

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 : 12h00 thứ hai, ngày 25 tháng 11 năm 2013

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

*Hãy đọc **Quy định nộp báo cáo bài tập***

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 hồ sơ tuyển sinh của một trường đại học.

Quản lý hồ sơ xét tuyển

Trường Đại học Dân Lập Hiện Đại hàng năm có nhu cầu tuyển sinh vài trăm sinh viên với hình thức xét tuyển. Đầu tháng sáu hàng năm, nhà trường bắt đầu nhận hồ sơ ứng tuyển của học sinh vừa kết thúc lớp 12. Hồ sơ bao gồm những thông tin sau : Họ tên học sinh, ngày tháng năm sinh, quê quán và 5 cột điểm lớp 12 của 5 môn toán, lý, hoá, văn và anh văn. Nhà trường kết thúc nhận hồ sơ khi đã nhận đủ 1000 hồ sơ hoặc vào cuối tháng sáu. Đến đầu tháng bảy, nhà trường bắt đầu thực hiện xét tuyển. Nhà trường sẽ tính điểm trung bình cộng của 5 môn trên. Những em nào có điểm trung bình cao hơn mức điểm sàn thì trúng tuyển.

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 hồ sơ xét tuyển. Chương trình này cho phép người sử dụng nhập vào thông tin các hồ sơ ứng tuyển, sửa đổi thông tin các hồ sơ đã nhập, xoá một hồ sơ nào đó, in thông tin tất cả các hồ sơ, in danh sách trúng tuyển.

Các yêu cầu cụ thể của chương trình như sau :

1. Có cấu trúc dữ liệu để lưu trữ các hồ sơ ứng tuyển

2. Cho phép nhập thông tin các hồ sơ ứng tuyển
3. Cho phép sửa thông tin một hồ sơ nào đó
4. Cho phép xoá một hồ sơ
5. Hỗ trợ tìm kiếm hồ sơ theo tên học sinh
6. Tính điểm trung bình
7. In thông tin tất cả các hồ sơ
8. In danh sách các hồ sơ trúng tuyển
9. Các hồ sơ được phân biệt thông qua họ tên ứng viên và họ tên không cho phép rỗng
10. (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. 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 code đoạn code in dữ liệu ra để kiểm soát việc nhập dữ liệu vào là đúng*
3. 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 2 bộ test sau đây để kiểm tra chương trình và chụp ảnh màn hình kết quả để dán vào báo cáo.

Data 1

1. Nhập dữ liệu cho 10 hồ sơ sau :

Họ tên	Ngày sinh	Quê quán	Toán	Lý	Hoá	Văn	Anh Văn
A	01/01/1995	Hue	10	9	8	7	6
B	02/02/1994	Da nang	9	5	7	4	8
C	03/03/1996	Hanoi	9	9	8	8	9

D	04/04/1997	Ho Chi Minh	10	9	7	4	7
E	05/05/1995	Quang Binh	7	6	4	8	9
F	06/06/1996	Quang Tri	8	6	8	7	6
G	07/07/1997	Hue	7	8	9	10	7
H	08/08/1997	Quang Nam	8	8	8	9	5
K	09/09/1996	Quang Ngai	9	7	8	9	10
L	10/10/1995	Da nang	10	8	7	6	10

2. Sửa ngày sinh của học sinh có tên C thành ngày 29/02/1996
3. Sửa điểm môn Hoá của học sinh có tên E thành 6 điểm
4. Xoá hồ sơ của học sinh G
5. In danh sách trúng tuyển với điểm sàn là 8

Data 2

1. Nhập dữ liệu cho 10 hồ sơ sau :

Họ tên	Ngày sinh	Quê quán	Toán	Lý	Hoá	Văn	Anh Văn
A	30/01/1995	Hue	10	9.5	8.3	7.1	6.7
B	30/02/1996	Da nang	9	5.5	7.2	4.1	8.3
C	30/03/1996	Hanoi	9	9	8	8	9
D	30/04/1997	Ho Chi Minh	9.5	9	7.3	4.5	7.9
E	30/05/1995	Quang Binh	7.3	6.5	4.7	8.3	9
F	30/06/1996	Quang Tri	8.2	6.1	8	7.4	6
G	30/07/1997	Hue	7.2	8.2	9.2	10	7
H	30/08/1997	Quang Nam	8.4	8.3	8.9	9	5
K	30/09/1996	Quang Ngai	9	7.1	8.2	9	10
L	30/10/1995	Da nang	10	8	7.3	6.1	10

2. Sửa ngày sinh của học sinh có tên B thành ngày 28/02/1996
3. Sửa điểm môn Văn của học sinh có tên G thành 6.5 điểm
4. Xoá hồ sơ của học sinh E
5. In danh sách trúng tuyển với điểm sàn là 8.3

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.

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.

Các phần dưới đây sẽ được trình bày tuần tự theo một quy trình mà nếu các 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 hồ sơ ứng tuyển

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 các hồ sơ ứng tuyển. Mỗi hồ sơ bao gồm các thông tin như : họ tên, ngày sinh, quê quán và điểm của 5 môn. Như vậy, cấu trúc dữ liệu phải thỏa 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 chuỗi ký tự cho họ tên
 - ba số cho ngày sinh
 - chuỗi ký tự cho quê quán
 - năm số cho điểm của 5 môn

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* : 10 mảng một chiều hoặc 10 List. Trong đó, thành phần thứ i trong mỗi mảng/List tương ứng với hồ sơ ứng tuyển thứ i
- *Cách 2* : ba mảng hai chiều :
 - một mảng hai chiều kiểu string với cố định 2 hàng. Hàng thứ nhất để lưu họ tên, hàng thứ hai lưu quê quán
 - một mảng hai chiều kiểu int với cố định 3 hàng. Hàng thứ nhất để lưu ngày sinh, hàng thứ hai lưu tháng sinh và hàng thứ ba để lưu năm sinh
 - một mảng hai chiều kiểu float với cố định 5 hàng. Hàng thứ nhất để lưu các điểm toán, hàng thứ hai lưu điểm lý, hàng thứ ba lưu điểm hoá, hàng thứ tư lưu điểm văn và hàng cuối cùng lưu điểm anh văn.
- *Cách 3* : một mảng một chiều hoặc List mà mỗi thành phần là một kiểu cấu trúc (struct). Mỗi cấu trúc nên được định nghĩa tốt để lưu trữ các thông tin của hồ sơ ứng tuyển. Ví dụ, chúng ta có thể định nghĩa một cấu trúc như sau :

```
1 struct HoSo
2 {
3     public string hoten; // ho va ten
4     public int ngay; // ngay sinh
5     public int thang; // thang sinh
6     public string quequan; // que quan
7     public float toan; // diem toan
8     public float ly; // diem ly
9     public float hoa; // diem hoa
10    public float van; // diem van
11    public float anhvan; // diem anh van
12 };
```

Nhập dữ liệu

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

1. Biết trước số lượng hồ sơ được nhập

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

2. Không biết trước số lượng hồ sơ được nhập

Tức là, số lượng hồ sơ không được biết trước, nên sau khi người sử dụng nhập xong dữ liệu của một hồ sơ 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 16, chủ đề 7.

Ví dụ - thuật toán đoạn chương trình nhập dữ liệu theo dạng không biết trước số lượng hồ sơ sẽ được nhập, cấu trúc dữ liệu theo cách 1 :

Khởi tạo 10 mảng một chiều với 1000 phần tử

Dùng biến i với giá trị khởi gán là 0 để kiểm soát phần tử hiện đang nhập

Nếu $i \geq 1000$ thì thông báo không nhận hồ sơ nữa

Ngược lại, cho người sử dụng nhập vào thông tin của hồ sơ thứ $i + 1$

Họ tên nếu không rỗng thì gán vào phần tử thứ i của mảng 1

Ngày sinh gán vào phần tử thứ i của mảng 2

Tháng sinh gán vào phần tử thứ i của mảng 3

Năm sinh gán vào phần tử thứ i của mảng 4

Quê quán gán vào phần tử thứ i của mảng 5

Điểm toán gán vào phần tử thứ i của mảng 6

Điểm lý gán vào phần tử thứ i của mảng 7

Điểm hoá gán vào phần tử thứ i của mảng 8

Điểm văn gán vào phần tử thứ i của mảng 9

Điểm anh văn gán vào phần tử thứ i của mảng 10

Tăng giá trị biến i lên một đơn vị

Hỏi người sử dụng có tiếp tục muốn nhập hồ sơ tiếp theo hay không ?

Nếu có thì quay lại lệnh thứ 3

Nếu không thì kết thúc việc nhập dữ liệu

Lấy giá trị của biến i làm số lượng phần tử trong 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.

Tính điểm trung bình

Sau khi đã nhập xong dữ liệu cho các hồ sơ, ta có thể cho chương trình tính điểm trung bình cộng theo 5 cột điểm.

Tìm kiếm hồ sơ theo họ tên

Với cấu trúc dữ liệu theo cách 1, thuật toán tìm kiếm theo họ tên như sau :

Đầu vào : chuỗi ký tự họ tên

Đầu ra : i - vị trí của hồ sơ trong mảng

Khởi gán $i = 0$

Nếu $i >$ số lượng phần tử trong mảng

Thông báo là không có hồ sơ cần tìm

Trả ra giá trị $i = -1$

Ngược lại,

Nếu phần tử thứ i trong mảng 1 là chuỗi bằng với chuỗi đầu vào

Kết thúc, giá trị của i là vị trí hồ sơ muốn tìm

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

Quay lại bước 2 trong thuật toán này

Sửa thông tin một hồ sơ nào đó

Để sửa thông tin một hồ sơ nào đó, ta có thể sử dụng thuật toán sau (cho cấu trúc dữ liệu theo cách 1) :

Yêu cầu người sử dụng nhập vào họ tên của hồ sơ muốn sửa

Sử dụng thuật toán *tìm kiếm hồ sơ theo họ tên* để lấy vị trí của hồ sơ trong mảng

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

Thông báo không có hồ sơ 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 hồ sơ ra màn hình

Yêu cầu người sử dụng nhập dữ liệu mới

Gán dữ liệu mới vào đúng mảng

Xoá một hồ sơ

Tương tự thuật toán sửa thông tin một hồ sơ, thuật toán xoá một hồ sơ như sau :

Yêu cầu người sử dụng nhập vào họ tên của hồ sơ muốn xoá

Sử dụng thuật toán *tìm kiếm hồ sơ theo họ tên* để lấy vị trí của hồ sơ trong mảng

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

Thông báo không có hồ sơ 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 hồ sơ 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

Thuật toán để thực hiện xoá một phần tử trên mảng một chiều như sau :

Đầu vào : i là vị trí muốn xoá

Dùng vòng lặp duyệt qua từ phần tử thứ i đến phần tử kế cuối của mảng

Trong mỗi vòng lặp, ta thực hiện

Lấy giá trị phần tử thứ $i+1$ gán cho phần tử thứ i

Giảm giá trị số lượng phần tử trong mảng xuống một đơn vị

In danh sách trúng tuyển

Để in danh sách trúng tuyển, người sử dụng cần nhập vào điểm sàn. Chương trình sẽ duyệt qua tất cả các hồ sơ, nếu điểm trung bình của hồ sơ nào cao hơn điểm sàn thì in thông tin của hồ sơ đó ra màn hình.

Ngoài ra, các bạn có thể mở rộng hơn chức năng này bằng cách sắp xếp giảm dần theo điểm trung bình và danh sách in ra sẽ từ điểm cao đến điểm thấp.

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 HO SO XET TUYEN

1 - Nhap ho so moi

2 - Sua thong tin ho so

3 - Xoa ho so

4 - In danh sach trung tuyen

5 - In tat ca ho so

6 - 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 6 (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.*