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 02 tháng 12 năm 2013***
Nộp bằng cách gởi 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 thu chi cá nhân.

Quản lý thu chi cá nhân

Trong thời buổi vật giá tăng cao mà thu nhập lại thấp, người dân ngày càng phải thắt chặt thu chi. Do đó, nhu cầu về một phần mềm quản lý thu chi cá nhân là rất cấp thiết. Phần mềm này cho phép người sử dụng nhập vào các khoản thu và chi. Mỗi khoản thu chi thuộc vào một loại thu chi nào đó. Phần mềm này hỗ trợ bốn phân loại thu nhập là lương, thưởng, lãi (ngân hàng) và khác; và sáu phân loại chi tiêu là tiền điện, tiền nước, lương thực, quần áo, y tế và khác. Người sử dụng có thể xem, sửa và xoá thông tin các khoản thu chi. Cuối mỗi tháng, người sử dụng có thể tổng hợp các thông tin về số tiền đã thu, chi, số tiền thu và chi cho mỗi loại thu chi, và số dư.

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 thu chi cá nhân. Chương trình này cho phép người sử dụng nhập vào các khoản thu chi, thêm các phân loại cho từng khoản thu chi, xem, sửa, xoá các khoản thu chi và thống kê thu chi theo phân loại hoặc theo tháng, năm.

Các yêu cầu cụ thể của chương trình như sau :

1. Có cấu trúc dữ liệu để lưu trữ các khoản thu chi
2. Thêm vào một khoản thu chi mới
3. Các khoản thu chi được phân biệt thông qua mã thu chi (Một số để phân biệt các khoản thu chi với nhau, mỗi khoản thu chi có một con số riêng. Chương trình có thể sinh tự động mã này.)
4. Cho phép phân loại các khoản thu theo 4 loại : lương, thưởng, lãi và khác
5. Cho phép phân loại các khoản chi theo 6 loại : tiền điện, tiền nước, lương thực, quần áo, y tế và khác
   Một khoản thu/chi có thể thuộc vào một hoặc nhiều phân loại
6. Tìm kiếm một khoản thu chi theo mã thu chi
7. Xem các khoản thu chi của một tháng nào đó
8. Cho phép sửa thông tin một khoản thu chi
9. Cho phép xoá một khoản thu chi
10. In thống kê thu chi theo tháng - Thống kê và in ra được các thông tin sau : số dư đầu tháng, tổng thu trong tháng, tổng thu, phầm trăm tổng thu theo phân loại, tổng chi trong tháng, tổng chi, phần trăm tổng chi theo phân loại và số dư cuối tháng.
11. In thống kê thu chi theo phân loại thu chi - Thống kê và in ra được các thông tin sau : tổng thu, phần trăm tổng thu, tổng chi, phần trăm tổng chi theo phân loại thu chi người sử dụng yêu cầu.
12. (Tuỳ chọn - cộng điểm) Hỗ trợ đối số dòng lệnh
13. (Tuỳ chọn - cộng điểm) Lưu các khoản thu chi vào file
14. (Tuỳ chọn - cộng điểm) Đọc các khoản thu chi từ file
15. (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 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 khoản chi tiêu sau :

Ngày	**Số tiền**	**Thu/Chi**	**Phân loại**
10/01/2013	4.000.000	Thu	Lương, Thưởng
02/01/2013	300.000	Chi	Tiền điện
05/01/2013	150.000	Chi	Tiền nước
08/01/2013	200.000	Chi	Lương thực
10/01/2013	1.500.000	Chi	Lương thực
15/01/2013	1.000.000	Thu	Lãi
17/01/2013	200.000	Chi	Áo quần
20/01/2013	300.000	Chi	Lương thực, Khác
24/01/2013	400.000	Chi	Y tế
02/02/2013	320.000	Chi	Tiền điện
05/02/2013	145.000	Chi	Tiền nước
10/02/2013	3.500.000	Thu	Lương
10/02/2013	1.700.000	Chi	Lương thực
15/02/2013	1.000.000	Thu	Lãi
17/02/2013	400.000	Chi	Áo quần
18/02/2013	500.000	Chi	Lương thực
19/02/2013	100.000	Chi	Khác
24/02/2013	400.000	Chi	Y tế

2. Sửa ngày của khoản chi ngày 24/02/2013 thành ngày 29/02/2013
3. Sửa số tiền của khoản thu ngày 15/01/2013 thành 800.000
4. Xoá khoản chi ngày 19/02/2013
5. In thống kê thu chi cho tháng 2 năm 2013
6. In thống kê cho phân loại Lương thực của tháng 1 năm 2013

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 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 *các* khoản thu chi. Mỗi khoản thu chi bao gồm các thông tin như : mã thu chi, số tiền, thu/chi, các phân loại thu và các phân loại chi, ngày thực hiện thu/chi. Như vậy, cấu trúc dữ liệu 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 đại diện cho mã thu chi
    - một số thực cho số tiền thu/chi
    - một ký tự hoặc chuỗi ký tự hoặc giá trị kiểu liệt kê (enum) để phân biệt khoảng tiền trên là thu hay chi
    - một danh sách các ký tự hoặc chuỗi ký tự hoặc giá trị kiểu liệt kê để lưu các phân loại thu
    - một danh sách các ký tự hoặc chuỗi ký tự hoặc giá trị kiểu liệt kê để lưu các phân loại chi
    - ba số cho ngày thực hiện thu/chi

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 mỗi thông tin của một khoản thu chi. Trong đó, thành phần thứ i trong mỗi mảng/`List` tương ứng với khoản thu chi thứ i.
- *Cách 2* : các mảng hai chiều
- *Cách 3* : kết hợp cả mảng một chiều và hai chiều
- *Cách 4* : một mảng một chiều hoặc `List` mà mỗi phần tử 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 ThuChi
2  {
3      public int ma; // ma thu chi
4      public double sotien; // so tien
5      public LoaiThuChi loaiTC; // loai thu/chi
6      public List<PhanLoaiThu> listThu; // danh sach
           cac phan loai thu
7      public List<PhanLoaiChi> listChi; // danh sach
           cac phan loai chi
8      public int ngay;
9      public int thang;
10     public int nam;
11 };
12
13 enum LoaiThuChi {Thu, Chi};
14 enum PhanLoaiThu {Luong, Thuong, Lai, Khac};
15 enum PhanLoaiChi {TienDien, TienNuoc, LuongThuc,
       QuanAo, YTe, Khac};
```

Điểm quan trọng cần chú ý ở đây là với mỗi khoản thu chi, phải lưu trữ được danh sách *các phân loại của nó.* Do đó, ở mảng/`List` được dành cho việc lưu trữ hai danh sách này phải là mảng hai chiều hoặc mảng có các thành phần kiểu `List` hoặc `List` các `List`.

Nhập dữ liệu

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

1. Biết trước *số lượng khoản thu chi* được nhập

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

2. Không biết trước *số lượng khoản thu chi* được nhập

   Tức là, số lượng khoản thu chi không được biết trước, nên sau khi người sử dụng nhập xong dữ liệu của một khoản thu chi nào đó thì chương trình sẽ hỏi xem người sử dụng có muốn nhập tiếp hay không. Tuỳ 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 khoản thu chi sẽ được nhập, cấu trúc dữ liệu theo cách 4 :

Đầu vào : biến kiểu $List < ThuChi >$ đã được khởi tạo hoặc đã có dữ liệu, gọi là $listTC$

Đầu ra : $listTC$ đã thêm vào các khoản thu chi mới

```
Khởi tạo một biến kiểu ThuChi, gọi là tc
Khởi tạo hai danh sách listThu và listChi trong biến tc
Gán giá trị cho biến thành phần ma của tc bằng giá trị lớn nhất của các
mã thu chi đã có trong listTC cộng thêm 1
Cho người sử dụng nhập vào thông tin của khoản thu chi
    Số tiền nếu không âm thì gán vào biến thành phần sotien
    Loại thu chi gán vào biến loaiTC
    Nếu là loại thu
        Gán các phân loại thu vào listThu
    Nếu là loại chi
        Gán các phân loại chi vào listChi
    Ngày gán vào biến thành phần ngay
```

Tháng gán vào biến thành phần $thang$

Năm gán vào biến thành phần nm

Gán biến tc vào $listTC$

Hỏi người sử dụng có muốn nhập tiếp khoản thu chi mới hay không.

Nếu có, thì trở lại bước đầu tiên

Ngược lại, thoát ra khỏi thuật toán này

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.

Cách tạo danh sách thu chi như trên sẽ cho ta một danh sách được sắp xếp tăng dần theo mã thu chi. Do mã thu chi của khoản thu chi mới là giá trị mã lớn nhất và khoản thu chi mới được *thêm* vào cuối danh sách. Tuy nhiên, bạn có thể cho danh sách sắp xếp tăng dần theo thời gian của khoản thu chi. Khi đó, việc xử lý danh sách sẽ đơn giản hơn.

Để tạo ra danh sách được sắp xếp tăng dần theo thời gian, thay vì *thêm* khoản thu chi mới vào cuối danh sách, bạn *chèn* nó vào vị trí sao cho, các khoản thu chi ở trước vị trí đó có thời gian nhỏ hơn và các khoản thu chi ở sau vị trí đó có thời gian lớn hơn.

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 một khoản thu chi vào danh sanh. Còn bạn dùng mảng thì tham khảo câu 4, phần Mảng, bài thực hành 5.

Tìm kiếm một khoản thu chi theo mã thu chi

Với cấu trúc dữ liệu theo cách 4, thuật toán tìm kiếm theo mã thu chi như sau :

Đầu vào : mã thu chi cần tìm

Đầu ra : i - vị trí của khoản thu chi 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 thu chi đầu vào

Kết thúc, giá trị của i là vị trí khoản thu chi 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 khoản thu chi nào đó

Để sửa thông tin một khoản thu chi nào đó, ta có thể sử dụng thuật toán sau (cho cấu trúc dữ liệu theo cách 4) :

```
Yêu cầu người sử dụng nhập vào mã thu chi của khoản thu chi muốn sửa
Sử dụng thuật toán tìm kiếm khoản thu chi theo mã thu chi để lấy vị trí
của khoản thu chi trong danh sách
Nếu kết quả của thuật toán tìm kiếm là -1 (i == −1),
    Thông báo không có khoản thu chi 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 khoản thu chi ra màn hình
    Yêu cầu người sử dụng nhập dữ liệu mới
    Gán dữ liệu mới vào
```

Xoá một khoản thu chi

Tương tự thuật toán sửa thông tin một khoản thu chi, thuật toán xoá một khoản thu chi như sau :

```
Yêu cầu người sử dụng nhập vào mã thu chi của khoản thu chi muốn xoá
Sử dụng thuật toán tìm kiếm khoản thu chi theo mã thu chi để lấy vị trí
của khoản thu chi trong danh sách
Nếu kết quả của thuật toán tìm kiếm là -1 (i == −1),
    Thông báo không có khoản thu chi 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 khoản thu chi 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 thống kê thu chi theo tháng

Theo yêu cầu của bài tập, bạn phải thống kê và in ra được những giá trị sau :

- số dư đầu tháng - tổng thu của tất cả các tháng trước tháng thống kê trừ cho tổng chi của tất cả các tháng trước tháng thống kê
- tổng thu trong tháng

- tổng thu theo các phân loại thu
- phần trăm tổng thu theo các phân loại thu
- tổng chi trong tháng
- tổng chi theo các phân loại chi
- phần trăm tổng chi theo các phân loại chi
- số dư cuối tháng - bằng số dư đầu tháng cộng tổng thu trong tháng rồi trừ cho tổng chi trong tháng

In thống kê thu chi theo phân loại

Tương tự như thống kê thu chi theo tháng, nhưng thông tin thống kê đơn giản hơn.

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 CHI TIEU
1 - Them khoan thu chi moi
2 - In thu chi theo thang nam
3 - Xoa khoan thu chi
4 - Sua thong tin khoan thu chi
5 - Thong ke thu chi theo thang
6 - Thong ke thu chi theo phan loai
7 - Lay du lieu tu file
8 - Luu du lieu ra file
9 - 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 9 (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 qlct.exe
2 qlct.exe <num>
3 qlct.exe <path>
4 qlct.exe -h
```

Với `qlct.exe` là tên chương trình, `qlct` là viết tắt của `Quản lý chi tiêu`.

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>` khoản thu chi mới. Đây là cách nhập dữ liệu khi biết trước số lượng khoản thu chi 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 13 và 14 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 khoản thu chi có thể được quy định như sau :

- Dòng đầu tiên trong tập tin là 1 số - thể hiện số lượng khoản thu chi lưu trong tập tin (gọi là `n`)
- Từ dòng thứ hai đến hết file, có tất cả là `n` nhóm dòng, mỗi nhóm có 5 dòng

- dòng thứ nhất là số tiền
- dòng thứ hai là ký tự hoặc chuỗi ký tự hoặc giá trị kiểu liệt kê thể hiện đây là khoản thu hay khoản chi
- dòng thứ ba liệt kê các giá trị phân loại thu (phân cách nhau bằng khoảng trống)
- dòng thứ tư liệt kê các giá trị phân loại chi (phân cách nhau bằng khoảng trống)
- dòng thứ năm là ngày tháng năm được in theo dạng dd/mm/yyyy

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

```
18
4000000
Thu
Luong Thuong

10/01/2013
300000
Chi

TienDien
02/01/2013
```

Thuật toán đọc dữ liệu từ file cho cấu trúc dữ liệu cách 4 như sau :

Đầu vào : biến kiểu $List < ThuChi >$ đã được khởi tạo hoặc đã có dữ liệu, gọi là $listTC$

Đầu ra : $listTC$ đã thêm vào các khoản thu chi mới được đọc từ file

Đọc dòng đầu tiên trong file để biết số lượng nhóm n
Khởi tạo biến $i = 0$
Nếu $i < n$
    Khởi tạo một biến kiểu $ThuChi$, gọi là tc
    Khởi tạo hai danh sách $listThu$ và $listChi$ trong biến tc
    Gán giá trị cho biến thành phần ma của tc bằng giá trị lớn nhất của các mã thu chi đã có trong $listTC$ cộng thêm 1
    Đọc dòng tiếp theo trong file gán cho biến $sotien$
    Đọc dòng tiếp theo trong file gán cho biến $loaiTC$
    Đọc dòng tiếp theo trong file
        Dùng hàm Split tách các từ trong chuỗi ký tự
        và gán vào danh sách $listThu$
    Đọc dòng tiếp theo trong file

Dùng hàm Split tách các từ trong chuỗi ký tự
và gán vào danh sách $listChi$
Đọc dòng tiếp theo trong file
Dùng hàm Split tách các số trong chuỗi ký tự
và gán vào ba biến $ngay$, $thang$, nam
Tăng giá trị của i lên một, rồi quay lại bước 3

Xuất dữ liệu ra file văn bản

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

Đầu vào : biến kiểu $List < ThuChi >$ đã được khởi tạo hoặc đã có dữ liệu, gọi là $listTC$

Đầu ra : file văn bản chứa dữ liệu các khoản thu chi

Ghi số lượng khoản thu chi trong $listTC$ vào dòng đầu tiên của file
Duyệt qua từng khoản thu chi trong $listTC$, với mỗi khoản thu chi
Ghi giá trị biến $sotien$ vào dòng tiếp theo
Ghi giá trị biến $loaiTC$ vào dòng tiếp theo
Duyệt qua các giá trị trong danh sách $listThu$ để tạo ra một chuỗi ký tự với mỗi giá trị trong danh sách cách nhau bằng khoảng trống
Ghi chuỗi ký tự vừa tạo ra vào dòng tiếp theo
Duyệt qua các giá trị trong danh sách $listThu$ để tạo ra một chuỗi ký tự với mỗi giá trị trong danh sách cách nhau bằng khoảng trống
Ghi chuỗi ký tự vừa tạo ra vào dòng tiếp theo
Định dạng chuỗi ngày tháng theo dạng dd/mm/yyyy từ dữ liệu của 3 biến $ngay$, $thang$, nam
Ghi chuỗi ngày tháng vừa tạo ra vào dòng tiếp theo

Chúc các em làm bài tốt !