

ĐẠI HỌC HUẾ  
Trường Đại học Kinh tế  
Khoa Hệ thống thông tin Kinh tế

**Cơ sở lập trình  
Bài tập**

*Hạn cuối nộp bài : 24h00 thứ ba, ngày 16 tháng 12 năm 2014*

*Nộp bằng cách gửi email về địa chỉ **lvman@hce.edu.vn***

*Hãy đọc kỹ **Quy định nộp báo cáo bài tập** trước khi làm bài*

## **Giới thiệu**

Trong bài tập này, các bạn sẽ vận dụng tất cả các khái niệm, kiến thức đã học trong môn Cơ sở lập trình và sử dụng ngôn ngữ C# để thiết kế, xây dựng một chương trình hoàn chỉnh cho phép quản lý đăng ký mượn phòng thực hành máy tính.

## **Quản lý đơn đăng ký mượn phòng thực hành máy tính**

Trường Đại học Kinh tế - Đại học Huế có nhu cầu mua một phần mềm để quản lý công tác đăng ký mượn phòng thực hành máy tính. Hiện tại, trường có 4 phòng máy tính là A5.1, A5.2, B5.2 và C4.1. Trường cho phép cán bộ, giáo viên sử dụng các phòng máy này để phục vụ công tác giảng dạy. Cán bộ, giáo viên muốn sử dụng phải đăng ký thông qua đơn đăng ký mượn phòng, ai đăng ký trước thì ưu tiên được mượn phòng thực hành. Mỗi đơn đăng ký mượn phòng bao gồm các thông tin sau: tên người mượn, phòng mượn, ngày mượn, các tiết mượn. Do đó, phần mềm này sẽ phải có những chức năng cơ bản sau :

- Cho phép đăng ký mượn phòng, sửa thông tin, huỷ mượn phòng
- Xem danh sách đăng ký mượn phòng theo ngày, theo người mượn
- In thống kê tần suất sử dụng phòng máy

## **Yêu cầu của bài tập**

Hãy sử dụng tất cả các kiến thức về ngôn ngữ lập trình C# đã được học để lập trình tạo ra một ứng dụng quản lý đăng ký mượn phòng thực hành máy tính. Chương trình cần có các chức năng sau :

1. Có cấu trúc dữ liệu để lưu trữ danh sách đơn mượn phòng thực hành máy tính
2. Thêm một đơn đăng ký mượn phòng thực hành máy tính
3. Mỗi đơn đăng ký sẽ được gắn với một mã số riêng biệt được chương trình tự động sinh ra
4. Các đơn đăng ký mượn phòng thực hành máy tính phải phân biệt qua phòng mượn, tiết mượn và ngày mượn. Các thông tin này không thể rỗng. Mỗi đơn đăng ký mượn phòng chỉ áp dụng cho duy nhất một phòng và một ngày.
5. Tìm kiếm một đơn đăng ký mượn phòng theo mã đơn mượn phòng
6. Sửa thông tin đơn đăng ký mượn phòng
7. Huỷ một đơn đăng ký mượn phòng
8. Xem danh sách đơn đăng ký mượn phòng theo thời gian (từ ngày đến ngày)
9. Xem danh sách đơn đăng ký mượn phòng theo tên người mượn phòng trong tháng nào đó
10. In thống kê tần suất sử dụng phòng máy cho tháng nào đó - Thống kê và in ra được các thông tin sau : số tiết sử dụng của mỗi phòng trong tháng và phần trăm sử dụng của mỗi phòng trong tháng.
11. (Tuỳ chọn - cộng điểm) Hỗ trợ đối số dòng lệnh
12. (Tuỳ chọn - cộng điểm) Lưu danh sách phòng thực hành máy tính và danh sách đơn mượn phòng vào file
13. (Tuỳ chọn - cộng điểm) Đọc danh sách phòng thực hành máy tính và danh sách đơn mượn phòng từ file
14. (Tuỳ chọn - cộng điểm) Dữ liệu đầu vào được kiểm tra và yêu cầu nhập lại khi dữ liệu không hợp lý

*Chú ý :*

1. Nên tổ chức chương trình dưới dạng hàm để dễ kiểm soát và sửa lỗi
2. Trong mỗi chức năng các bạn nên xây dựng thành các hàm chức năng nhỏ hơn. Như vậy, chương trình sẽ không bị rối và dễ gỡ lỗi. Đồng thời các chức năng khác có thể sử dụng lại các hàm đó.

3. Nên code từng phần một, sau mỗi phần phải kiểm tra kỹ càng đoạn code vừa được viết là hoạt động tốt rồi mới chuyển sang phần khác. *Ví dụ : sau khi code phần cho phép nhập dữ liệu vào, thì nên viết đoạn code in dữ liệu ra để kiểm soát việc nhập dữ liệu vào là đúng*
4. Khi kiểm tra thì nên kiểm tra với lượng ít số liệu để dễ kiểm soát chương trình. Sau khi chương trình đã đúng với số ít dữ liệu đó thì mới chạy chương trình với nhiều dữ liệu hơn

## Test

Các bạn dùng bộ test sau đây để kiểm tra chương trình và chụp ảnh màn hình kết quả để dán vào báo cáo.

1. Nhập dữ liệu các đơn mượn phòng sau :

Ngày	Tiết	Người mượn	Phòng	Ngày	Tiết	Người mượn	Phòng
13/10/2014	1-8	H.Hải	A5.2	13/10/2014	3-4	Sơn	A5.1
13/10/2014	7-8	H.Giang	A5.2	14/10/2014	1-4	Hoà DL	A5.1
14/10/2014	5-8	Hoà HT	A5.1	14/10/2014	1-4	Sơn	A5.2
14/10/2014	5-6	Phương	A5.2	14/10/2014	7-8	H.Giang	A5.2
15/10/2014	1-4	Hoà DL	A5.1	15/10/2014	5-8	Trang DL	A5.2
15/10/2014	1-4	H.Hải	A5.2	15/10/2014	5-8	H.Hải	A5.1
16/10/2014	5-8	Hoà HT	A5.2	16/10/2014	5-6	Mẫn	A5.1
16/10/2014	1-4	Sơn	A5.1	16/10/2014	1-4	Long	A5.2
16/10/2014	9-10	Tú	A5.1	17/10/2014	1-4	Hoà HT	A5.2
17/10/2014	5-8	Trang DL	A5.2	17/10/2014	5-8	H.Giang	A5.1
17/10/2014	3-4	Sơn	A5.1	18/10/2014	1-4	Long	A5.1
18/10/2014	5-6	Mẫn	A5.1	18/10/2014	5-6	H.Giang	A5.2
18/10/2014	7-8	H.Giang	A5.1	21/10/2014	5-6	Phương	A5.2
22/10/2014	3-4	Nhật	A5.1	23/10/2014	5-6	Phương	A5.2
24/10/2014	3-4	Nhật	A5.1	25/10/2014	3-4	Phòng Đào tạo	A5.1
27/10/2014	7-8	Tú	A5.1	28/10/2014	5-6	Phương	A5.1
29/10/2014	7-8	Tú	A5.1	29/10/2014	3-4	Nhật	A5.1
29/10/2014	5-6	Phương	A5.1	30/10/2014	5-6	Phương	A5.1
31/10/2014	3-4	Nhật	A5.1	03/11/2014	5-6	Phương	A5.1
03/11/2014	7-8	Tú	A5.2	03/11/2014	1-2	Mẫn	A5.1
03/11/2014	5-6	Mẫn	A5.2	04/11/2014	5-6	Phương	A5.1
05/11/2014	7-8	Tú	A5.2	05/11/2014	3-4	Nhật	A5.1
05/11/2014	5-6	Phương	A5.1	05/11/2014	1-2	Mẫn	A5.1

05/11/2014	5-6	Mẫn	A5.2	06/11/2014	5-6	Phương	A5.1
07/11/2014	3-4	Nhật	A5.1	07/11/2014	1-2	Mẫn	A5.1
07/11/2014	4-5	Mẫn	A5.1	10/11/2014	7-8	Tú	A5.2
10/11/2014	5-6	Phương	A5.1	10/11/2014	1-2	Mẫn	A5.1
10/11/2014	5-6	Mẫn	A5.2	11/11/2014	5-6	Phương	A5.1
12/11/2014	7-8	Tú	A5.2	12/11/2014	5-6	Phương	A5.1
12/11/2014	3-4	Nhật	A5.1	12/11/2014	1-2	Mẫn	A5.1
12/11/2014	5-6	Mẫn	A5.2	13/11/2014	5-6	Phương	A5.1

- Sửa tiết mượn 7-8 của người mượn là H.Giang ngày 14/10/2014 thành tiết 5-8
- Sửa ngày mượn của người mượn là Phương ngày 13/11/2014 thành ngày mới là 31/11/2014
- Xoá đơn mượn ngày 18/10/2014 của người mượn là Mẫn
- Xem danh sách đơn đăng ký mượn phòng từ ngày 01/11/2014 đến 13/11/2014
- Xem danh sách đơn đăng ký mượn phòng của người mượn là Tú trong tháng 11
- In thống kê tần suất sử dụng phòng máy cho tháng 10

## Hướng dẫn lập trình

Những thuật toán, cấu trúc dữ liệu và cách xử lý được đưa ra trong các phần sau đây chỉ là **đề nghị**. Các bạn có thể đưa ra thuật toán, cấu trúc dữ liệu và cách xử lý riêng và hãy chỉ ra trong báo cáo của bạn. Thuật toán, cấu trúc dữ liệu và cách xử lý của bạn là tốt hơn sẽ được đánh giá cao hơn và ngược lại, điểm của bạn sẽ thấp nếu chương trình của bạn hỗ trợ ít chức năng hơn hoặc cách xử lý ít phức tạp hơn.

Các phần dưới đây sẽ được trình bày tuần tự theo một quy trình mà nếu bạn làm theo từng bước thì có thể hoàn thành tốt chương trình. Các bước sau sẽ là phát triển tiếp của bước trước, nên không làm bước trước thì sẽ khó hoàn thành được bước sau.

## Cấu trúc dữ liệu để lưu trữ các khoản thu chi

Cấu trúc dữ liệu là một trong hai thành phần cốt lõi để tạo nên một chương trình máy tính và nó có tính quyết định đến việc lựa chọn thuật toán, cách xử lý trong chương trình. Do đó, khi phân tích thiết kế chương trình, ta nên lựa chọn cấu trúc dữ liệu tốt để việc xử lý sau này sẽ đơn giản và thuận tiện hơn.

Đầu tiên, chúng ta cần phân tích xem chương trình cần lưu những dữ liệu gì. Theo mô tả của bài toán, chúng ta cần lưu trữ thông tin các đơn đăng ký mượn phòng. Mỗi đơn mượn

phòng bao gồm các thông tin : mã đơn, tên người mượn, phòng mượn, ngày mượn, các tiết mượn. Như vậy, cấu trúc dữ liệu để lưu danh sách các đơn mượn phòng trên phải thoả mãn hai đặc điểm :

1. cho phép lưu trữ một loạt các số liệu
2. mỗi thành phần trong loạt số liệu đó bao gồm :
  - một số cho mã đơn mượn phòng
  - một chuỗi ký tự để lưu tên người mượn
  - một chuỗi ký tự để lưu tên phòng muốn mượn
  - ba số nguyên để lưu ngày, tháng và năm muốn mượn
  - danh sách các số nguyên để đại diện cho các tiết muốn mượn, hoặc chỉ cần hai số nguyên để đại diện cho tiết bắt đầu mượn và tiết kết thúc mượn.

Với đặc điểm đầu tiên, bạn có thể dùng mảng, **List**, **Stack** hoặc **Queue** để lưu trữ một loạt các số liệu. Mảng rõ ràng là đơn giản nhất, nhưng mảng lại không có tính linh động khi chúng ta cần mở rộng kích thước của mảng. **List** thì linh động và mạnh mẽ hơn. **Stack** và **Queue** thì có thể dùng nhưng hai cấu trúc này phù hợp cho các bài toán khác hơn là để lưu một danh sách dữ liệu. Dù là mảng kém linh động như vậy nhưng chúng ta vẫn có thể dùng. Nếu sử dụng mảng, bạn sẽ cấp phát số thành phần cố định ban đầu cho mảng và khi đó, bạn chỉ có thể lưu trữ nhiều nhất bằng số lượng cố định đó.

Với đặc điểm thứ hai, chỉ một mảng một chiều hoặc một **List** sẽ không thể nào phù hợp. Do đó, ta có thể sử dụng :

- *Cách 1* : nhiều mảng một chiều hoặc nhiều **List**. Mỗi mảng/**List** có kiểu dữ liệu phù hợp để lưu trữ cho *một* thông tin của đơn mượn phòng. Trong đó, thành phần thứ *i* trong mỗi mảng/**List** đó tương ứng với đơn mượn phòng thứ *i*.

*Ví dụ* (dùng nhiều mảng một chiều): Chúng ta sẽ dùng tám mảng một chiều : một mảng kiểu **int** để lưu mã đơn mượn phòng, hai mảng kiểu **string** để lưu tên người mượn và tên phòng mượn, ba mảng kiểu **int** để lưu ngày, tháng, năm, và hai mảng **int** để lưu hai số đại diện cho tiết bắt đầu mượn và tiết kết thúc mượn.

- *Cách 2* : hai mảng một chiều hoặc hai **List** mà mỗi phần tử là một kiểu cấu trúc (**struct**). Cấu trúc nên được định nghĩa tốt để lưu trữ các thông tin của phòng thực hành và đơn đăng ký mượn phòng.

Ví dụ, chúng ta có thể định nghĩa hai cấu trúc như sau :

```
1 struct DonMuonPhong
2 {
```

```

3      public int ma; // ma đơn mượn
4      public string tenNgMuon; // ten người mượn
5      public string phongMuon; // phong mượn
6      public int ngay;
7      public int thang;
8      public int nam;
9      public int tietBDMuon; // tiet bắt đầu mượn
10     public int tietKTMuon; // tiet kết thúc mượn
11 };

```

## Nhập dữ liệu đơn đăng ký mượn phòng thực hành máy tính

Trong việc nhập dữ liệu, bạn có thể áp dụng theo hai kiểu sau :

### 1. *Biết trước* số lượng đơn sẽ được nhập

Tức là, bạn sẽ yêu cầu người sử dụng nhập vào số lượng đơn muốn nhập. Sau đó, dựa vào giá trị số lượng đó, bạn có thể dùng một vòng lặp để cho phép người sử dụng nhập vào dữ liệu từng đơn.

### 2. *Không biết trước* số lượng đơn sẽ được nhập

Tức là, số lượng đơn không được biết trước (chương trình không hỏi giá trị này), nên sau khi người sử dụng nhập xong dữ liệu của một đơn mượn phòng nào đó thì chương trình sẽ hỏi xem người sử dụng có muốn nhập tiếp hay không. Tùy vào câu trả lời của người sử dụng mà chương trình cho phép người sử dụng tiếp tục nhập hoặc thoát ra khỏi tiến trình nhập này.

Hoặc một cách khác, các bạn có thể dừng việc nhập khi người sử dụng nhập vào ký tự kết thúc file (end of file - Ctrl + Z).

Xem ví dụ vòng lặp `while` ở slide số 15, chủ đề 5 (Các cấu trúc điều khiển 2).

Ví dụ - thuật toán đoạn chương trình nhập dữ liệu theo dạng không biết trước số lượng đơn đăng ký sẽ được nhập, cấu trúc dữ liệu theo cách 2 :

**Đầu vào** : biến kiểu `List < DonMuonPhong >` đã được khởi tạo hoặc đã có dữ liệu, gọi là `listDonMuonPhong`

**Đầu ra** : `listDonMuonPhong` đã thêm vào các đơn mượn mới

Khai báo một biến kiểu `DonMuonPhong`, gọi là `donMuon`

Cho người sử dụng nhập vào thông tin của đơn đăng ký mượn phòng mới

Nếu ngày mượn, tiết mượn, phòng mượn trùng với những đơn đăng ký đã có thì

Thông báo trùng dữ liệu với đơn mượn phòng đã có  
Chuyển đến phần Hỏi người sử dụng có muốn nhập tiếp đơn mới  
hay không

Ngược lại

Sinh tự động mã đơn mới không trùng với các mã đơn đã có  
Gán mã đơn mới đó cho biến thành phần *ma*  
Tách ngày mượn gán vào ba biến thành phần *ngay*, *thang*, *nam*  
Tách tiết mượn để gán vào danh sách *tietMuon*

Phòng mượn gán vào biến thành phần *phongMuon*

Tên người mượn nếu không rỗng thì gán vào biến thành phần  
*tenNgMuon*

Gán biến *donMuon* vào *listDonMuonPhong*

Hỏi người sử dụng có muốn nhập tiếp đơn mới hay không.

Nếu có, thì trở lại bước đầu tiên

Ngược lại, thoát ra khỏi thuật toán này

### Cách sinh mã đơn tự động

Bạn có thể sinh mã tự động theo cách sau : mỗi lần thêm một đơn mượn mới, bạn tìm giá trị lớn nhất trong các mã đơn đã có, rồi tăng thêm một đơn vị. Giá trị mới này là mã đơn mới.

Nếu bạn muốn tiết kiệm mã đơn, bạn có thể lấy các mã đơn của các đơn đã bị xoá để sử dụng.

Cách tạo danh sách đơn mượn phòng như thuật toán trên sẽ cho ta một danh sách được sắp theo thứ tự nhập vào (Vì với mỗi đơn mới ta chỉ thêm vào cuối (sử dụng hàm **Add** của **List**) danh sách hiện có). Tuy nhiên, để tiện cho các thao tác lọc dữ liệu, cũng như thống kê, bạn có thể cho danh sách này sắp xếp tăng dần theo ngày mượn, tiết mượn. Có hai cách để tạo ra danh sách được sắp xếp như vậy :

1. Sắp xếp danh sách theo thứ tự mong muốn
2. Khi có một đơn đăng ký mới, bạn không *thêm* vào mà bạn *chèn* nó vào vị trí thích hợp với thứ tự sắp xếp mong muốn

Bạn tham khảo các câu trong phần vòng lặp **for** của bài thực hành 3 để biết cách xác định một ngày là nhỏ hơn hay lớn hơn một ngày khác (sử dụng ngày tuyệt đối). Nếu bạn dùng cấu trúc kiểu **List** thì bạn dùng hàm **Insert** để chèn đơn mượn phòng mới vào danh sách. Còn bạn dùng mảng thì tham khảo câu 4, phần Mảng, bài thực hành 5.

*Những điều trình bày trên đây cũng phù hợp với cấu trúc dữ liệu mảng.*

*Sau đoạn chương trình nhập dữ liệu, bạn nên có đoạn chương trình cho phép in ra dữ liệu đã được nhập vào để người sử dụng kiểm tra lại dữ liệu mà họ đã nhập.*

## Tìm kiếm một đơn mượn phòng thực hành theo mã đơn

Với cấu trúc dữ liệu theo cách 2, thuật toán tìm kiếm theo mã đơn mượn phòng như sau :

*Đầu vào* : mã phòng cần tìm

*Đầu ra* :  $i$  - vị trí của đơn đó trong danh sách

Khởi gán  $i = 0$

Nếu  $i >$  số lượng phần tử trong danh sách

Thông báo là không có hồ sơ cần tìm

Trả ra giá trị  $i = -1$

Ngược lại,

Nếu phần tử thứ  $i$  trong danh sách có biến thành phần  $ma$  bằng với mã đơn cần tìm

Kết thúc, giá trị của  $i$  là vị trí đơn cần tìm trong danh sách

Ngược lại, tăng giá trị của  $i$  lên một đơn vị

Quay lại bước 2 trong thuật toán này

## Sửa thông tin một đơn mượn phòng

Để sửa thông tin một đơn mượn phòng thực hành nào đó, ta có thể sử dụng thuật toán sau (cho cấu trúc dữ liệu theo cách 2) :

Yêu cầu người sử dụng nhập vào mã đơn của đơn mượn phòng muốn sửa

Sử dụng thuật toán *tìm kiếm đơn mượn phòng theo mã đơn* để lấy

vị trí của đơn mượn phòng trong danh sách

Nếu kết quả của thuật toán tìm kiếm là  $-1$  ( $i == -1$ ),

Thông báo không có đơn với mã muốn tìm, yêu cầu nhập lại nếu muốn

Nếu có nhập lại thì quay lại bước 1 trong thuật toán này

Nếu không thì thoát khỏi thuật toán

Ngược lại,

In thông tin hiện có của đơn mượn phòng ra màn hình

Yêu cầu người sử dụng nhập dữ liệu mới

Gán dữ liệu mới vào

## Xoá một đơn mượn phòng

Tương tự thuật toán sửa thông tin một đơn mượn phòng, thuật toán xoá một đơn mượn phòng như sau :

Yêu cầu người sử dụng nhập vào mã đơn của đơn mượn phòng muốn xoá

Sử dụng thuật toán *tìm kiếm đơn mượn phòng theo mã đơn* để lấy vị trí của đơn mượn phòng trong danh sách

Nếu kết quả của thuật toán tìm kiếm là  $-1$  ( $i == -1$ ),



Thông báo không có đơn mượn phòng cần tìm, yêu cầu nhập lại nếu muốn  
Nếu có nhập lại thì quay lại bước 1 trong thuật toán này  
Nếu không thì thoát khỏi thuật toán

Ngược lại,

In thông tin hiện có của đơn mượn phòng ra màn hình  
Xác nhận yêu cầu xoá của người sử dụng  
Nếu người sử dụng xác nhận xoá thì thực hiện xoá  
Nếu không, thì thoát khỏi thuật toán

## In danh sách

Bài tập yêu cầu hai chức năng in : danh sách đơn đăng ký mượn phòng theo thời gian và danh sách đơn đăng ký mượn phòng theo người mượn trong tháng nào đó. Tuy nhiên, bạn có thể sẽ cần viết cả chức năng in danh sách tất cả các đơn mượn phòng.

Với chức năng in danh sách đơn mượn phòng theo thời gian, bạn cho phép người dùng nhập vào khoảng thời gian cần in danh sách (ngày bắt đầu, ngày kết thúc). Sau đó, bạn duyệt qua danh sách các đơn mượn phòng. Nếu đơn nào có ngày mượn nằm trong khoảng thời gian trên thì bạn in thông tin đơn đó ra. Để kiểm tra một ngày là ngày trước của một ngày nào đó, bạn có thể đổi ngày sang dạng ngày tuyệt đối, rồi so sánh trên ngày tuyệt đối. Xem câu 2, phần lệnh lặp for, bài thực hành 3 để biết cách đổi thành ngày tuyệt đối.

## Thống kê tần suất sử dụng

Bài tập yêu cầu bạn thống kê được hai giá trị sau :

- số tiết sử dụng của mỗi phòng trong tháng
- phần trăm sử dụng của mỗi phòng trong tháng

Bài tập đã cố định chỉ có 4 phòng có thể sử dụng, nên bạn có thể tạo ra bốn biến đếm, hoặc một mảng bốn thành phần kiểu `int` để chứa kết quả đếm số tiết sử dụng.

Để tính phần trăm sử dụng, bạn cần biết tổng số tiết mỗi tháng. Bạn đã biết mỗi ngày sẽ có 10 tiết. Tuy nhiên, để biết mỗi tháng có bao nhiêu ngày có thể học thì bạn phải biết hai giá trị :

1. số ngày của tháng  
*Xem câu 2, phần lệnh lặp for, bài thực hành 3.*

2. số ngày chủ nhật trong tháng đó.

Để tính số ngày chủ nhật trong tháng, bạn sử dụng công thức sau :

$$(b - a - (b\%7 + 1) + 8)/7 \quad (1)$$

Trong đó, a là ngày tuyệt đối của ngày đầu tháng, b là ngày tuyệt đối của ngày cuối tháng.

## Menu chương trình

Trong các chương trình có nhiều chức năng, thì người lập trình thường hỗ trợ một menu chương trình. Người sử dụng chỉ việc chọn một số hoặc một chữ cái để thực hiện một chức năng nào đó. Khi một chức năng nào đó đã thực hiện xong thì menu đó lại được in ra lại và chương trình hỏi người sử dụng tiếp tục muốn thực hiện chức năng nào. Chương trình chỉ kết thúc/dừng khi người sử dụng chọn số hoặc chữ cái tương đương chức năng dừng chương trình.

Ví dụ menu chương trình có thể như sau khi chương trình chạy :

CHUONG TRINH QUAN LY DON DANG KY MUON PHONG MAY TINH

- 1 - Them don dang ky moi
- 2 - Sua don dang ky muon phong
- 3 - Xoa don dang ky muon phong
- 4 - Danh sach tat ca cac don muon phong
- 5 - Danh sach cac don muon phong theo thoi gian
- 6 - Danh sach cac don muon phong theo nguoi muon va thang
- 7 - Thong ke tan suat su dung
- 8 - Doc du lieu tu file
- 9 - Ghi du lieu ra file
- 10 - Thoat

Ban chon so may : \_

Để hỗ trợ tính năng này, người ta thường dùng vòng lặp **while**. Và khi người sử dụng nhập số là 10 (theo ví dụ trên) thì mới ngắt vòng lặp **while**. Xem slide 23, chủ đề 5 - Các cấu trúc điều khiển (phần 2) để biết cách tạo một menu như trên.

## Xử lý đối số dòng lệnh

Xem slide 37-40, chủ đề 6.

Với chương trình này, các bạn có thể hỗ trợ 4 dạng đối số dòng lệnh sau :

```
1 qldmp.exe
2 qldmp.exe <num>
3 qldmp.exe <path>
4 qldmp.exe -h
```

Với `qltmp.exe` là tên chương trình, `qltmp` là viết tắt của Quản lý đơn mượn phòng.

Dạng thứ nhất chỉ gọi chương trình mà không đưa vào đối số nào cả. Như vậy, chương trình sẽ in ra menu chương trình và chờ người sử dụng lựa chọn chức năng tiếp theo để thực hiện.

Dạng thứ hai gọi chương trình kèm với một số (`<num>`). Chương trình sẽ thực hiện chức năng nhập `<num>` đơn mượn phòng mới. Đây là cách nhập dữ liệu khi biết trước số lượng đơn mượn phòng được nhập.

Dạng thứ ba là gọi chương trình kèm theo đường dẫn (`<path>`) đến tập tin văn bản chứa dữ liệu. (Đọc thêm phần Nhập dữ liệu từ file văn bản bên dưới)

Dạng thứ tư kèm theo đối số `-h` để yêu cầu in ra hướng dẫn sử dụng của chương trình.

Thuật toán của phần xử lý đối số dòng lệnh như sau :

Kiểm tra số lượng đối số dòng lệnh

Nếu lớn hơn 3 thì thông báo lỗi và in ra hướng dẫn sử dụng chương trình

Nếu không có đối số nào thì chạy chương trình bình thường

Nếu có 1 đối số

Đối số đó là chuỗi `-h` (so sánh chuỗi ký tự trong đối số với chuỗi `"-h"`) thì in ra hướng dẫn sử dụng

Nếu thuộc dạng 2 (thử đổi chuỗi thành số) thì cho nhập dữ liệu theo dạng biết trước số lượng khoản thu chi sẽ được nhập

Nếu thuộc dạng 3 thì chuyển đến nhập dữ liệu từ file văn bản

**Chú ý :** Nếu bạn không làm yêu cầu 12 và 13 thì có thể bỏ qua việc xử lý dạng thứ 3 của đối số dòng lệnh

## Nhập dữ liệu từ file văn bản

Nội dung của tập tin chứa dữ liệu các đơn mượn phòng có thể được quy định như sau :

- Mỗi dòng chứa dữ liệu cho một đơn mượn phòng, bao gồm 5 thành phần phân tách với nhau bằng dấu phẩy (,)
  - thành phần thứ nhất là mã đơn
  - thành phần thứ hai là ngày mượn, theo dạng `dd/mm/yyyy`
  - thành phần thứ ba là tiết bắt đầu mượn và tiết kết thúc mượn, được phân tách với nhau bằng dấu gạch ngang (–)
  - thành phần thứ tư là tên người mượn
  - thành phần thứ năm là phòng mượn

Bởi vì tên người mượn có thể có khoảng trống nên phải dùng dấu phẩy để phân tách giữa các thành phần của mỗi dòng.

Ví dụ sau đây là một phần nội dung của tập tin `test.txt` :

```

1 1,13/10/2014,1-8,H.Hai,A5.2
2 2,13/10/2014,3-4,Son,A5.1
3 3,13/10/2014,7-8,H.Giang,A5.2
4 4,14/10/2014,1-4,Hoa DL,A5.1
5 5,14/10/2014,5-8,Hoa HT,A5.1
6 6,14/10/2014,1-4,Son,A5.2
7 7,14/10/2014,5-6,Phuong,A5.2

```

Thuật toán đọc dữ liệu từ file cho cấu trúc dữ liệu cách 2 như sau :

*Đầu vào* : biến kiểu *List < DonMuonPhong >* đã được khởi tạo hoặc đã có dữ liệu, gọi là *listDonMP*

*Đầu ra* : *listDonMP* đã thêm vào các đơn mượn phòng mới được đọc từ file

Đọc dòng đầu tiên

Nếu đọc được dữ liệu (1 dòng văn bản)

Khởi tạo một biến kiểu *DonMuongPhong*, gọi là *don*

Dùng hàm Split tách các thành phần trong chuỗi đọc được gán vào biến *listTP*

Nếu *listTP* không đủ 5 thành phần

Thông báo file dữ liệu bị lỗi, thoát khỏi thuật toán

Ngược lại,

Đổi thành phần đầu tiên trong *listTP* thành số và gán vào biến thành phần *ma* của biến *don*

Dùng hàm Split để tách các thành phần trong thành phần thứ hai của *listTP*, rồi gán vào biến thành phần *ngayMuon* của biến *don*

Dùng hàm Split để tách các thành phần trong thành phần thứ ba của *listTP*, rồi gán vào hai biến thành phần *tietBDMMuon* và *tietKTMMuon* của biến *don*

Gán biến thành phần thứ tư và năm của *listTP* vào hai biến thành phần *tenNgMuon* và *phongMuon* của biến *don*

Đọc dòng tiếp theo trong file, quay lại bước 2 của thuật toán này

## Xuất dữ liệu ra file văn bản

Thuật toán xuất dữ liệu ra file văn bản như sau :

*Đầu vào* : biến kiểu *List < DonMuonPhong >* đã được khởi tạo hoặc đã có dữ liệu, gọi là *listDonMP*

*Đầu ra* : file văn bản chứa dữ liệu các đơn mượn phòng

Duyệt qua từng đơn mượn phòng trong *listDonMP*, với mỗi đơn mượn phòng

Sử dụng cú pháp token giữ chỗ để tạo ra một chuỗi ký tự chứa 5 thành phần (mã đơn, ngày mượn, các tiết mượn, người mượn và phòng mượn) theo cú pháp của file dữ liệu đã quy định, các thành phần phân tách nhau bởi dấu phẩy.

Ghi chuỗi ký tự được tạo ra vào file.

*Chúc các em làm bài tốt !*