phù hợp với sinh viên.
* Tính Đóng gói: Các thuộc tính riêng như MaLop và KhoaHoc được đóng gói bên trong SinhVien, đảm bảo bảo mật dữ liệu và tránh thay đổi không mong muốn.

**Mã nguồn mẫu của Lớp SinhVien**

**(DIAGRAM)**



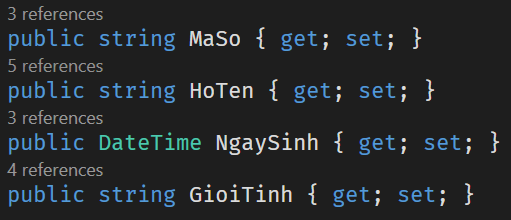
1. **Class GiangVien**

Lớp GiangVien là một lớp con của lớp Nguoi, sử dụng các thuộc tính cơ bản từ lớp cha và bổ sung các thuộc tính, phương thức đặc thù cho giảng viên. Điều này cho phép lớp GiangVien quản lý thông tin giảng viên chi tiết hơn, bao gồm bộ môn, phục vụ mục tiêu xây dựng hệ thống quản lý thời khóa biểu.

Cấu trúc và Thuộc tính của Lớp GiangVien:

Lớp GiangVien kế thừa các thuộc tính từ lớp Nguoi, bao gồm:

* MaSo (string): Mã số định danh duy nhất của mỗi cá nhân.
* HoTen (string): Tên đầy đủ của cá nhân.
* NgaySinh (DateTime): Ngày sinh của cá nhân.
* GioiTinh (string): Giới tính của cá nhân.



Ngoài các thuộc tính kế thừa, lớp GiangVien có thêm các thuộc tính riêng như sau:

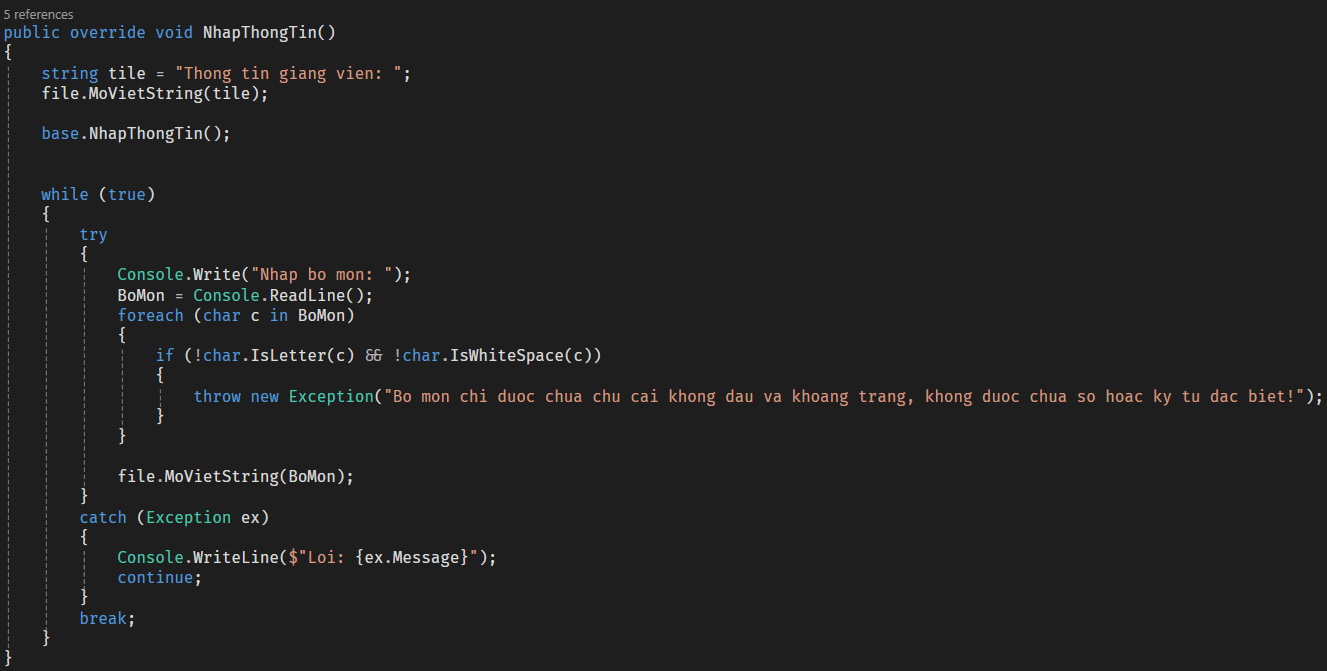
* BoMon (string): Bộ môn của giảng viên, giúp phân loại sinh viên theo môn học.



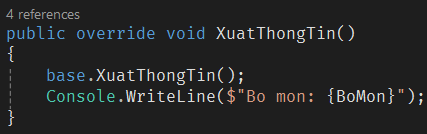
Các Phương thức của Lớp GiangVien

Lớp GiangVien có một số phương thức chính, trong đó các phương thức NhapThongTin() và XuatThongTin() được ghi đè từ lớp cha Nguoi để đáp ứng nhu cầu quản lý thông tin riêng biệt cho giảng viên.

1. **NhapThongTin()**
   * Phương thức NhapThongTin() ghi đè (override) từ Nguoi, cho phép nhập thông tin giảng viên bao gồm các thuộc tính kế thừa và các thuộc tính riêng như bộ môn.
   * Cơ chế hoạt động: Sau khi gọi phương thức NhapThongTin() của lớp cha để nhập thông tin cơ bản, hệ thống sẽ yêu cầu nhập các thông tin bổ sung như bộ môn.



1. **XuatThongTin()**
   * Phương thức XuatThongTin() ghi đè từ Nguoi, dùng để hiển thị thông tin giảng viên bao gồm cả các thuộc tính cơ bản và các thông tin riêng như bộ môn.
   * Cơ chế hoạt động: Phương thức này hiển thị thông tin kế thừa từ Nguoi trước, sau đó bổ sung các thông tin đặc thù của GiangVien.



**Phân tích OOP trong Lớp GiangVien**

* Tính Kế thừa: GiangVien kế thừa các thuộc tính và phương thức từ lớp cha Nguoi, tránh lặp lại mã nguồn và giúp dễ bảo trì.
* Tính Đa hình: GiangVien ghi đè các phương thức từ Nguoi, cho phép điều chỉnh phương thức nhập và xuất thông tin để phù hợp với giảng viên.
* Tính Đóng gói: Các thuộc tính riêng như BoMon được đóng gói bên trong GiangVien, đảm bảo bảo mật dữ liệu và tránh thay đổi không mong muốn.

**Mã nguồn mẫu của Lớp GiangVien**

**(DIAGRAM)**

