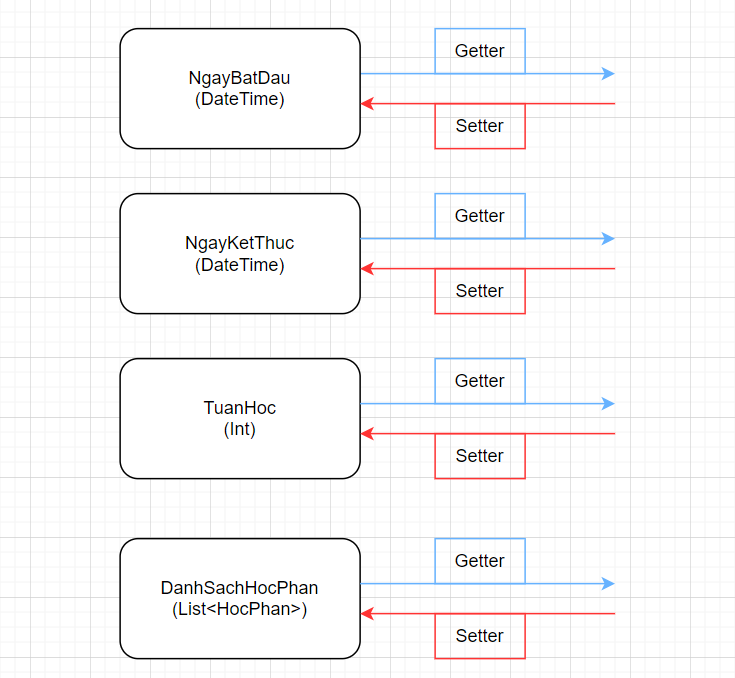
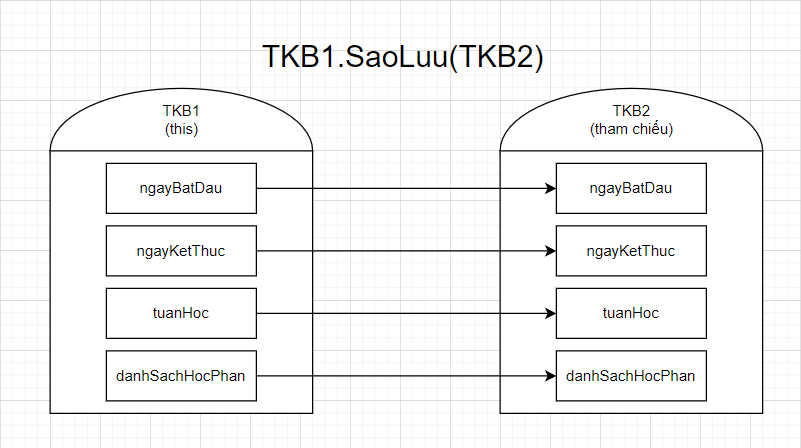
**\*Lớp ThoiKhoaBieu:**



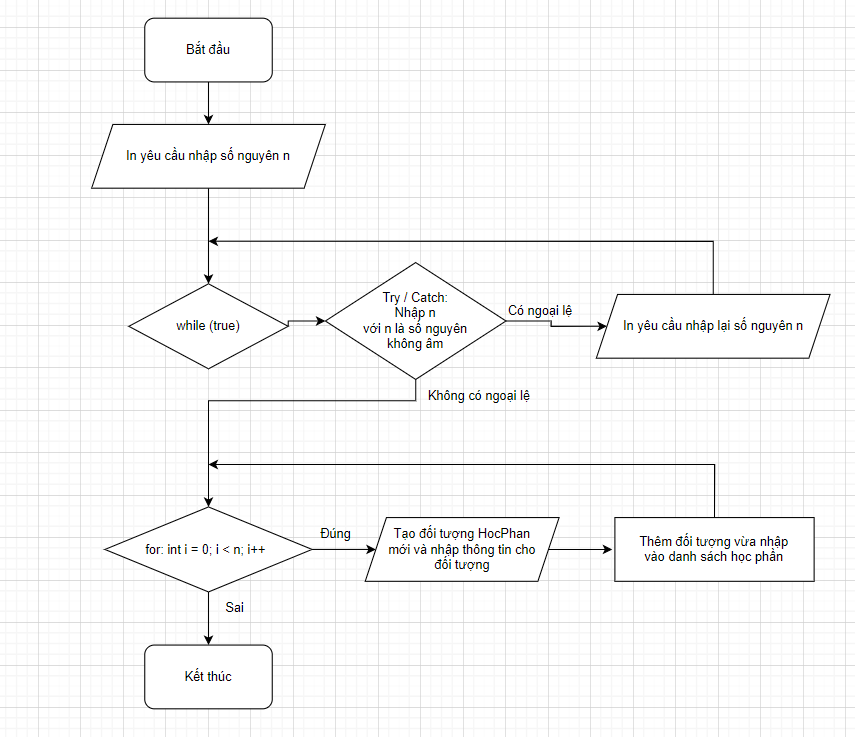
Lớp thời khóa biểu gồm 4 trường dữ liệu (thuộc tính): ngayBatDau là ngày bắt đầu của thời khóa biểu tuần học đó với kiểu dữ liệu là DateTime; ngayKetThuc là ngày kết thúc của tuần học đó với kiểu dữ liệu là DateTime, ngày kết thúc được tính toán bằng cách cộng thêm 6 ngày vào ngày bắt đầu; tuanHoc là 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.

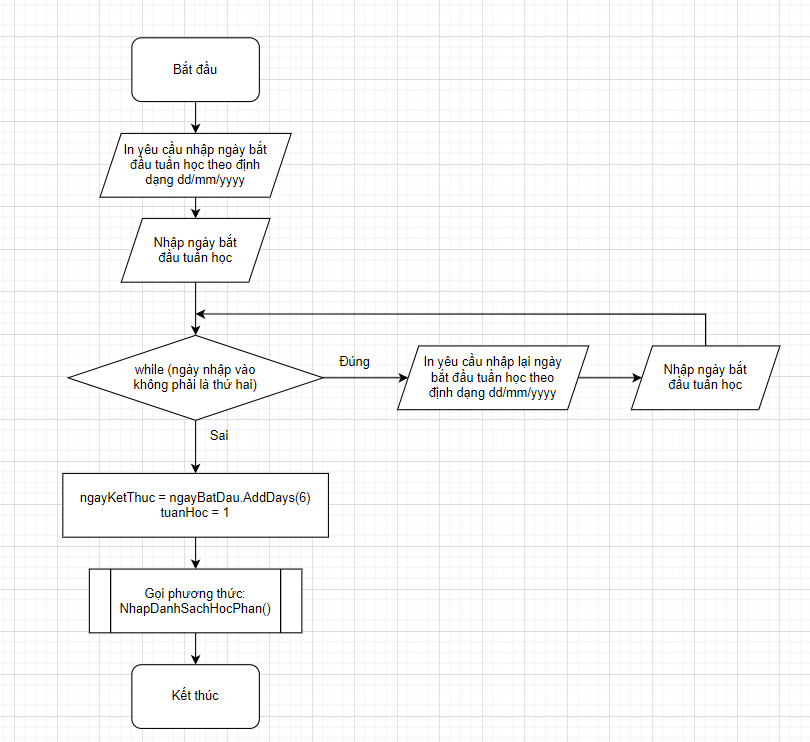
Lớp thời khóa biểu có tổng cộng 8 phương thức:



**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.

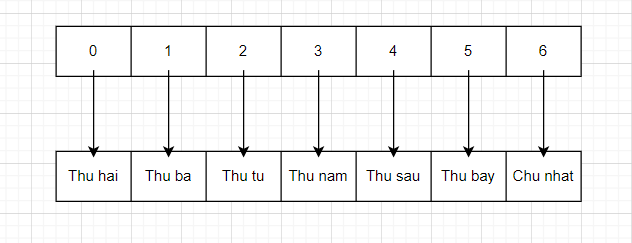


**Phương thức nhập danh sách học phần:** NhapDanhSachHocPhan(). Phương thức có tác dụng nhập danh sách các học phần có trong thời khóa biểu. Phương thức này sẽ chỉ được gọi bởi phương thức nhập thời khóa biểu bên dưới, nên access modifier của phương thức nhập danh sách học phần sẽ là private, nhằm hạn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



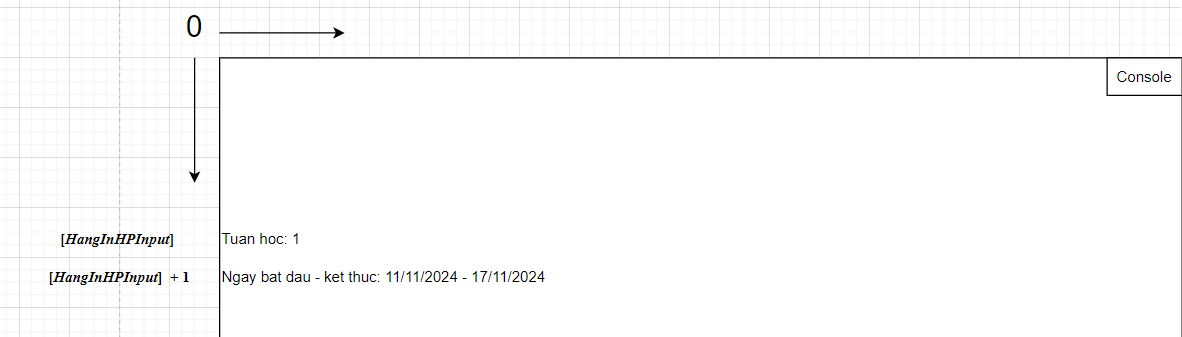
**Phương thức nhập thời khóa biểu:** NhapTKB(). Phương thức này có tác dụng nhập các trường dữ liệu cần thiết cho đối tượng thời khóa biểu là tuần học đầu tiên trong năm học.

**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.

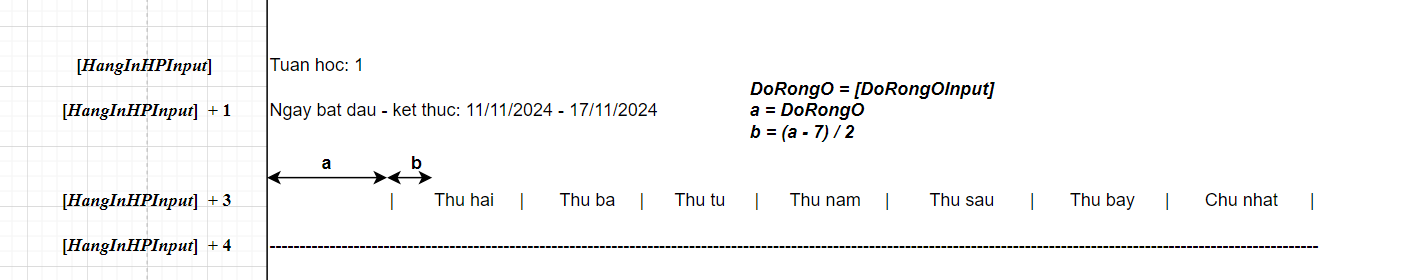


**Phương thức chuyển đổi thứ trong năm từ dạng mã số thành một chuỗi:** TraVeThu(). Phương thức có tham số truyền vào là mã số thứ – bắt đầu từ số 0 là thứ hai – dưới dạng số nguyên và trả về một chuỗi là thứ ngày tương ứng trong tuần.

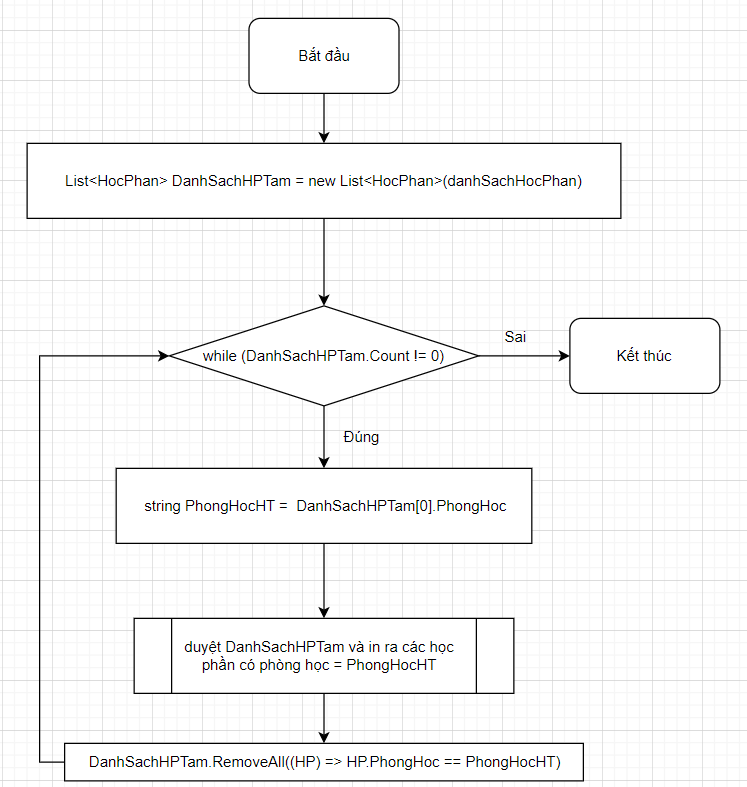
**Phương thức in khung thời khóa biểu:** InTKB(). Đây là 1 trong 3 phương thức giúp in ra được thời khóa biểu trên màn hình console. Phương thức InTKB() nhận vào 2 tham số là độ rộng của các cột [*DoRongOInput*] của thời khóa biểu và hàng bắt đầu in [*HangInHPInput*] trên màn hình console – Bắt đầu từ 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.

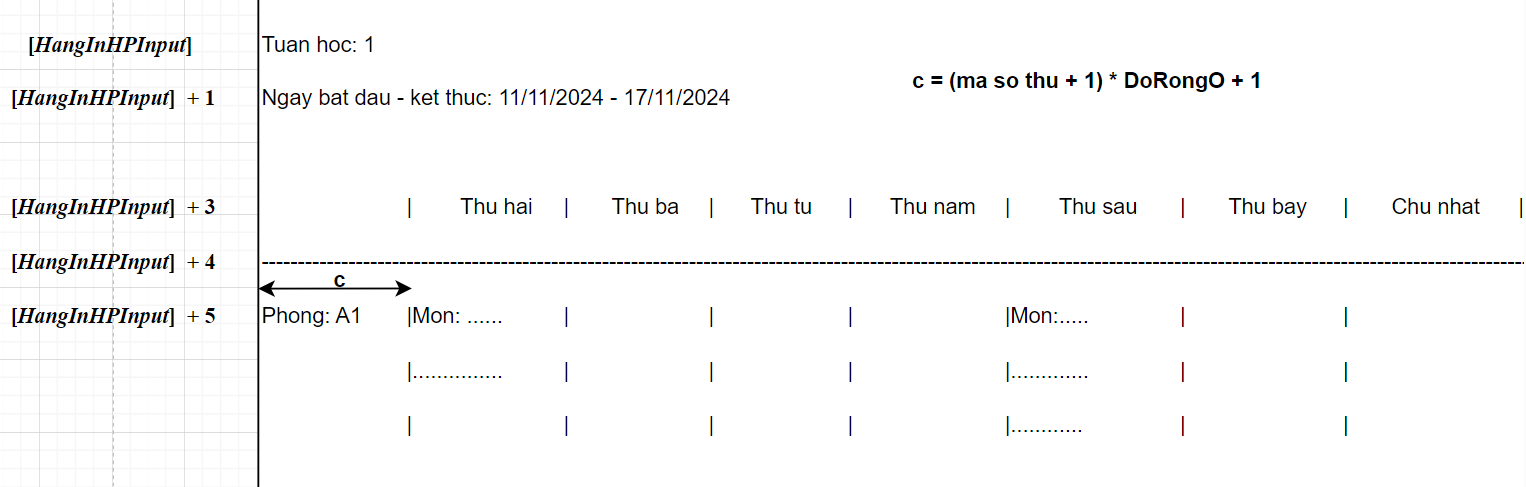


Tiếp theo, phương thức sẽ in hàng các thứ ngày trong tuần, in thứ hai bắt đầu từ cột thời khóa biểu thứ 2 (cột thời khóa biểu đầu tiên dành cho việc in phòng học) và in ở vị trí hàng thứ [*HangInHPInput*] + 3, mỗi cột của thời khóa biểu rộng bằng *DoRongO* = [*DoRongOInput*] và ngăn cách nhau bởi các 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).



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.



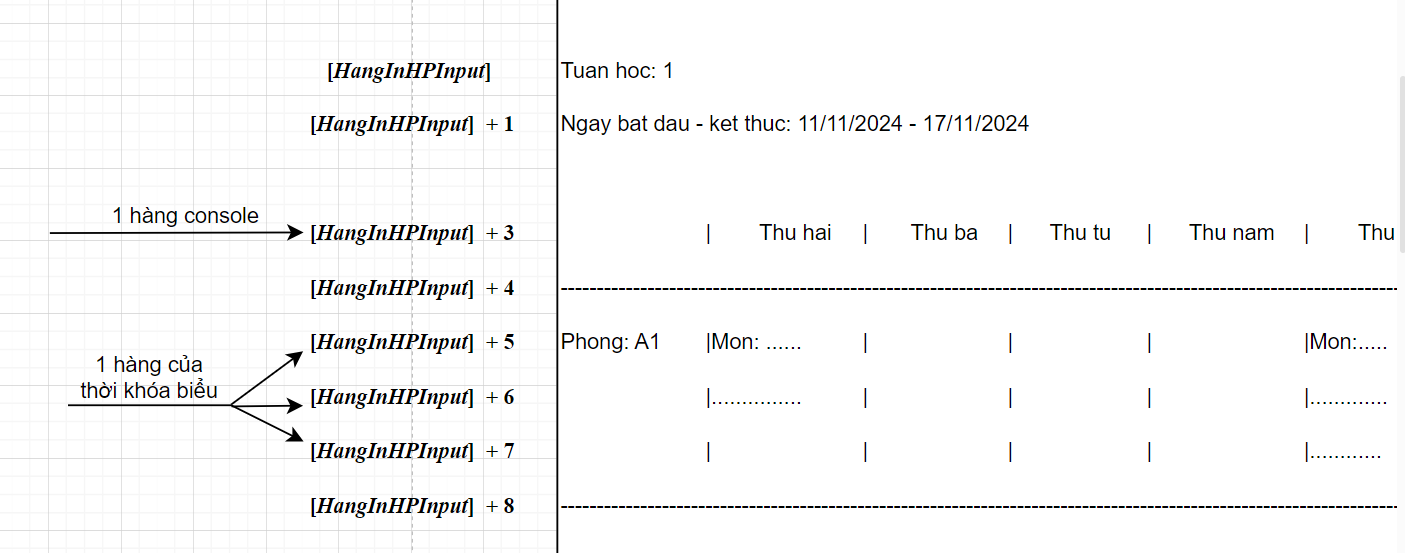


Tùy vào lượng thông tin có trong một đối tượng học phần mà khoảng không gian cần thiết cho việc in thông tin học phần có thể thay đổi (chỉ thay đổi về số hàng cần thiết vì độ rộng cột đã được cho cố định. Để nắm bắt được số lượng hàng cần thiết lớn nhất, biến *ChieuCaoMax* được sử dụng để lưu giá trị 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.

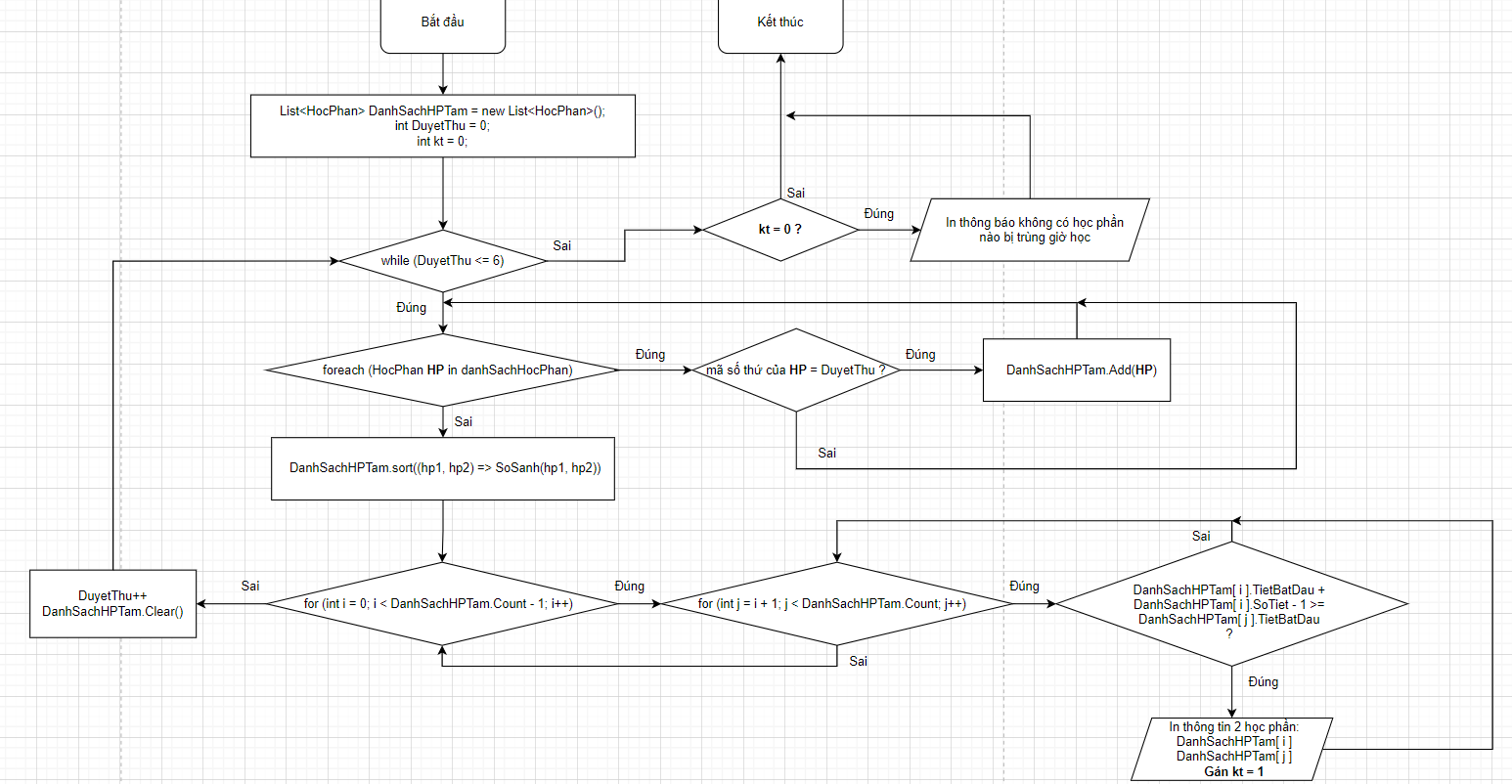
- Khi trừ bớt đi 1 sẽ ra vị trí hàng (trên console) để in các kí tự ‘-‘ ngăn cách giữa các hàng (của thời khóa biểu)

- 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.



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.

****

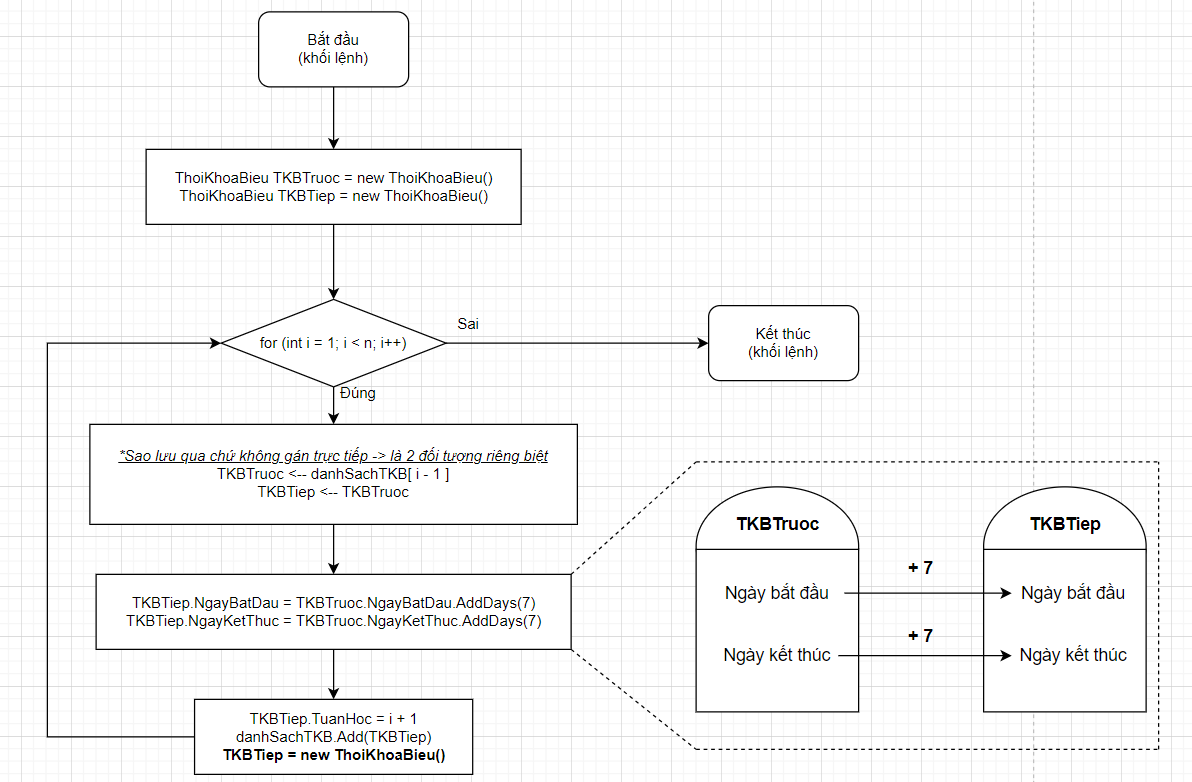
**Phương thức kiểm tra trùng giờ học của học phần:** KiemTraTrung(). Phương thức có tác dụng kiểm tra xem có học phần nào có giờ học bị trùng so với học phần khác hay không, bằng cách tạo một danh sách học phần tạm dùng để lưu các học phần diễn ra cùng ngày trong tuần (có cùng mã số thứ), sau đó sắp 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]).

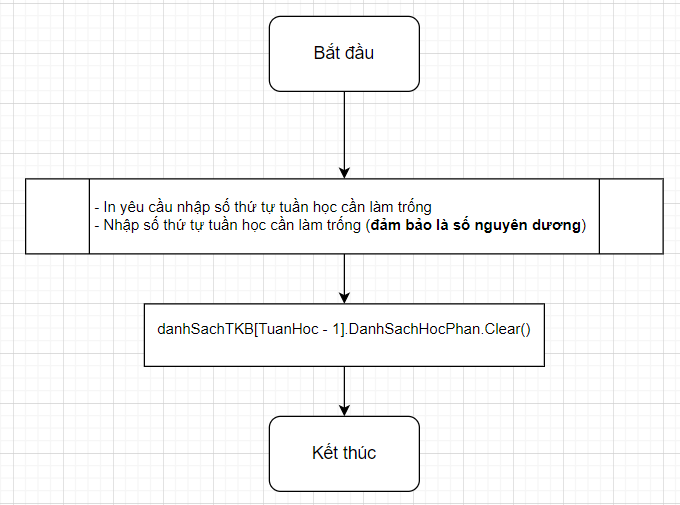
**\*Lớp quản lý thời khóa biểu:**

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).

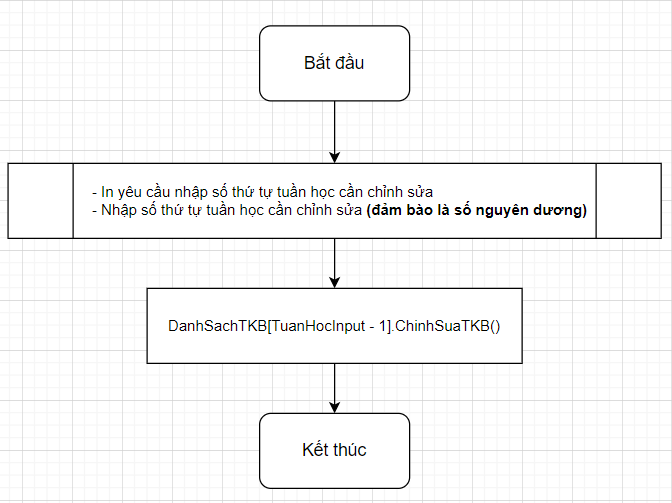
**Phương thức nhân bản tuần học:** NhanBanTuanHoc(). Phương thức này được truyền vào một đối tượng thời khóa biểu và có tác dụng nhân bản n tuần học (thời khóa biểu được truyền vào) giống hệt nhau trong một năm học. Số n được nhập bởi người dùng và được đảm bảo n nhập vào là số nguyên dương bằng việc 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*).

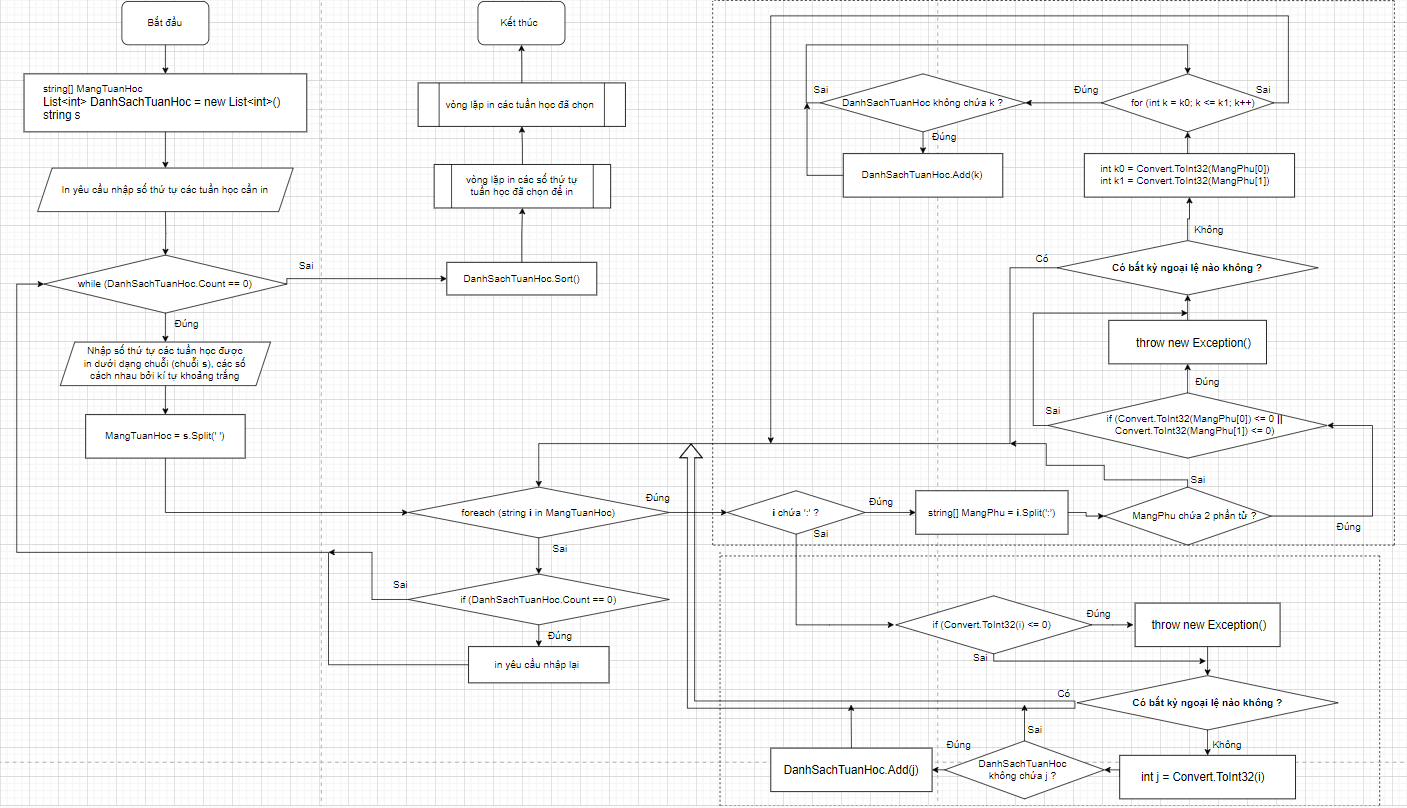




**Phương thức làm trống thời khóa biểu:** LamTrongTKB(). Phương thức có tác dụng làm trống một tuần học được chỉ định, không xóa thời khóa biểu ra khỏi danh sách thời khóa biểu mà chỉ làm trống danh sách học phần chứa trong nó, tức thời khóa biểu của một tuần học sau khi được làm trống sẽ chỉ giữ lại 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.

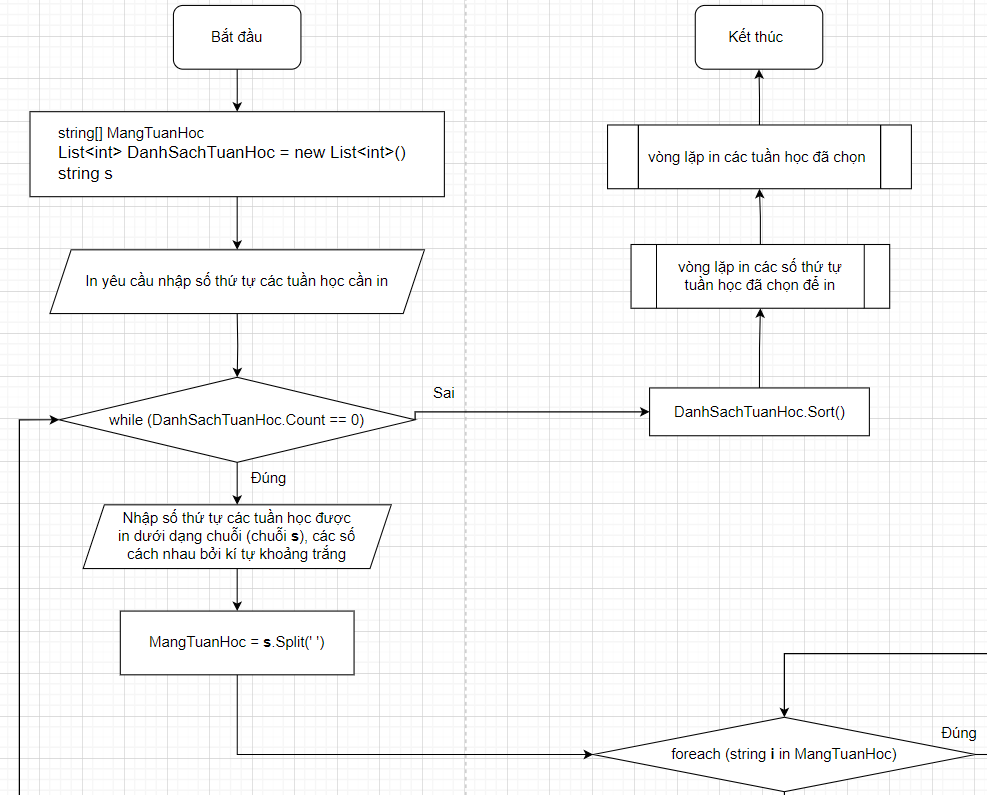


**Phương thức chỉnh sửa tuần học được chỉ định:** ChinhSuaTuanHoc(). Phương thức cho phép người dùng nhập số thứ tự tuần học cần chỉnh sửa rồi gọi đến phương thức ChinhSuaTKB() của tuần học (đối tượng thời khóa biểu) đó.



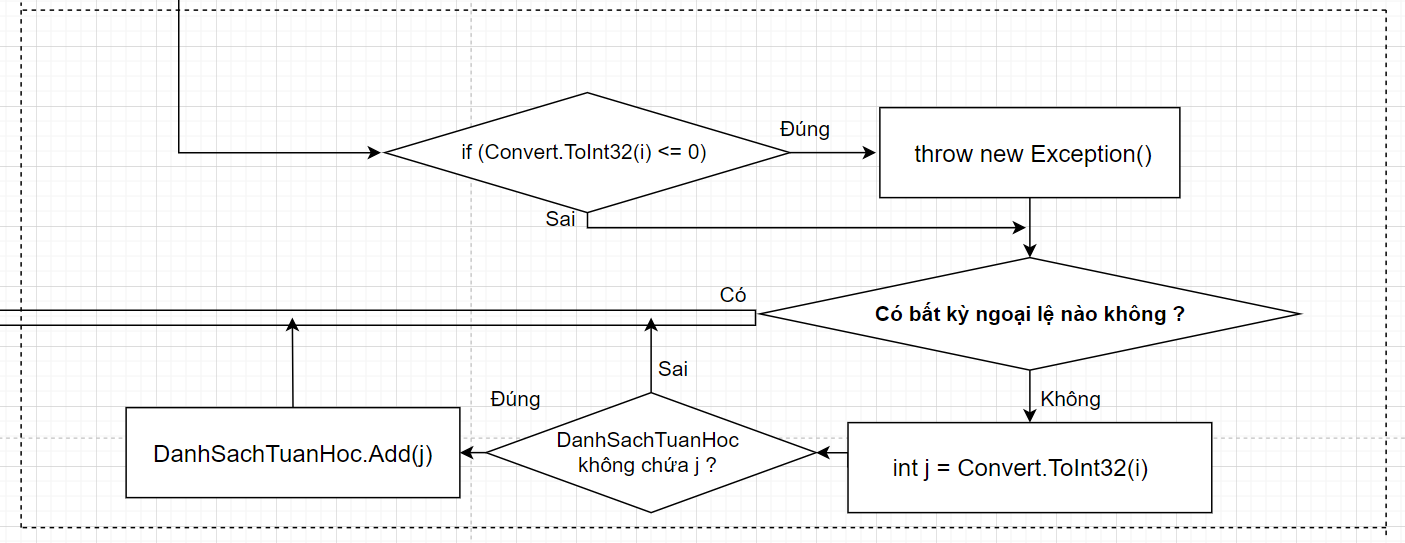
**Phương thức in thời khóa biểu theo tuần:** InTKBTheoTuan(). Phương thức nhận vào tham số là độ rộng ô (cột) của thời khóa biểu. Phương thức này chủ yếu để lập một danh sách các tuần học mà người dùng muốn in ra màn hình console. Sau mỗi lần in một thời khóa biểu, phương thức sẽ làm trống màn hình console 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 s được nhập, công việc tách các số nguyên ra từ chuỗi s sẽ tiếp diễn trong vòng lặp foreach.

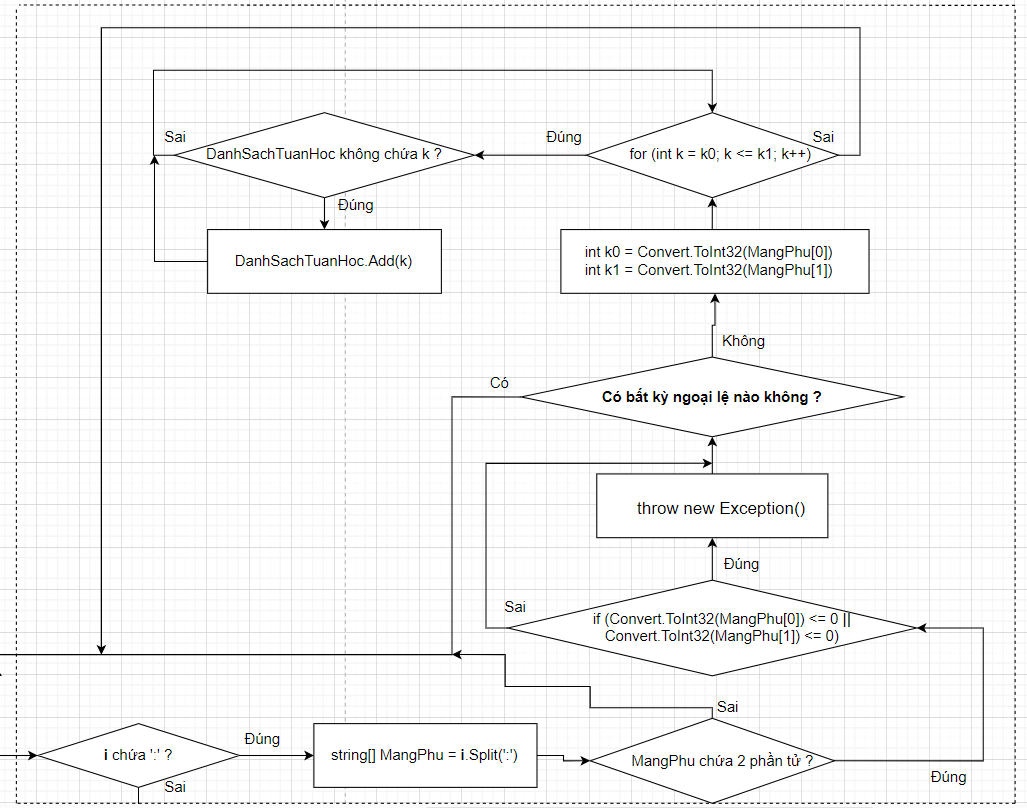


Để tạo thuận lợi cho việc sử dụng, phương thức cung cấp 2 cách thức nhập số tuần học cần in và ứng với mỗi cách nhập sẽ là một nhánh logic giúp lọc ra số nguyên từ chuỗi con, được tách ra từ chuỗi s (các chuỗi con giờ đây được lưu trong MangTuanHoc) với chuỗi i trong vòng lặp foreach duyệt lần lượt các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

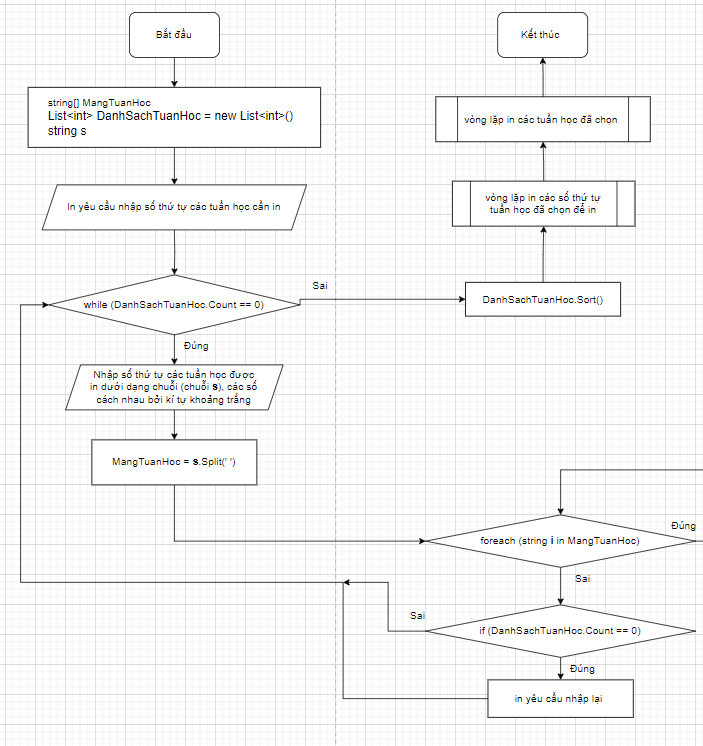
- Người dùng có thể nhập đơn lẻ từng số nguyên ứng với số thứ tự tuần mà mình muốn in. Điều kiện để vào nhánh logic này là chuỗi i không chứa kí tự ‘:’. Trong khối try/catch, các câu lệnh có thể gây ra ngoại lệ là lệnh chuyển chuỗi i thành số nguyên (nếu gây ra ngoại lệ tức chuỗi i không phải là số nguyên) 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.

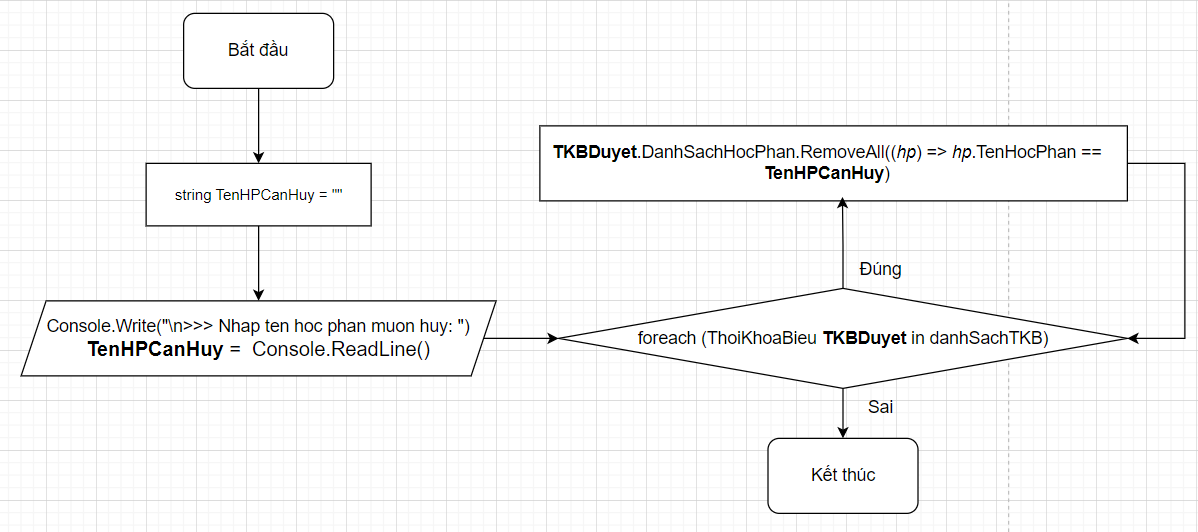


- Nhập <tuần bắt đầu>:<tuần kết thúc> để in tất cả các tuần từ <tuần bắt đầu> đến <tuần kết thúc>, ví dụ: nhập 1:5 tương ứng với 1 2 3 4 5. Điều kiện để vào được nhánh logic này là chuỗi i phải chứa kí tự ‘:’. Các ngoại lệ có thể bắt là: không thể chuyển một phần tử trong MangPhu thành số nguyên được (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.

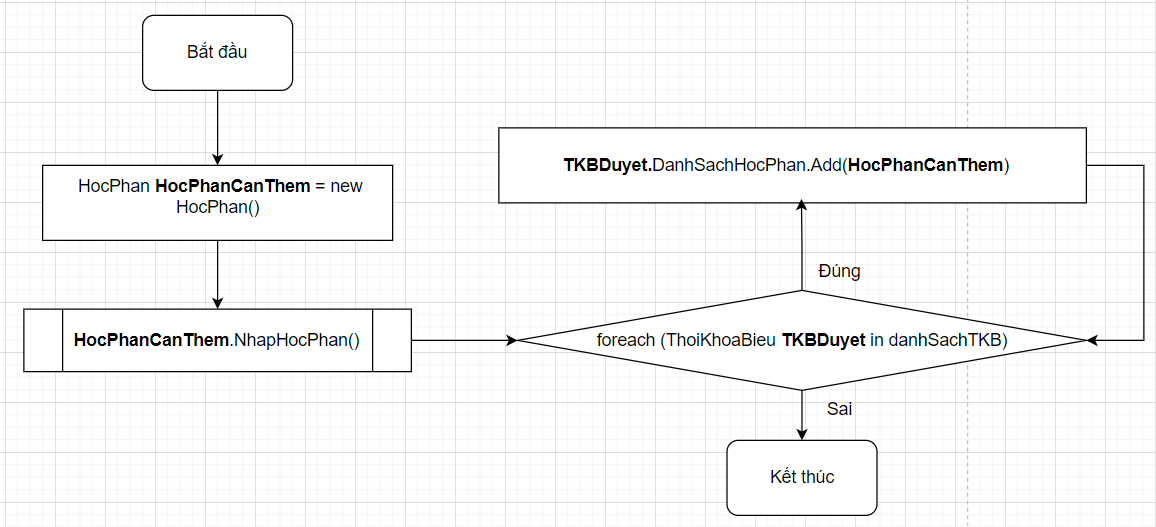


Các giá trị người dùng nhập vào bị trùng được xử lí để cho ra một danh sách các giá trị độc nhất, ví dụ: nhập 1 2 3 4 3:6 tương ứng với 1 2 3 4 5 6. Kết thúc vòng lặp foreach, nếu không có bất kì số nguyên nào được ghi nhận thì in ra yêu cầu nhập lại rồi tiếp tục vòng lặp while. Kết thúc vòng lặp while, 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.



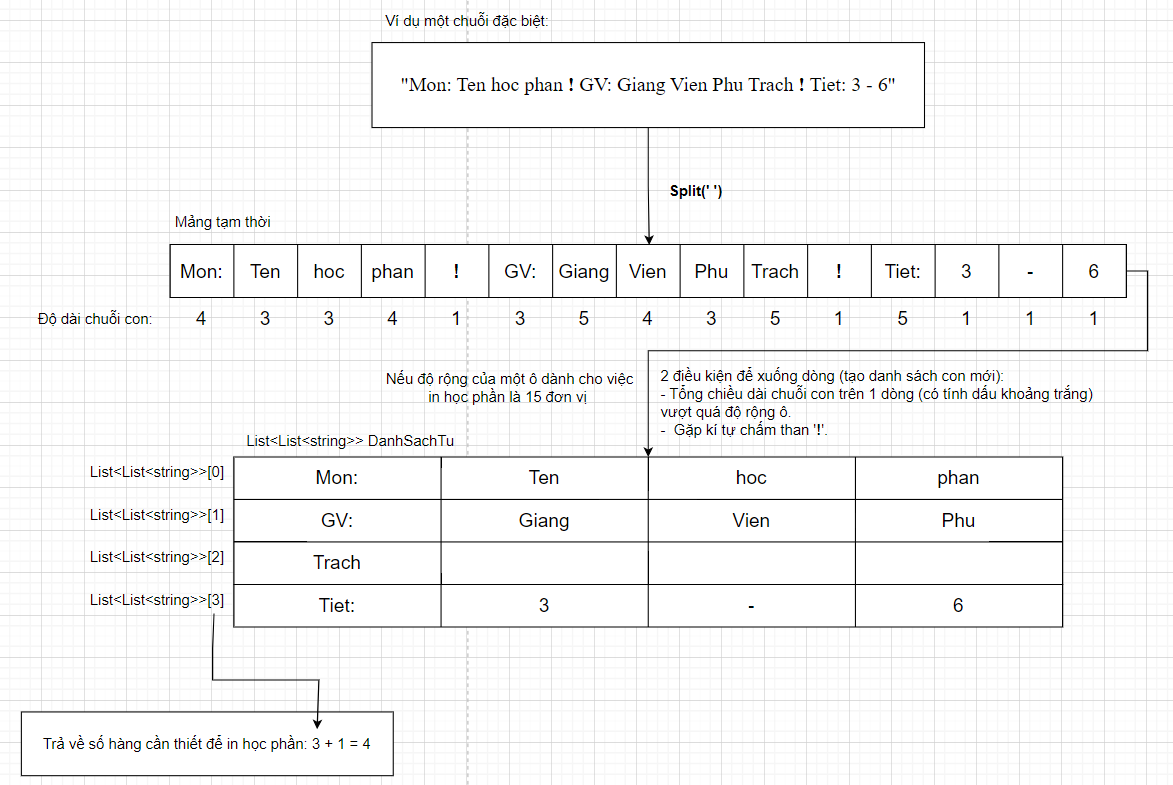


**Phương thức hủy học phần:** HuyHocPhan(). Phương thức cho phép xóa một học phần trên tất cả các thời khóa biểu được quản lý, tên học phần được nhập bởi người dùng.



**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.

**\*Lớp học phần (bổ sung vào các phương thức của lớp của Thu)**

**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).