**HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION FACULTY FOR HIGH QUALITY TRAINING INFORMATION TECHNOLOGY**



**BÁO CÁO CUỐI KỲ**

**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Giảng viên hướng dẫn : Thầy Hoàng Công Trình**

**Nhóm trưởng: Nguyễn Nhật Nam - 23162058**

**Thành viên  : Trương Anh Khoa - 23162044**

**Thành viên  : Vũ Đức Thắng – 23162095**

**Thành viên  : Hoàng Văn Vương Thu - 23162099**

**Thành viên  : Vũ Hoàng Vũ - 23162116**

**Lớp** **: Chiều thứ 7**

***Tp. Hồ Chí Minh, Tháng 11,  2024***

**MỤC LỤC**

[**CHƯƠNG 1: GIAI ĐOẠN KHỞI ĐẦU 3**](#_Toc182842286)

[1) Tính cấp thiết của đề tài 3](#_Toc182842287)

[2) Mục đích của đề tài 4](#_Toc182842288)

[3) Khảo sát đề tài 5](#_Toc182842289)

[4) Mô tả yêu cầu đề tài 6](#_Toc182842290)

[**CHƯƠNG 2: GIAI ĐOẠN TIỀN XÂY DỰNG 7**](#_Toc182842291)

[**CHƯƠNG 3: GIAI ĐOẠN XÂY DỰNG : PHÂN TÍCH 7**](#_Toc182842292)

[1) Class Nguoi 7](#_Toc182842293)

[2) Class Interface IThongTin 10](#_Toc182842294)

[3) Class SinhVien 11](#_Toc182842295)

[4) Class GiangVien 13](#_Toc182842296)

[5) Class HocPhan 16](#_Toc182842297)

[6) Class ThoiKhoaBieu 24](#_Toc182842298)

[7) Class QuanLyTKB 30](#_Toc182842299)

[8) Class TapTin 38](#_Toc182842300)

[9) Class KetThuc 41](#_Toc182842301)

[10) Program 42](#_Toc182842302)

[**CHƯƠNG 4: KẾT LUẬN VÀ ĐÁNH GIÁ 47**](#_Toc182842303)

[1) Những điểm mạnh 47](#_Toc182842304)

[2) Những hạn chế 47](#_Toc182842305)

[3) Bài học kinh nghiệm 47](#_Toc182842305)

[4) Cách giải quyết khó khăn 47](#_Toc182842305)

[5) Kết luận 47](#_Toc182842305)

# CHƯƠNG 1: GIAI ĐOẠN KHỞI ĐẦU

1. **Tính cấp thiết của đề tài**

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**

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**

**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:

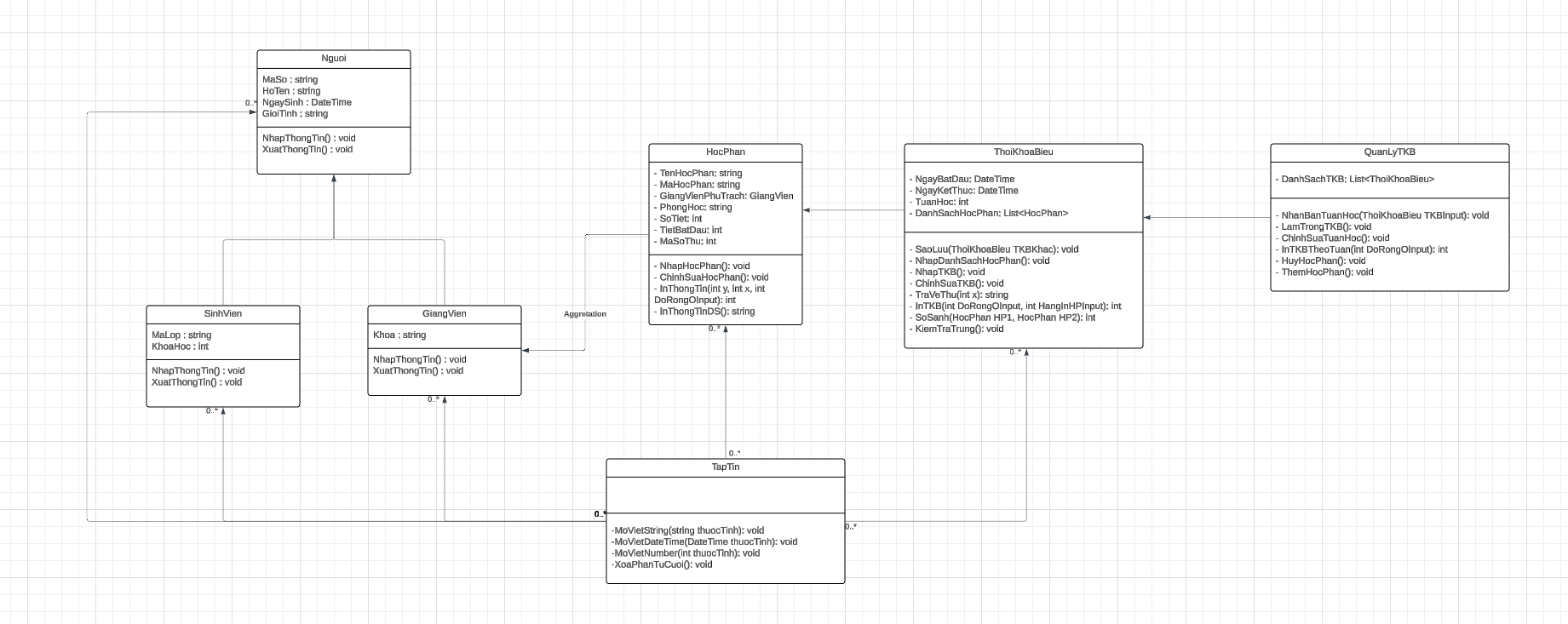
* **Lớp Môn Học:** Chứa thông tin về môn học, thời gian, giảng viên và phòng học.
* **Lớp Giảng Viên:** Quản lý thông tin về giảng viên, lịch dạy và các môn học phụ trách.
* **Lớp Sinh Viên:** Quản lý lịch học của sinh viên, môn học đã đăng ký và điểm số.
* **Lớp Thời Gian:** Quản lý thông tin về các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

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 2: GIAI ĐOẠN TIỀN XÂY DỰNG

****

# CHƯƠNG 3: GIAI ĐOẠN XÂY DỰNG : PHÂN TÍCH

1. **Class Nguoi**

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.

A screen shot of a computer code

Description automatically generated

**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:

**+ 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.

A computer screen shot of a black screen

Description automatically generated

A computer screen with text

Description automatically generated

A computer screen shot of a program code

Description automatically generated

**+ 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.

A screen shot of a computer code

Description automatically generated

**Phân tích OOP trong class 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.

A white rectangular box with black text

Description automatically generated**Ảnh minh họa (Sơ đồ và Mã nguồn)**

**(DIAGRAM)**

## A black screen with white text Description automatically generated Class Interface IThongTin

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 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 SinhVien**

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.

A screen shot of a computer code

Description automatically generated

**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).

A black background with white text

Description automatically generated

**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.

A screen shot of a computer program

Description automatically generated **+ 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.

**+ 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.

A screen shot of a computer code

Description automatically generated

**Phân tích OOP class 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.

**A diagram of a computer

Description automatically generatedMã 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.

A screen shot of a computer code

Description automatically generated

**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.

A black background with white text

Description automatically generated

**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.

**+ 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.

A computer screen shot of a black screen

Description automatically generated

**+ 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.

A screen shot of a computer code

Description automatically generated

**Phân tích OOP trong class 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.
* A diagram of a computer

  Description automatically generated**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)**

1. **Class HocPhan**

**Lớp HocPhan có các thuộc tính sau:**

* **TenHocPhan (string):** Lưu tên của học phần.
* **MaHocPhan (string):** Lưu mã học phần.
* **GiangVienPhuTrach (GiangVien):** Lưu thông tin giảng viên dạy học phần.
* **PhongHoc (string):** Lưu tên phòng học.
* **SoTiet (int):** Lưu trữ số tiết học một buổi.
* **TietBatDau (int):** Lưu trữ tiết bắt đầu của môn đó, kết hợp với SoTiet để tính toán thời lượng của 1 môn học trong 1 buổi.
* **MaSoThu (int):** 0 – thứ Hai, 1 – thứ Ba,… 6 – Chủ Nhật: dùng trong ThoiKhoaBieu để in học phần tương ứng với thứ ngày.

A screen shot of a computer program

Description automatically generated A screen shot of a computer code

Description automatically generated

**Lớp HocPhan có các phương thức sau:**

* **Phương thức khởi tạo không tham số**

A computer screen with white text

Description automatically generated

Khởi tạo gán các giá trị mặc định cho các thuộc tính: tenHocPhan, maHocPhan, phongHoc là chuỗi rỗng. soTiet, tietBatDau, maSoThu được gán giá trị 0. Đảm bảo mọi đối tượng được tạo ra luôn có giá trị hợp lệ ban đầu, tránh lỗi.

* Phương thức khởi tạo đối tượng HPKhac

A computer screen shot of white text

Description automatically generated

Tạo một bản sao để sao chép toàn bộ tất cả thuộc tính từ đối tượng khác là HocPhan đã tồn tại. Đối tượng mới được tạo ra sẽ giống hệt một đối tượng đã tồn tại, giúp tiết kiệm thời gian và mã nguồn. Có thể thay đổi hoặc xử lý đối tượng mới này mà không gây ảnh hưởng gì đến đối tượng gốc.

* **Phương thức khởi tạo có tham số**

A black screen with blue and white text

Description automatically generated

Các tham số sẽ được gán tương ứng trực tiếp cho các thuộc tính của đối tượng. Gán sẵn các thông tin một cách đầy đủ để thử nghiệm chương trình.

* **Phương thức NhapHocPhan()**
* **Cho người dùng nhập vào những thông tin như:** tên học phần, mã học phần, giảng viên phụ trách, phòng học, tiết bắt đầu, số tiết, mã số thứ.
* Dùng try…catch để xử lý lỗi, báo cho người dùng nếu có lỗi nhập sai và cho nhập lại. Thông tin hợp lệ sẽ được lưu lại vào file.

A computer screen shot of text

Description automatically generated

A computer screen shot of code

Description automatically generated

A computer screen with white text

Description automatically generated

A computer screen shot of code

Description automatically generated

* **Phương thức ChinhSuaHocPhan()**
* In ra những thông tin vừa được nhập vào của học phần để người dùng kiểm tra

A computer screen shot of a program code

Description automatically generated

* Nếu thấy thông tin bị nhập sai, cho người dùng chọn nhập lại để chỉnh sửa thông tin

A screen shot of a computer program

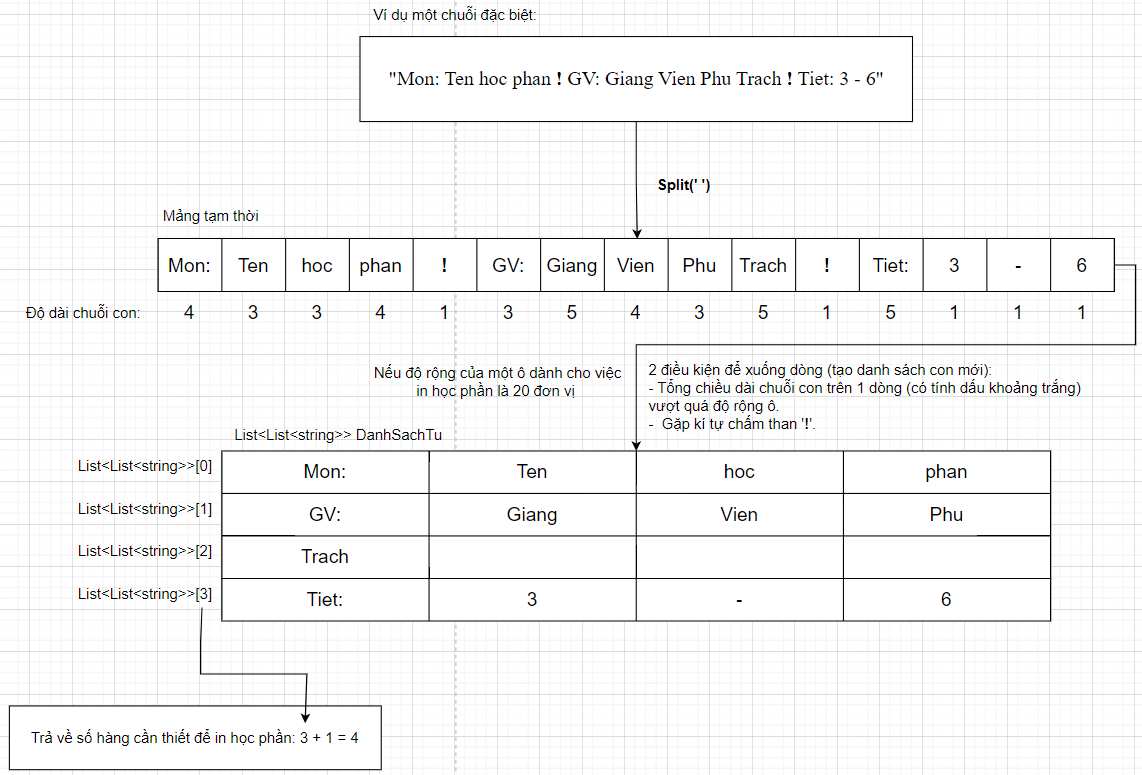
Description automatically generated

A computer screen shot of a program code

Description automatically generated

* Thông báo đã cập nhật thông tin mới nếu nhập vào hợp lệ và lưu lại thông tin mới này.
* **Phương thức InThongTin(x, y, độ rộng ô)**

**Phương thức in môn học cho khung thời khóa biểu:** InThongTin(). Đây là phương thức quan trọng giúp hoàn thành việc in thời khóa biểu ở dạng trực quan hơn trên màn hình console. Phương thức có tác dụng in thông tin của học phần vào các ô trong thời khóa biểu. Phương thức có 3 tham số: y (vị trí hàng console bắt đầu in), x (vị trí cột console bắt đầu in), độ rộng của ô hay cột thời khóa biểu. Phương thức trả về số hàng console cần thiết để in đủ thông tin của học phần này, giá trị trả về này là cần thiết cho một số tác vụ – đặt lại vị trí con trỏ chuột của console – bên trong các phương thức in thời khóa biểu.



Ý tưởng của phương thức này là chuyển một chuỗi đặc biệt chứa các thông tin cần của học phần thành một danh sách 2 chiều List<List<string>>. Danh sách cha chứa các phần tử bên trong là các danh sách con, tương ứng với từng dòng in các thông tin học phần, và các danh sách con chứa các phần tử với kiểu dữ liệu dạng chuỗi, ứng với từng từ trong chuỗi đặc biệt (các chuỗi con cách nhau bằng khoảng trắng và được cắt vào một mảng bằng phương thức Split(‘ ‘)). Chuỗi đặc biệt chứa các từ mang thông tin và ngăn cách nhau bởi khoảng trắng, chuỗi có chứa kí tự chấm than ‘!’ với ý nghĩa yêu cầu xuống dòng (thêm danh sách con mới).

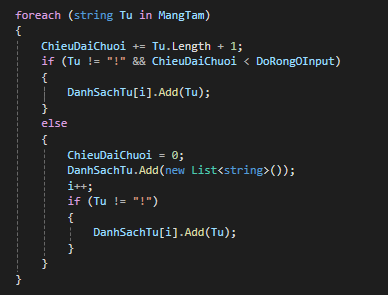
**Tham số:**

* y, x là tọa độ hàng và cột để in thông tin theo khung cho mỗi học phần với độ rộng ô.
* Tạo chuỗi lưu cần in. Khi gặp kí tự “!” sẽ xuống dòng mới để ghi thông tin tiếp theo.
* Tạo một danh sách lưu các chuỗi

A screen shot of a computer

Description automatically generated

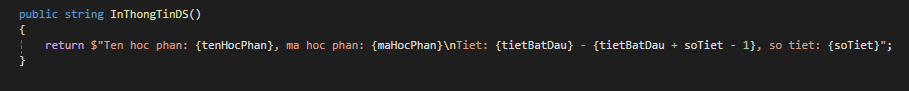
* Nếu các từ đồng thời khác “!” và chiều dài < DoRongOInput thì được in trên 1 hàng, còn lại thì xuống 1 hàng mới.



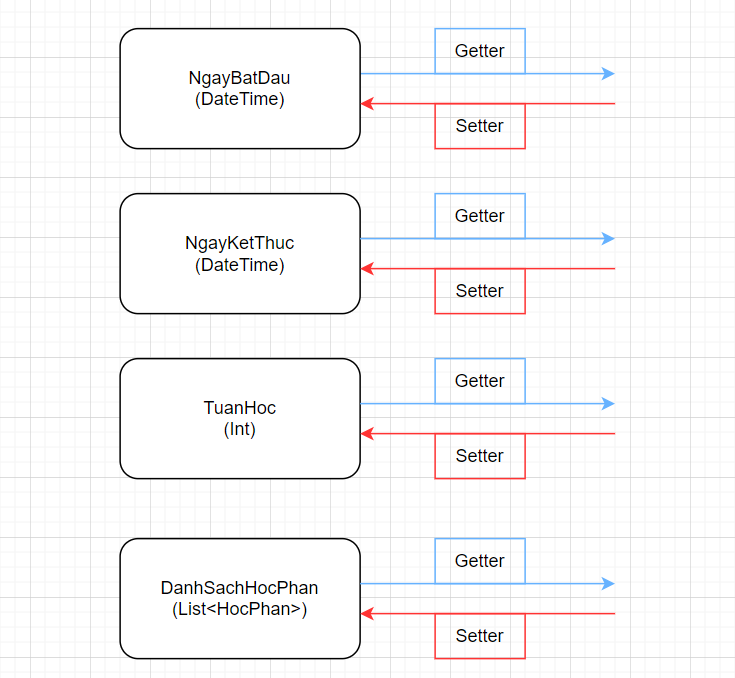
* Vòng lặp để duyệt các danh sách con (hàng), in ra thông in và trả về tổng số hàng cần thiết để in ở học phần này (return i)

A screen shot of a computer code

Description automatically generated

* **Phương thức InThongTinDS()**
* In ra thông tin về tên học phần, mã học phần, các tiết học (bắt đầu – kết thúc) của môn học, số tiết của môn học này trong một tuần.

1. **Class ThoiKhoaBieu**



Lớp thời khóa biểu gồm 4 trường dữ liệu (thuộc tính): ngayBatDau là ngày bắt đầu của thời khóa biểu tuần học đó với kiểu dữ liệu là DateTime; ngayKetThuc là ngày kết thúc của tuần học đó với kiểu dữ liệu là DateTime, ngày kết thúc được tính toán bằng cách cộng thêm 6 ngày vào ngày bắt đầu; tuanHoc là thứ tự tuần học trong thời khóa biểu của 1 năm học với kiểu dữ liệu int; danhSachHocPhan là 1 đối tượng danh sách với kiểu dữ liệu của các phần tử chứa trong nó là đối tượng HocPhan, dùng để lưu trữ các học phần có trong thời khóa biểu đó. Các thuộc tính tương ứng của các trường dữ liệu đều có đầy đủ các setter và getter cho phép lấy và gán giá trị từ bên ngoài.

Lớp cung cấp 2 cách thức khởi tạo: khởi tạo với các giá trị ban đầu và khởi tạo với các giá trị được truyền vào.

**Lớp thời khóa biểu có tổng cộng 8 phương thức:**

A diagram of a network

Description automatically generated

**Phương thức sao lưu:** SaoLuu(). Phương thức có tham số truyền vào là 1 đối tượng thời khóa biểu, phương thức này có tác dụng chép tất cả các dữ liệu trong đối tượng thời khóa biểu hiện tại (this) vào các trường dữ liệu tương ứng trong đối tượng thời khóa biểu khác được truyền vào.

A diagram of a flowchart

Description automatically generated

**Phương thức nhập danh sách học phần:** NhapDanhSachHocPhan(). Phương thức có tác dụng nhập danh sách các học phần có trong thời khóa biểu. Phương thức này sẽ chỉ được gọi bởi phương thức nhập thời khóa biểu bên dưới, nên access modifier của phương thức nhập danh sách học phần sẽ là private, nhằm hạn chế quyền chỉnh sửa trường dữ liệu danhSachHocPhan trong đối tượng thời khóa biểu.

A diagram with text and symbols

Description automatically generated with medium confidence

**Phương thức nhập thời khóa biểu:** NhapTKB(). Phương thức này có tác dụng nhập các trường dữ liệu cần thiết cho đối tượng thời khóa biểu là tuần học đầu tiên trong năm học.

**Phương thức chỉnh sửa thời khóa biểu:** ChinhSuaTKB(). Phương thức có tác dụng chỉnh sửa các học phần hiện có trong thời khóa biểu (tuần học đang xét) bằng cách in ra danh sách các học phần hiện có trong thời khóa biểu dưới dạng một menu có số để lựa chọn, người sử dụng có thể chỉnh sửa học phần bằng cách nhấn số tương ứng hoặc thoát chỉnh sửa.

A grid with arrows and numbers

Description automatically generated

**Phương thức chuyển đổi thứ trong năm từ dạng mã số thành một chuỗi:** TraVeThu(). Phương thức có tham số truyền vào là mã số thứ – bắt đầu từ số 0 là thứ hai – dưới dạng số nguyên và trả về một chuỗi là thứ ngày tương ứng trong tuần.

**Phương thức in khung thời khóa biểu:** InTKB(). Đây là 1 trong 3 phương thức giúp in ra được thời khóa biểu trên màn hình console. Phương thức InTKB() nhận vào 2 tham số là độ rộng của các cột [*DoRongOInput*] của thời khóa biểu và hàng bắt đầu in [*HangInHPInput*] trên màn hình console – Bắt đầu từ hàng số 0 tính từ hàng trên cùng. Sau khi có được 2 tham số đầu vào, con trỏ chuột trên màn hình console được đặt tại vị trí cột thứ 0, vị trí hàng là [*HangInHPInput*] bằng lệnh SetCursorPosition(), sau đó in các thông tin tuanHoc, ngayBatDau và ngayKetThuc trên 2 hàng.

A white background with black lines

Description automatically generated with medium confidence

Tiếp theo, phương thức sẽ in hàng các thứ ngày trong tuần, in thứ hai bắt đầu từ cột thời khóa biểu thứ 2 (cột thời khóa biểu đầu tiên dành cho việc in phòng học) và in ở vị trí hàng thứ [*HangInHPInput*] + 3, mỗi cột của thời khóa biểu rộng bằng *DoRongO* = [*DoRongOInput*] và ngăn cách nhau bởi các kí tự ‘|’. Các kí tự ‘|’ được in ở những vị trí cột [vt\_1 \* *DoRongO*] với vt\_1 nằm trong đoạn [1, 8]. Các chuỗi thứ ngày (“Thu hai”, “Thu ba”,…) được in ở những vị trí cột [vt\_2 \* *DoRongO* + *DoRongNuaO*] với vt\_2 nằm trong đoạn [1, 7] và *DoRongNuaO* = ((*DoRongO* - 7) / 2), số 7 trong phép trừ lấy từ chiều dài của chuỗi “Thu hai” (lấy trung bình chiều dài chuỗi từ chuỗi “Thu hai” đến “Chu nhat” là 7). Xuống dòng và in kí tự ‘-’ vừa đủ với tổng chiều dài bề ngang của 8 cột của thời khóa biểu (8 \* DoRongO).

A close-up of a math problem

Description automatically generated

Các học phần được in lần lượt theo từng hàng bằng cách tạo một danh sách học phần tạm với các phần tử được sao chép từ danhSachHocPhan trong đối tượng thời khóa biểu, các học phần được sắp xếp cùng 1 phòng học sẽ được tìm kiếm trên danh sách học phần tạm và in trên cùng một hàng. Sau mỗi lần in các học phần có cùng một phòng học thì xóa hết những học phần đó ra khỏi danh sách tạm bằng phương thức RemoveAll(), đó là lí do vì sao phải tạo một danh sách mới sao chép từ danhSachHocPhan, tránh làm mất trường dữ liệu danhSachHocPhan của đối tượng thời khóa biểu. Các học phần được in ở vị trí cột bằng ((mã số thứ của học phần + 1) \* DoRongO + 1), mã số thứ được bắt đầu bằng 0 (“Thu hai”), vị trí cột ((mã số thứ của học phần + 1) \* DoRongO) là vị trí in kí tự ‘|’ nên phải cộng thêm 1 để bắt đầu in thông tin học phần ở vị trí cột tiếp theo.

A diagram of a flowchart

Description automatically generated

A screenshot of a calendar

Description automatically generated

Tùy vào lượng thông tin có trong một đối tượng học phần mà khoảng không gian cần thiết cho việc in thông tin học phần có thể thay đổi (chỉ thay đổi về số hàng cần thiết vì độ rộng cột đã được cho cố định. Để nắm bắt được số lượng hàng cần thiết lớn nhất, biến *ChieuCaoMax* được sử dụng để lưu giá trị này. Sau mỗi lần in thông tin một học phần vừa vặn vào một ô trong thời khóa biểu, nó sẽ trả về số lượng hàng cần thiết để in đủ thông tin cho học phần đó, giá trị này sau đó được đem đi so sánh với *ChieuCaoMax*, nếu giá trị đó lớn hơn *ChieuCaoMax* thì cập nhật lại giá trị *ChieuCaoMax* (bằng với giá trị đó). Sau mỗi lần in thông tin các học phần có cùng phòng học (tức hoàn thành in một hàng của thời khóa biểu), biến *HangInHP* được cộng dồn bằng (*ChieuCaoMax +1)*.

**Biến *HangInHP* có 3 ý nghĩa:**

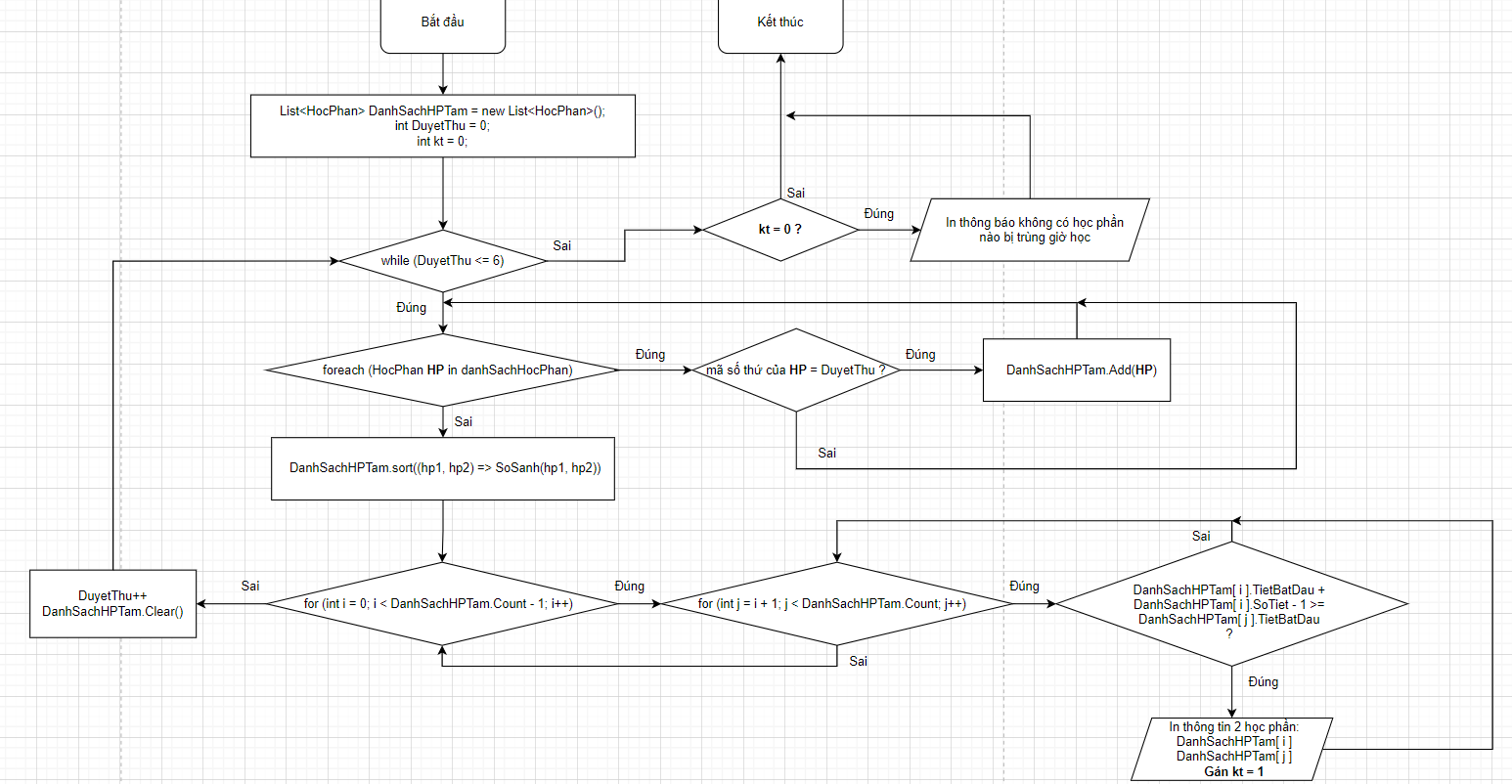
* Là vị trí hàng (trên console) để in thông tin phòng học (hàng của thời khóa biểu) tiếp theo.
* Khi trừ bớt đi 1 sẽ ra vị trí hàng (trên console) để in các kí tự ‘-‘ ngăn cách giữa các hàng (của thời khóa biểu)
* Sau khi hoàn thành in toàn bộ thời khóa biểu, biến *HangInHP* sẽ lưu vị trí hàng sau cùng, được dùng cho vòng lặp for để in các kí tự ‘|’ ngay dưới đây.

A screenshot of a computer screen

Description automatically generated

Kết thúc in các học phần, một vòng lặp for được chạy để in các kí tự ‘|’ ở các vị trí cột [vt\_1 \* *DoRongO*], ở các tất cả các vị trí hàng từ [*HangInHPInput*] + 4 cho đến *HangInHP*. Kết thúc in, phương thức in thời khóa biểu sẽ trả về *HangInHP* với ý nghĩa là vị trí hàng (trên console) cuối cùng của thời khóa biểu.

**Phương thức so sánh:** SoSanh(). Phương thức nhận vào 2 đối tượng học phần (HP1 và HP2) và tiến hành so sánh dữ liệu tiết bắt đầu của HP1 và HP2. Nếu tiết bắt đầu của HP1 lớn hơn tiết bắt đầu của HP2 thì trả về 1, ngược lại trả về 0. Phương thức so sánh này đóng vai trò là bộ so sánh được dùng cho hàm sort danh sách, chứa trong phương thức kiểm tra trùng ngay dưới đây.

****

**Phương thức kiểm tra trùng giờ học của học phần:** KiemTraTrung(). Phương thức có tác dụng kiểm tra xem có học phần nào có giờ học bị trùng so với học phần khác hay không, bằng cách tạo một danh sách học phần tạm dùng để lưu các học phần diễn ra cùng ngày trong tuần (có cùng mã số thứ), sau đó sắp xếp danh sách học phần này tăng dần theo tiết bắt đầu (sử dụng phương thức so sánh ở phía trên làm bộ so sánh). Hai vòng lặp for lồng – for i ở ngoài, for j ở trong – sẽ được dùng để duyệt từng cặp phần tử khác nhau trong danh sách, kiểm tra xem nếu học phần [i] có giờ học trùng với học phần [j] không bằng điều kiện (tiết bắt đầu của học phần [i] + số tiết học phần [i] - 1 >= tiết bắt đầu học phần [j]).

1. **Class QuanLyTKB**

Lớp quản lý thời khóa biểu giúp quản lý nhiều thời khóa biểu (tuần học), vì thế nên đối tượng quản lý thời khóa biểu cũng chính là tập hợp thời khóa biểu của một năm học/một học kì. Lớp sử dụng một danh sách với kiểu dữ liệu của các phần tử chứa trong nó là đối tượng HocPhan, trường dữ liệu này có tên gọi là *danhSachTKB*. Lớp quản lý thời khóa biểu cung cấp 2 phương thức khởi tạo: khởi tạo rỗng và khởi tạo có truyền vào danh sách học phần (trường dữ liệu duy nhất trong lớp).

**Phương thức nhân bản tuần học:** NhanBanTuanHoc(). Phương thức này được truyền vào một đối tượng thời khóa biểu và có tác dụng nhân bản n tuần học (thời khóa biểu được truyền vào) giống hệt nhau trong một năm học. Số n được nhập bởi người dùng và được đảm bảo n nhập vào là số nguyên dương bằng việc phối hợp giữa vòng lặp while và khối try/catch, tương tự như một số phương thức trong lớp thời khóa biểu. Đối tượng thời khóa biểu truyền vào [*TKBInput*] sẽ được sao lưu vào một đối tượng thời khóa biểu mới *TKBtuan1*, điều nào nhằm bảo toàn dữ liệu cho đối tượng thời khóa biểu gốc [*TKBInput*].

Đối tượng thời khóa biểu mới *TKBtuan1* sau khi nhận các giá trị từ [*TKBInput*] sẽ được thêm vào danh sách thời khóa biểu (*danhSachTKB)*. Tiếp theo, sử dụng 2 đối tượng thời khóa biểu mới được tạo ra là *TKBTruoc* và *TKBTiep* cùng với vòng lặp for duyệt từ [1, n – 1] để thêm n – 1 thời khóa biểu còn lại vào danh sách thời khóa biểu (*danhSachTKB*).

A diagram of a computer program

Description automatically generated

A diagram of a diagram

Description automatically generated with medium confidence

**Phương thức làm trống thời khóa biểu:** LamTrongTKB(). Phương thức có tác dụng làm trống một tuần học được chỉ định, không xóa thời khóa biểu ra khỏi danh sách thời khóa biểu mà chỉ làm trống danh sách học phần chứa trong nó, tức thời khóa biểu của một tuần học sau khi được làm trống sẽ chỉ giữ lại các thông tin: số thứ tự tuần học, ngày bắt đầu và ngày kết thúc. Sử dụng phương thức Clear() để làm trống danh sách học phần chứa trong thời khóa biểu.

A diagram of a computer

Description automatically generated with medium confidence

**Phương thức chỉnh sửa tuần học được chỉ định:** ChinhSuaTuanHoc(). Phương thức cho phép người dùng nhập số thứ tự tuần học cần chỉnh sửa rồi gọi đến phương thức ChinhSuaTKB() của tuần học (đối tượng thời khóa biểu) đó.

A diagram of a flowchart

Description automatically generated

**Phương thức in thời khóa biểu theo tuần:** InTKBTheoTuan(). Phương thức nhận vào tham số là độ rộng ô (cột) của thời khóa biểu. Phương thức này chủ yếu để lập một danh sách các tuần học mà người dùng muốn in ra màn hình console. Sau mỗi lần in một thời khóa biểu, phương thức sẽ làm trống màn hình console ở khoảng không gian đã in thời khóa biểu trước đó để chuẩn bị không gian cho việc in thời khóa biểu tiếp theo.

Bắt đầu phương thức, người dùng được yêu cầu nhập một chuỗi các số cách nhau bởi dấu khoảng cách, chuỗi được lưu vào biến s ngay khi bắt đầu vòng lặp while, DanhSachTuanHoc được dùng để lưu các số nguyên dương là số thứ tự tuần học cần in, điều kiện để kết thúc vòng lặp while là DanhSachTuanHoc có chứa phần tử. Sau khi chuỗi được nhập, công việc tách các số nguyên ra từ chuỗi s sẽ tiếp diễn trong vòng lặp foreach.

A diagram with text and symbols

Description automatically generated

Để tạo thuận lợi cho việc sử dụng, phương thức cung cấp 2 cách thức nhập số tuần học cần in và ứng với mỗi cách nhập sẽ là một nhánh logic giúp lọc ra số nguyên từ chuỗi con, được tách ra từ chuỗi s (các chuỗi con giờ đây được lưu trong MangTuanHoc) với chuỗi i trong vòng lặp foreach duyệt lần lượt các phần tử (chuỗi con) trong MangTuanHoc:

* Người dùng có thể nhập đơn lẻ từng số nguyên ứng với số thứ tự tuần mà mình muốn in. Điều kiện để vào nhánh logic này là chuỗi i không chứa kí tự ‘:’. Trong khối try/catch, các câu lệnh có thể gây ra ngoại lệ là lệnh chuyển chuỗi i thành số nguyên (nếu gây ra ngoại lệ tức chuỗi i không phải là số nguyên) và khối lệnh tạo ngoại lệ mới nếu số nguyên i bé hơn hoặc bằng 0. Nếu có bất kỳ ngoại lệ nào được bắt, tiến trình sẽ tiếp tục vòng lặp foreach; nếu không, chuyển i thành số nguyên và lưu vào biến j, nếu j chưa có trong danh sách thì thêm j vào DanhSachTuanHoc rồi mới tiếp tục vòng lặp foreach. Đây là nhánh logic thứ nhất.

A diagram of a flowchart

Description automatically generated

* Nhập <tuần bắt đầu>:<tuần kết thúc> để in tất cả các tuần từ <tuần bắt đầu> đến <tuần kết thúc>, ví dụ: nhập 1:5 tương ứng với 1 2 3 4 5. Điều kiện để vào được nhánh logic này là chuỗi i phải chứa kí tự ‘:’. Các ngoại lệ có thể bắt là: không thể chuyển một phần tử trong MangPhu thành số nguyên được (tức ít nhất 1 trong 2 không phải số nguyên), có ít nhất một số nguyên không dương. Nếu bắt được ngoại lệ, tiếp tục vòng lặp foreach; nếu không, chuyển 2 phần tử trong MangPhu thành 2 số nguyên lưu vào k0 và k1, rồi thêm vào DanhSachTuanHoc tất cả các số nguyên (chưa có trong danh sách) chạy trong đoạn [k0, k1]. Đây là nhánh logic thứ hai.

A diagram of a computer program

Description automatically generated with medium confidence

Các giá trị người dùng nhập vào bị trùng được xử lí để cho ra một danh sách các giá trị độc nhất, ví dụ: nhập 1 2 3 4 3:6 tương ứng với 1 2 3 4 5 6. Kết thúc vòng lặp foreach, nếu không có bất kì số nguyên nào được ghi nhận thì in ra yêu cầu nhập lại rồi tiếp tục vòng lặp while. Kết thúc vòng lặp while, danh sách tuần học được sắp xếp tăng dần, vòng lặp for thứ 2 sẽ duyệt tìm các thời khóa biểu có số thứ tự tuần học tương ứng để in ra màn hình console. Tham số độ rộng ô được truyền vào phương thức sẽ được tiếp tục truyền vào đối số đầu tiên của phương thức in thời khóa biểu, đối số thứ hai là vị trí hàng trên console để in thời khóa biểu và được giữ cố định xuyên xuốt quá trình in các tuần học.

A diagram of a flowchart

Description automatically generated

A diagram of a computer

Description automatically generated

**Phương thức hủy học phần:** HuyHocPhan(). Phương thức cho phép xóa một học phần trên tất cả các thời khóa biểu được quản lý, tên học phần được nhập bởi người dùng.

A diagram of a flowchart

Description automatically generated

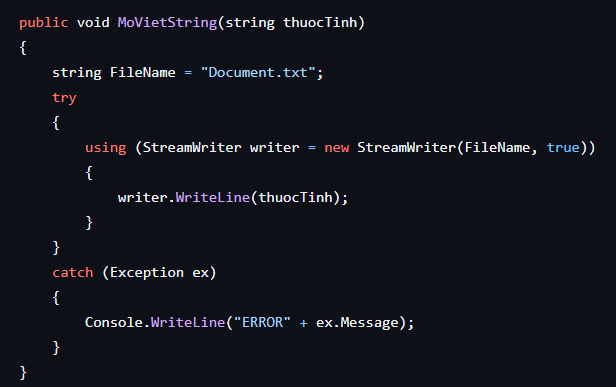
**Phương thức thêm học phần:** ThemHocPhan(). Phương thức cho phép thêm một học phần trên tất cả các thời khóa biểu được quản lý, thông tin học phần được nhập bởi người dùng.

1. **Class TapTin**

Ở class TapTin này nhằm mục đích tạo ra để ghi những thông tin mà người dung nhập sang 1 file txt để tiện cho người dùng theo dõi thông tin sinh viên và thời khóa biểu. Trong đó chúng tôi đã tạo sẵn 1 file txt mang tên “Document” ở vị trí thư mục bin.

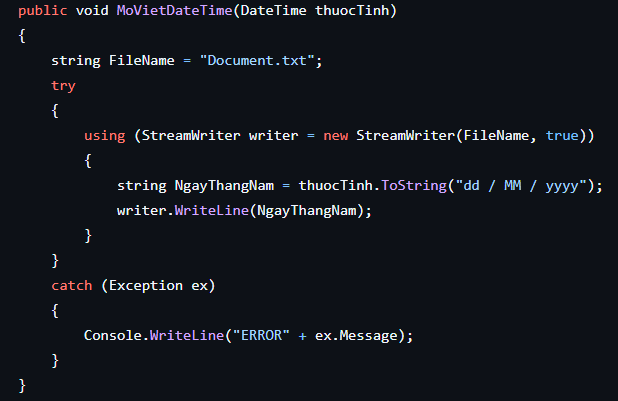
**Phương thức:**

* **Phương thức MoVietString:**



**Ở phương thức này để ghi giá trị người dùng nhâp ở định dạng string:**

* Đặt tên biến Filename chứa tên file txt đã lưu trong thư mục bin.
* Trong try thì dùng hàm StreamWriter để ghi dữ liệu dạng chuỗi khi người dùng nhập trên console có tham số truyền là thuocTinh.
* Trong catch nếu ghi vô file txt không được thì sẽ thông báo ra lỗi .
* **Phương thức MoVietDateTime:**



**Ở phương thức này để ghi giá trị người dùng nhâp ở định dạng DateTime :**

* Đặt tên biến Filename chứa tên file txt đã lưu trong thư mục bin.
* Trong try, đầu tiên thì sẽ có biến NgayThangNam mang định dạng string để chuyển định dạng DateTime thành string sau đó dùng hàm StreamWriter để ghi dữ liệu dạng chuỗi khi người dùng nhập trên console có tham số truyền là thuocTinh.
* Trong catch nếu ghi vô file txt không được thì sẽ thông báo ra lỗi .
* **Phương thức MoVietNumber:**

A computer screen shot of a program

Description automatically generated

**Ở phương thức này để ghi giá trị người dùng nhâp ở định dạng int:**

* Đặt tên biến Filename chứa tên file txt đã lưu trong thư mục bin
* Trong try thì dùng hàm StreamWriter để ghi dữ liệu dạng số thực khi người dùng nhập trên console có tham số truyền là thuocTinh
* Trong catch nếu ghi vô file txt không được thì sẽ thông báo ra lỗi
* **Phương thức XoaPhanTuCuoi:**

A screen shot of a computer program

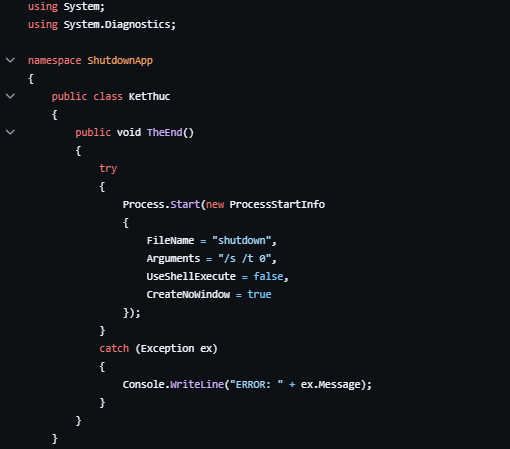
Description automatically generated

**Ở phương thức này tạo ra nằm mục đích xóa dòng cuối ở file txt khi người dùng nhập sai định dạng theo yêu cầu:**

* Đặt tên biến Filename chứa tên file txt đã lưu trong thư mục bin.
* Tạo 1 cái list chứa các dòng bằng hàm đọc ReadAllLines.
* Nếu số lượng dòng lớn hơn 1 thì sẽ xóa dòng cuối và viết cập nhật lại hết tất cả các dòng.

1. **Class KetThuc**

Class KetThuc nhằm mục đích shutdown máy giúp người dùng không muốn coi thời khóa biểu nữa thì sử dụng mang đến tiện ích người dùng.

****

Class này phải sử dụng namespace ShutdownApp để sử dụng hàm để shutdown máy. Trong phương thức TheEnd chúng tôi sẽ sử dụng lệnh Process.Start để gọi đối tượng ProcessStartInfo chứa các thông tin cấu hình cho tiến trình.

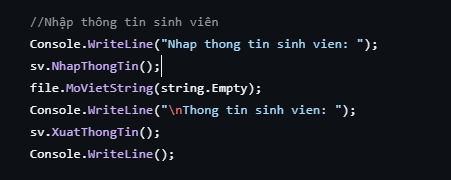
* **FileName:** tên chương trình cần thực thi “shutdown”.
* **Arguments:** truyền các tham số cho lệnh shutdown với “/s” là tắt máy còn “/t 0” là thời gian chờ trước khi tắt là 0s.
* **UseShellExecute:** true là không tạo cửa số console mới cho tiến trình. Điều này sẽ làm cho lệnh chạy trong nền không hiện bất kỳ giao diện.

## Program

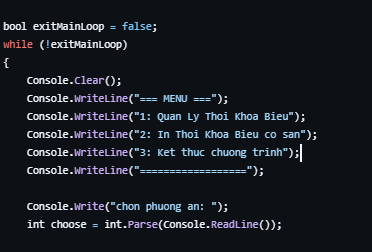
Ở Program sẽ chỉnh kích thước Console và cho chạy MENU để người dùng muốn chọn option để quản lý thời khóa biểu dành cho sinh viên:



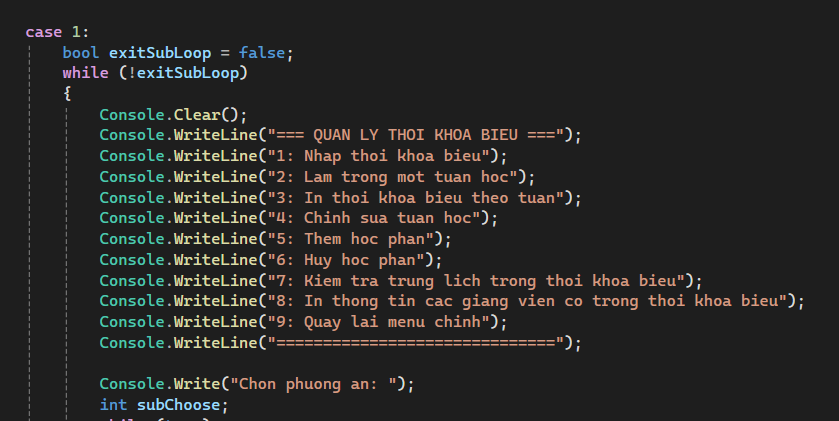
Ở đây sẽ khởi tạo đối tượng class ThoiKhoaBieu, QuanLyTKB, TapTin và SinhVien. Đồng thời sẽ điều chỉnh kích thước console khi ở chế độ window host để xuất ra giao diện thời khóa biểu sao cho hợp lý.



Ở đây sẽ yêu cầu nhập thông tin sinh viên và sẽ xuất ra dữ liệu cá nhân thông tin sinh viên sau khi người dùng nhập trên console.

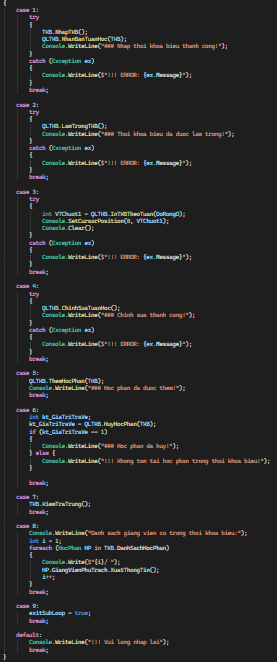


Chúng tôi sẽ tạo ra vòng lặp while để hiện ra giao diện menu để yêu cầu người dùng chọn option để thực thi chương trình. Và ở trong vòng lặp while sẽ có hàm switch case chính để cho người dùng chọn option để thực thi chương trình:



* **Case 1:**

Trong case 1 sẽ có switch case phụ yêu cầu chọn 1 trong 7 case liên quan đến nhập in sủa thêm hủy để quản lý thời khóa biểu của sinh viên.



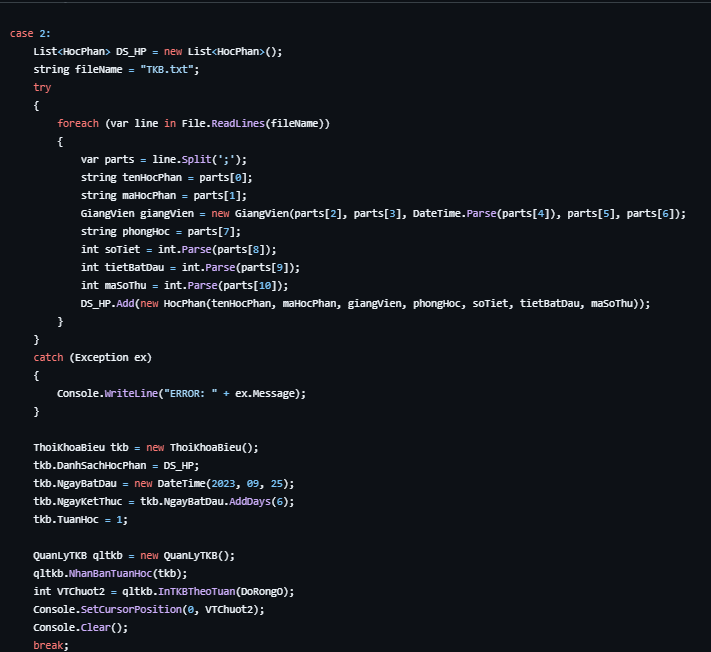
**Thì ở trong switch case phụ sẽ có 7 case:**

* **case 1** yêu cầu người dung nhập thời khóa biểu
* **case 2** sẽ làm trống một tuần học
* **case 3** sẽ in ra thời khóa biểu mà người dùng đã nhập trước đó và sau khi in sẽ có hàm clear cả console
* **case 4** sẽ chỉnh sửa thời khóa biểu
* **case 5** sẽ them học phần trong thời khóa biểu
* **case 6** sẽ hủy học phần
* **case 7** sẽ kiểm tra trùng lịch trong thời khóa biểu
* **case 8 sẽ** in thông tin các giảng viên có trong thời khóa biểu
* **case 9** sẽ cho biến exitSubLoop là true để out vòng lặp while
* Nếu người dùng nhâp ko đúng option từ 1 tới 7 sẽ thông báo yêu cầu nhập lại

Sau khi chạy xong 1 trong 7 case thì chương trình sẽ hỏi người dùng bạn có muốn tiếp tục ko, khi ấy nếu người dung nhập “yes” thì vòng lặp while tiếp tục chạy và sẽ yêu cầu chọn option còn nhập “no” sẽ thoát vòng lặp while và trở về menu chính.

* **Case 2:**

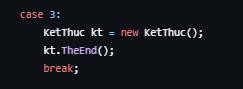
Trong case 2 thì sẽ in ra thời khóa biểu có sẫn được lấy từ tập tin file txt “TKB.txt” :



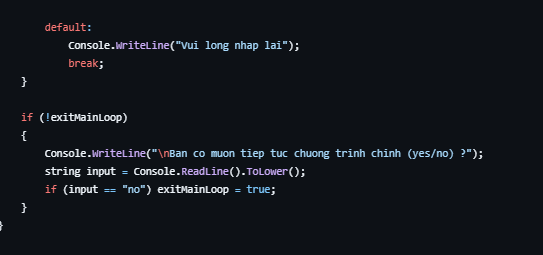
Chúng tôi sẽ khởi tạo list mang tên DS\_HP từ đối tượng class HocPhan. Sau đó sẽ đọc file txt và tách từng giá trị trong mỗi dòng để add vô list DS\_HP. Sau đó khởi tạo đối tượng class ThoiKhoaBieu và QuanLyTKB để in ra thời khóa biểu và sau đó clear Console.

* **Case 3:**

Trong case 3 sẽ thưc thi shutdown máy nhằm giúp người dùng sau khi coi thời khóa biểu cảm thấy mệt mỏi và không muốn coi nữa sẽ tắt máy giúp tiện lợi cho user:



Khởi tạo đối tượng class KetThuc và cho chạy phương thức TheEnd để kết thúc.



Nếu người dung không nhập đúng 1 trong 3 case chương trình sẽ yêu cầu người dùng nhâp lại. Sau khi chạy xong 1 trong 3 case thì chương trình sẽ hỏi người dùng bạn có muốn tiếp tục ko, khi ấy nếu người dung nhập “yes” thì vòng lặp while tiếp tục chạy và sẽ yêu cầu chọn option còn nhập “no” sẽ thoát vòng lặp while và thoát chương trình.

# CHƯƠNG 4: KẾT LUẬN VÀ ĐÁNH GIÁ

1. **Những điểm mạnh**

* Quản lý học phần theo thời gian thực: Có thể thay đổi các thông tin trong thời khóa biểu bất kì lúc nào.
* Lưu lại lịch sử thời khóa biểu của nhiều học kỳ, thuận tiện cho tra cứu, tìm kiếm.
* Giao diện người dùng: Thiết kế giao diện thân thiện, gần gũi với người dùng, dễ dàng sử dụng, có thể tùy chỉnh giao diện.
* Tích hợp thêm lịch kiểm tra và thi cuối kì của sinh viên, điểm số của từng học phần.
* Tính năng thông báo và nhắc nhở cho sinh viên mỗi khi có bài kiểm tra hoặc bài thi sắp diễn ra.
* Tối ưu hóa thuật toán để tạo ra hiệu suất tốt và ổn định nhất của chương trình.

1. **Những hạn chế**

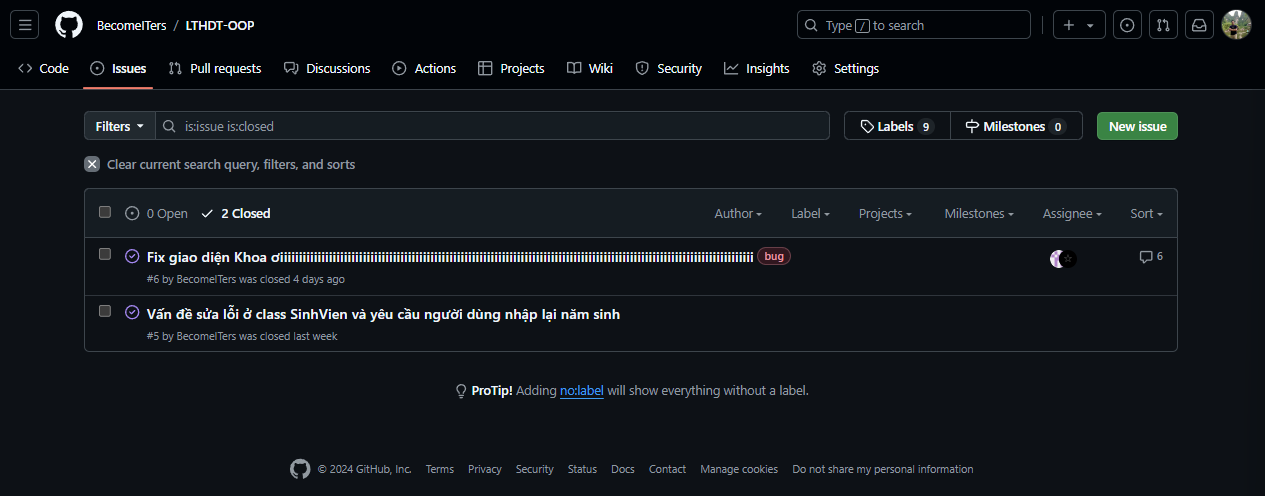
* Cải thiện giao diện người dùng
* Xem lại thời khóa biểu cũ mà người dùng đã nhập trước đó
* Đa ngôn ngữ
* Tính năng Undo/Redo

1. **Bài học kinh nghiệm**

* Lập kế hoạch rõ ràng trước khi bắt đầu triển khai
* Phân chia công việc hợp lý
* Xử lý lỗi và kiểm tra đầu vào
* Tối ưu hóa cấu trúc và giao diện người dùng

1. **Cách giải quyết khó khăn**

Khi các thành viên trong nhóm chạy chương trình và test. Sau khi gặp lỗi và tìm thấy bug sẽ viết thông báo up lên ở mục Issues trên Github và hastag người đảm nhiệm class đó để fix đồng thời nêu lên ý kiến để giải quyết vấn đề dễ dàng hơn.



1. **Kết luận**

Qua một khoảng thời gian học tập và nghiên cứu, nhóm em đã nắm được cơ bản và vận dụng những kiến thức được dạy trên lớp để thực hiện đồ án quản lí thời khóa biểu sinh viên. Đồ án này đảm bảo tính đầy đủ và chính xác cho thông tin trên thời khóa biểu sinh viên, giúp người dùng dễ dàng theo dõi và tra cứu.

Tuy nhiên đồ án vẫn còn hạn chế và khả năng thay đổi thông tin theo thời gian thực, chưa linh hoạt khi sử dụng. Đồ án được hoàn thành ở mức độ cơ bản, chưa thể áp dụng vào thực tế do cần có sự hiểu biết chuyên sâu về rất nhiều lĩnh vực khác.

Nhóm em mong sẽ nhận được những đóng góp quý báu từ thầy để đồ án được hoàn thiện hơn, tạo nền tảng tốt để tiếp tục học tập các kỹ năng lập trình.