

NGÂN HÀNG BÀI TẬP MÔN LẬP TRÌNH HDT

PHẦN I : ĐỊNH NGHĨA LỚP

Bài 1: Viết chương trình xây dựng một lớp hình tròn đơn giản có thành phần dữ liệu là bán kính r và có các phương thức như: nhập dữ liệu cho r , tính toán và hiển thị chu vi, diện tích của hình tròn đó.

Bài 2: Viết chương trình xây dựng lớp **tam giác**, có các thành phần sau:

- Các thuộc tính là các cạnh a, b, c
- Các hàm thành phần bao gồm:
 - + Hàm nhập giá trị cho các cạnh (Kiểm tra tính hợp lệ đảm bảo là 3 cạnh của một tam giác)
 - + Hàm tính diện tích tam giác
 - + Hàm kiểm tra tam giác(đều, vuông cân, cân, vuông, thường)
 - + Hàm hiển thị thông tin(diện tích, tính chất tam giác)

Chương trình chính: Khai báo một đối tượng lớp tam giác, nhập các cạnh cho tam giác, hiển thị ra màn hình các thông tin của tam giác (diện tích, tính chất tam giác).

Bài 3: Viết chương trình tạo lớp điểm (point) có thuộc tính là các tọa độ x, y và các phương thức cơ bản như khởi tạo, hủy bỏ, nhập, xuất, tính khoảng cách giữa hai điểm.

Chương trình chính: Nhập vào n điểm từ bàn phím. Tìm và hiển thị khoảng cách lớn nhất giữa hai điểm trong n điểm đó

Bài 4: Viết chương trình xây dựng lớp **xâu ký tự**, có các thành phần sau :

- Các thuộc tính : $\text{char} * \text{str}, \text{int length}$
- Hàm tạo không tham số
- Hàm tạo một tham số ($\text{char} * s$)
- Hàm nhập xâu.

- Hàm hiển thị chuỗi và độ dài chuỗi
- Hàm nối chuỗi để cộng hai chuỗi (xaukytu s)
- Hàm hủy: Hủy bộ nhớ đã cấp phát cho chuỗi.

Chương trình chính:

- + Khai báo đối tượng lớp chuỗi ký tự s1 bằng hàm khởi tạo không tham số, nhập chuỗi s1, hiển thị chuỗi s1 ra màn hình
- + Khai báo đối tượng lớp chuỗi ký tự s2 bằng hàm khởi tạo 2 tham số, hiển thị chuỗi s2 ra màn hình
- + Nối 2 chuỗi s1, s2 và hiển thị kết quả ra màn hình.

Bài 5: Viết chương trình tạo một lớp vector gồm có các thành phần sau:

- Các thuộc tính : float * v; int n
- Các phương thức:
 - Hàm thiết lập không tham số:
 - + Nhập số phần tử của v.
 - + Cấp phát bộ nhớ cho v.
 - + Nhập các tọa độ của v.
 - Hàm thiết lập một tham số
 - Vào: n (số phần tử của v).
 - Ra: n (số phần tử của v).
 - + Cấp phát bộ nhớ cho v (n phần tử).
 - + Nhập các tọa độ của v.
 - Hàm thiết lập hai tham số.
 - + Vào: n (số phần tử của v).
 - : Các tọa độ của v
 - + Ra:
 - Gán thuộc tính n;
 - Gán giá trị cho các tọa độ của v;

- Hàm thiết lập sao chép.
+ Vào: Đối tượng vector p.
+ Ra: Sao chép dữ liệu từ p sang đối tượng vừa tạo
- Hàm hiển thị:
+ Hiển thị số phân tử n;
+ Hiển thị các tọa độ của v.
- Hàm huỷ:
+ Giải phóng bộ nhớ đã cấp phát cho v ;

Chương trình chính:

- + Tạo các đối tượng lớp vector.
- + Sử dụng các hàm đã được định nghĩa trong lớp vector.

Bài 6: Viết một chương trình xây dựng hai lớp: **một lớp thí sinh** và **một lớp danh sách thí sinh**.

Trong đó lớp **thí sinh** có dữ liệu bao gồm các thông tin: số báo danh, điểm toán, điểm hoá, điểm lý (Số báo danh gồm cả phần chữ và phần số, phần số gồm 3 chữ số tự động tăng DDL100, DDL101, DDL102...).

Lớp **danh sách thí sinh** có dữ liệu một mảng các thí sinh và số lượng phần tử thuộc mảng đó. Viết chương trình thực hiện các công việc sau:

1. Nhập và hiển thị một danh sách các thí sinh từ bàn phím
2. Sắp xếp danh sách các thí sinh theo thứ tự tăng dần về điểm số
3. Hiển thị thông tin của các sinh viên có tổng điểm ≥ 18 điểm.

Bài 7: Viết chương trình xây dựng lớp **Time**, mô tả các thông tin về giờ, phút, giây. Lớp Time có các thành phần sau:

- + Các thuộc tính mô tả giờ, phút, giây;
- + Các hàm thành phần dùng để xác lập giá trị cho từng thành phần giờ, phút, giây (Có kiểm tra điều kiện giờ (0->23), phút(0->59), giây(0->59);
- + Hàm thành phần setTime(int,int,int) để xác lập thời gian

- + Hàm hiển thị giờ theo định dạng 24 tiếng (vd : 23:54:40);
- + Hàm hiển thị giờ theo định dạng 12 tiếng(vd : 11:54:40 PM);
- + Hàm tăngGiây() để tăng thời gian mỗi lần lên một giây. Chú ý các trường hợp tăng sang phút tiếp theo, tăng sang giờ tiếp theo, tăng sang ngày tiếp theo.

Chương trình chính khai báo một đối tượng thời gian là 23:59:58 và thực hiện tăng thời gian 5 giây đồng thời hiển thị thời gian cho mỗi lần tăng.

Bài 8 : Viết chương trình xây dựng một lớp **vector** như trong bài tập 5, một lớp **ma trận** với các phương thức cơ bản như khởi tạo, nhập, xuất, hủy. Sử dụng hàm bạn xây dựng một hàm ngoài lớp nhân ma trận (một đối tượng thuộc lớp matrix) với vector (đối tượng thuộc lớp vector).

Chương trình chính:

- + Tạo các đối tượng lớp vector, nhập dữ liệu cho các đối tượng, hiển thị thông tin ra màn hình.
- + Tạo các đối tượng lớp **ma trận**, hiển thị thông tin ra màn hình.

PHẦN 2 : ĐỊNH NGHĨA TOÁN TỬ TRÊN LỚP

Bài 1: Xây dựng một lớp **Point** gồm:

Thuộc tính: float x, y

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử: >>, <<
- Hàm bạn: KhoangCach (để tính khoảng cách giữa 2 Point)

Xây dựng lớp **Dagiac** gồm:

Thuộc tính: Point *a và Size;

Phương thức:

- Cấu tử.

- Hủy: hủy đối tượng đã được khởi tạo.
- Nạp chồng toán tử: >>, <<

Hàm bạn:

- KiemTra: kiểm tra xem đa giác có phải là tam giác hay không.
- KiemTraCan: kiểm tra xem đa giác có phải đa giác cân hay không?

Chương trình chính: Nhập vào từ bàn phím n Dagiac, hiển thị thông tin những Dagiac thỏa mãn là tam giác lên màn hình.

Bài 2: Xây dựng lớp số phức (Complex) có dạng $\text{real} + \text{image} * i$:

- Các thuộc tính : real , image;
- Các phương thức cơ bản như khởi tạo, nhập, xuất, hủy bỏ...
- Định nghĩa các toán tử sau trên lớp Complex:
 - + Toán tử cộng (+)
 - + Toán tử cộng thành phần real của số phức b lên x ($b.\text{real} + x$)
 - + Toán tử trừ (-)
 - + Toán tử nhập (>>)
 - + Toán tử xuất (<<)

Chương trình chính:

- Nhập vào 2 số phức a, b.
- Tính tổng, hiệu 2 số phức a và b đã nhập, hiển thị kết quả ra màn hình.
- Nhập vào 1 mảng các số phức, sắp xếp mảng vừa nhập theo thứ tự phần thực tăng dần, hiển thị mảng đã nhập ra màn hình.

Bài 3: Viết chương trình định nghĩa lớp Phân số :

- Các thuộc tính: ts, ms;
- Các phương thức cơ bản như khởi tạo, nhập, xuất, hủy bỏ...
- Định nghĩa các toán tử sau trên lớp Phân số:
 - + Toán tử cộng (+)
 - + Toán tử trừ (-)
 - + Toán tử nhân (*)
 - + Toán tử chia (/)

- + Toán tử nhập (>>)
- + Toán tử xuất (<<)

Chương trình chính:

- Nhập vào 2 phân số a, b (sử dụng toán tử >>).
- Hiển thị a, b ra màn hình (sử dụng toán tr <<);
- Tính tổng, hiệu, tích, thương của 2 phân số và hiển thị kết quả ra màn hình.

Bài 4 : Xây dựng lớp **vector** có các thành phần sau:

- Các thuộc tính : int n; int * v;
- Hàm thiết lập không tham số
- Hàm thiết lập hai tham số (int size, int * a);
- Hàm thiết lập sao chép
- Hàm hủy
- Định nghĩa các toán tử sau trên lớp vector:
 - + Toán tử gán (=)
 - + Toán tử so sánh bằng (=)
 - + Toán tử so sánh khác (!=)
 - + Toán tử lấy phần tử tại chỉ số ([])
 - + Toán tử nhập (>>)
 - + Toán tử xuất (<<)

Chương trình chính : Tạo các đối tượng lớp vector, kiểm tra việc sử dụng các toán tử đã định nghĩa, trong đó có sử dụng biến và hàm static để đếm số đối tượng vector được khởi tạo

Bài 5 : Xây dựng lớp **string** có các thành phần sau

- Các thuộc tính : char * s, int length;
- Các hàm thành phần cơ bản : hàm khởi tạo một tham số, hàm khởi tạo sao chép, hàm hủy
- Định nghĩa các toán tử sau trên lớp:
 - + Toán tử gán (=),
 - + Toán tử cộng (+=),
 - + Toán tử kiểm tra chuỗi rỗng (!),
 - + Toán tử so sánh bằng (=),

- + Toán tử so sánh không bằng (!=),
- + Toán tử so sánh lớn hơn(>),
- + Toán tử so sánh nhỏ hơn (<),
- + Toán tử so sánh lớn hơn hoặc bằng(>=),
- + Toán tử so sánh nhỏ hơn hoặc bằng(<=),
- + Toán tử lấy phần tử của xâu tại chỉ số ([]),
- + Toán tử nhập (>>),
- + Toán tử xuất (<<)

Chương trình chính : Tạo các đối tượng lớp **string**, kiểm tra việc sử dụng các toán tử đã định nghĩa.

PHẦN 3: KHUÔN HÌNH HÀM, KHUÔN HÌNH LỚP

Bài 1: Xây dựng lớp Phân số có các thành phần sau:

- Thành phần dữ liệu gồm: tuso,mauso;
- Các phương thức cơ bản như: Khởi tạo không tham số, khởi tạo có tham số, hủy bỏ
- Định nghĩa các toán tử sau trên lớp Phân số:
 - + Toán tử nhập phân số >>
 - + Toán tử hiển thị phân số <<
 - + Toán tử cộng 2 phân số (+)
 - + Toán tử trừ 2 phân số (-)
 - + Toán tử nhân 2 phân số (*)
 - + Toán tử chia 2 phân số (*)
 - + Toán tử so sánh hai phân số (>)

Sử dụng **khôn hình hàm** để xây dựng chương trình chính thực hiện việc tìm phân số lớn nhất trong một danh sách các phân số và tìm số lớn nhất trong một danh sách các số nguyên được nhập vào từ bàn phím.

Bài 2: Xây dựng lớp **Date** có các thành phần sau:

Thuộc tính: Ngay, Thang, Nam

Phương thức:

- Khởi tạo.
- Nạp chồng toán tử: >>, <<
- Nạp chồng toán tử lớn hơn: >

Xây dựng một khuôn hình lớp danh sách kê gồm:

Thuộc tính: T *arr; Size

Phương thức:

- Khởi tạo
- Nạp chồng toán tử: >>, <<

- Sắp xếp mảng danh sách theo chiều tăng dần

Chương trình chính:

- Nhập vào từ bàn phím danh sách các số nguyên.
- Sắp xếp danh sách theo chiều tăng dần.
- Hiển thị danh sách sau sắp xếp.
- Nhập vào từ bàn phím DanhSach các Date.
- Sắp xếp danh sách theo chiều tăng dần.
- Hiển thị danh sách sau sắp xếp.

Bài 3: Xây dựng khuôn hình hàm **Swap** cho phép trao đổi giá trị 2 biến số truyền vào. Xây dựng khuôn hình hàm **Sort** dùng để sắp xếp giá trị của một mảng trong đó có sử dụng khuôn hình hàm Swap. Xây dựng khuôn hình hàm **Print** để in giá trị của một mảng ra màn hình.

Xây dựng lớp **Phanso** để sử dụng các khuôn hình hàm trên.

Bài 3: Xây dựng họ khuôn hình hàm **Min** cho phép:

- + Tìm giá trị nhỏ nhất giữa 2 số nguyên (int), 2 số thực (float), 2 chuỗi ký tự.
- + Tìm phần tử nhỏ nhất trong 1 mảng nguyên, thực, ký tự, chuỗi ký tự.

Bài 4: Xây dựng họ khuôn hình hàm **Max** cho phép:

- + Tìm giá trị lớn nhất giữa 2 số nguyên (int), 2 số thực (float), 2 chuỗi ký tự.
- + Tìm phần tử lớn nhất trong 1 mảng nguyên, thực, ký tự, chuỗi ký tự.

Bài 5: Xây dựng khuôn hình lớp vector có:

- Các thuộc tính: $T * v$, $int n$
- Các hàm thành phần cơ bản như khởi tạo, nhập, xuất, hủy bỏ.
- Định nghĩa các toán tử trên lớp vector:
 - + Toán tử gán (=)
 - + Toán tử so sánh bằng (==)
 - + Toán tử so sánh khác (!=)
 - + Toán tử lấy phần tử tại chỉ số ([])
 - + Toán tử nhập (>>)
 - + Toán tử xuất (<<)

Sử dụng khuôn hình lớp vector với các kiểu dữ liệu int, float, phanso.

Bài 6: Định nghĩa khuôn hình lớp ngăn xếp **Stack1** với:

- Các thuộc tính: T a[max]; int top;
- Các hàm thành phần:
 - + Khởi tạo không tham số.
 - + Khởi tạo 2 tham số
 - + Hủy bỏ
 - + Kiểm tra stack rỗng
 - + Kiểm tra stack đầy
 - + Chèn một phần tử có kiểu T vào stack// void push(T & itemp)
 - + Lấy phần tử ra khỏi stack// T pop()
 - + Lấy/Xem giá trị phần tử ở đỉnh stack (không loại phần tử đó khỏi stack)

Sử dụng khuôn hình lớp Stack1 với các kiểu dữ liệu int, char, float, số phức.

Bài 7: Định nghĩa khuôn hình lớp ngăn xếp **Stack2** với:

- Các thuộc tính:
 - o Top (con trỏ đến phần tử ở đỉnh ngăn xếp)
- Các hàm thành phần:
 - + Khởi tạo không tham số.
 - + Khởi tạo 2 tham số
 - + Hủy bỏ
 - + Kiểm tra stack rỗng
 - + Chèn một phần tử có kiểu T vào stack// void push(T & itemp)
 - + Lấy phần tử ra khỏi stack// T pop()
 - + Lấy/Xem giá trị phần tử ở đỉnh stack (không loại phần tử đó khỏi stack)

Sử dụng khuôn hình lớp Stack2 với các kiểu dữ liệu int, char, float, số phức.

Bài 8: Định nghĩa khuôn hình lớp hàng đợi **Queue1** với:

- Các thuộc tính:
 - o Mảng lưu trữ các phần tử (kiểu T)
 - o Chỉ số phần tử đầu hàng đợi
 - o Chỉ số phần tử cuối hàng đợi
- Các hàm thành phần:

- + Khởi tạo không tham số.
- + Khởi tạo 2 tham số.
- + Hủy bỏ.
- + Kiểm tra hàng đợi rỗng.
- + Kiểm tra hàng đợi đầy.
- + Chèn một phần tử có kiểu T vào queue.
- + Lấy phần ở đầu hàng đợi.
- + Lấy/Xem giá trị phần tử ở đầu hàng đợi (không loại phần tử đó khỏi queue).

Sử dụng khuôn hình lớp Queue1 với các kiểu dữ liệu int, char, float, số phức.

Bài 9: Định nghĩa khuôn hình lớp hàng đợi **Queue2** với:

- Các thuộc tính:
 - front: con trỏ đến phần tử đầu hàng đợi.
 - rear: con trỏ đến phần tử cuối hàng đợi
- Các hàm thành phần:
 - + Khởi tạo không tham số.
 - + Khởi tạo 2 tham số.
 - + Hủy bỏ.
 - + Kiểm tra hàng đợi rỗng
 - + Chèn một phần tử có kiểu T vào queue
 - + Lấy phần ở đầu hàng đợi
 - + Lấy/Xem giá trị phần tử ở đầu hàng đợi (không loại phần tử đó khỏi queue)

Sử dụng khuôn hình lớp Queue2 với các kiểu dữ liệu int, char, float, số phức.

Bài 10: Xây dựng lớp Danh sách kê có:

- Các thuộc tính :T a[max], int size;
- Các phương thức cơ bản như :
 - + Hàm tạo;
 - + Hàm hủy
 - + Hàm kiểm tra danh sách rỗng
 - + Hàm kiểm tra danh sách đầy

- + Hàm chèn một phần tử vào danh sách
- + Hàm xóa một phần tử trong danh sách
- + Hàm tìm kiếm phần tử trong danh sách

Sử dụng khuôn hình lớp Danh sách kèm với các kiểu dữ liệu int, char, float, chuỗi ký tự, Phân số.

Bài 10: Xây dựng khuôn hình lớp Danh sách liên kế đơn có các thuộc tính:

- pHead (con trỏ đến nút đầu tiên trong danh sách)
- pTail (con trỏ đến nút cuối cùng trong danh sách);
- Các phương thức cơ bản như :
 - + Hàm tạo;
 - + Hàm hủy;
 - + Hàm tạo nút
 - + Hàm kiểm tra danh sách rỗng
 - + Hàm chèn một phần tử vào danh sách
 - + Hàm xóa một phần tử trong danh sách
 - + Hàm tìm kiếm phần tử trong danh sách

Sử dụng khuôn hình lớp Danh sách liên kết với các kiểu dữ liệu int, char, float, chuỗi ký tự, Phân số.

Bài 11: Xây dựng khuôn hình lớp Danh sách liên kế đôi có các thuộc tính:

- pHead (con trỏ đến nút đầu tiên trong danh sách)
- pTail (con trỏ đến nút cuối cùng trong danh sách);
- Các phương thức cơ bản như :
 - + Hàm tạo;
 - + Hàm hủy;
 - + Hàm tạo nút
 - + Hàm kiểm tra danh sách rỗng
 - + Hàm chèn một phần tử vào danh sách

- + Hàm xóa một phần tử trong danh sách
- + Hàm tìm kiếm phần tử trong danh sách

Sử dụng khuôn hình lớp Danh sách liên kết đôi với các kiểu dữ liệu int, char, float, chuỗi ký tự, Phân số.

Bài 12: Sử dụng khuôn hình lớp **BinaryTree**, với :

- Thuộc tính: Node *root;
- Mỗi nút trên cây gồm 3 thuộc tính: T data, Node*left, Node*right;
- Các hàm thành phần:
 - + Khởi tạo.
 - + Hủy bỏ.
 - + Chèn nút vào cây.
 - + Duyệt cây.
 - + Tìm kiếm một nút trên cây
 - + Xóa 1 nút trên cây

Sử dụng khuôn hình lớp BinaryTree với các kiểu dữ liệu int, char, float.

Bài 13: Xây dựng lớp **CauThu** gồm:

Thuộc tính: HoTen, Tuoi, ViTri, CanNang, ChieuCao

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử >>, <<
- Nạp chồng toán tử < (Một CauThu là nhỏ hơn nếu chiều cao thấp hơn, nếu chiều cao bằng nhau thì xét cân nặng cao hơn)

Xây dựng khuôn hình **DanhSach** gồm:

Thuộc tính: T *ds; int Size

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử >>, <<

- Sắp xếp DanhSach theo chiều giảm dần.

Chương trình chính: Nhập vào từ bàn phím:

- 1 danh sách n số nguyên
- 1 danh sách m CauThu.

Sắp xếp 2 danh sách đã nhập, hiển thị thông tin lên màn hình.

Bài 14: Xây dựng lớp **Oxy** gồm:

Thuộc tính: float x, y;

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- Nạp chồng toán tử - (tính khoảng cách giữa 2 Oxy)

Xây dựng khuôn hình **DanhSach** gồm:

Thuộc tính: T *ds; int Size

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử nhập (cấp phát động)
- Nạp chồng toán tử xuất
- Hàm MAX trả về khoảng cách xa nhất giữa 2 giá trị bất kỳ trong ds.

Chương trình chính: Nhập vào từ bàn phím 1 DanhSach. Hiển thị danh sách; hiển thị khoảng cách xa nhất giữa 2 giá trị bất kỳ trong DanhSach đã nhập.

Bài 15: Xây dựng lớp **ThoiGian** gồm:

Thuộc tính: Gio, Phut, Giay

Phương thức:

- Khởi tạo.
- Nạp chồng toán tử: >>, <<
- Hàm bạn: KhoangCach(ThoiGian a, ThoiGian b) : tính khoảng cách giữa 2 mốc ThoiGian.
- Nạp chồng toán tử lớn hơn: >

Xây dựng một khuôn hình **DanhSach** gồm:

Thuộc tính: T *arr; Size

Phương thức:

- Khởi tạo
- Nạp chồng toán tử: >>, <<
- Sắp xếp danh sách

Chương trình chính:

- Nhập vào từ bàn phím DanhSach các mốc ThoiGian.
- Sắp xếp danh sách theo chiều tăng dần.
- Hiện thị danh sách sau sắp xếp.

PHẦN 4: KẾ THỪA

Bài 1: Xây dựng lớp **Color** gồm:

Thuộc tính: TenMau, MaMau

Phương thức:

- Cấu tử không tham số
- Cấu tử có tham số
- Hủy
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- getTenMau : hàm trả về TenMau

Xây dựng lớp **Point** gồm:

Thuộc tính: int x, y

Phương thức:

- Cấu tử không tham số
- Cấu tử có tham số
- Hủy
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- CheoChinh : hàm kiểm tra Point có thuộc đường chéo chính hay không (1 điểm thuộc đường chéo chính khi và chỉ khi tung độ bằng hoành độ).

Xây dựng lớp **Pixel** kế thừa từ lớp **Color** và **Point** bao gồm thêm:

Phương thức:

- Cấu tử không tham số
- Cấu tử có tham số
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- KiemTra: hàm kiểm tra Pixel thuộc đường chéo chính và có màu “Xanh”

hay không?

Chương trình chính: Nhập vào từ bàn phím n Pixel (n nhập từ bàn phím). Hiện thị thông tin các Pixel thuộc đường chéo chính và có màu xanh.

Bài 2: Xây dựng lớp **VeMayBay** (TenChuyen, NgayBay, GiaVe) gồm:

Thuộc tính:

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất
- getGiaVe : hàm trả về giá vé.

Xây dựng lớp **Người** gồm:

Thuộc tính: HoTen, GioiTinh, Tuoi

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất

Xây dựng lớp **HànhKhách** (mỗi hành khách được mua nhiều vé) kế thừa lớp **Người** bổ sung thêm:

Thuộc tính: VeMayBay *ve; int SoLuong;

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất
- TongTien: trả về Tổng số tiền phải trả của hành khách

Chương trình chính: Nhập vào 1 danh sách n hành khách (n nhập từ bàn phím). Hiển thị danh sách hành khách và số tiền phải trả tương ứng của mỗi khách hàng.

Sắp xếp danh sách hành khách theo chiều giảm dần của Tổng tiền.

Bài 3: Định nghĩa lớp **CanBo** gồm

Thuộc tính: HoTen, GioiTinh, NamSinh

Phương thức:

- Thiết lập, Hủy bỏ, nhập, xuất,
- Tui: Tính tuổi của CanBo với năm hiện tại là 2019
- NghiHuu: bool
- Luong: float

Định nghĩa lớp **BienChe** kế thừa lớp CanBo gồm

Thuộc tính: DonVi, HSLuong

Phương thức:

- Thiết lập, hủy bỏ, nhập, xuất.
- NghiHuu: BienChe nghỉ hưu khi đủ 60 tuổi.
- Luong: BienChe đã nghỉ hưu thì nhận lương hưu bằng 70% lương cứng,

công nhân chưa nghỉ hưu có $Lương = HSLuong * 1.310.000$

Định nghĩa lớp **HopDong** kế thừa lớp CanBo gồm

Thuộc tính: NamCongTac, HeSoThamNien, MucLuong

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất
- NghiHuu: HopDong nghỉ hưu khi số năm công tác là 35.
- Luong: HopDong đã nghỉ hưu thì không có lương, công nhân chưa nghỉ

hưu có Lương = MucLuong * HeSoThamNien

Chương trình chính: Khai báo một con trỏ **CanBo**, thực hiện định nghĩa một BienChe và HopDong, hiển thị thông tin của các đối tượng đó.

Bài 4: Xây dựng lớp **MonHoc** gồm

Thuộc tính: TenMon, DiemTP, DiemThi, SoTC

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất
- Hàm DTB trả về Điểm trung bình môn học
(DTB = DiemTP*0.3 + DiemThi*0.7)
- Hàm TinChi trả về SoTC của môn học.

Xây dựng lớp **SinhVien** gồm

Thuộc tính: MaSV, TenSV, SoMH (số lượng môn học)
MonHoc *a

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử: >>, <<
- Hàm DTK trả về điểm tổng kết của sinh viên.

$$DTK = \frac{\sum_{k=0}^n DTB_k * SoTC_k}{\sum_{j=0}^n SoTC_k}$$

(Điểm tổng kết bằng tổng của DTB*SoTC của tất cả các môn / tổng số tín chỉ của tất cả các môn).

Chương trình chính: Nhập từ bàn phím 1 SinhVien, hiển thị thông tin SinhVien lên màn hình với DTK của SinhVien đó.

Bài 5: Xây dựng lớp **ConNguoi** gồm:

Thuộc tính: HoTen, Tuoì, GioiTinh

Phương thức:

- Cấu tử.

- Hủy
- Nhập: ảo hóa
- Xuất: ảo hóa
- KhenThuong: ảo hóa

Xây dựng lớp **Sinhvien** kế thừa lớp **ConNguoi** bổ sung thêm:

Thuộc tính: Lớp, DTB

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất
- KhenThuong (Sinh viên được khen thưởng nếu DTB ≥ 8.0)

Xây dựng lớp **GiangVien** kế thừa lớp **ConNguoi** bổ sung thêm

Thuộc tính: BoMon, SoBaiBao

Phương thức:

- Cấu tử.
- Hủy
- Nhập
- Xuất
- KhenThuong (Giảng viên được khen thưởng nếu SoBaiBao ≥ 1 bài)

Khởi tạo một con trỏ ConNguoi,

- Khai báo một mảng gồm n SinhVien qua con trỏ này, nhập vào danh sách SinhVien và hiển thị danh sách SinhVien được khen thưởng.
- Khai báo một mảng gồm m GiangVien qua con trỏ này, nhập vào danh sách GiangVien và hiển thị danh sách GiangVien được khen thưởng.

Bài 6: Xây dựng lớp **Oxy** gồm:

Thuộc tính: float x, y;

Phương thức:

- Cấu tử.
- Hủy
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- Hàm bạn TrungDiem trả về giá trị trung điểm của 2 Oxy

Xây dựng lớp **Oxyz** kế thừa từ lớp **Oxy** bổ sung thêm:

Thuộc tính: float z;

Phương thức:

- Cấu tử.

- Hủy
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- Hàm bạn TrungDiem trả về giá trị trung điểm của 2 Oxyz

Chương trình chính: nhập vào từ bàn phím một mảng các điểm **Oxyz**.

Hiển thị giá trị các điểm đã nhập; Tìm và hiển thị các cặp điểm đối xứng qua tâm O.

Bài 7: Quản lý mượn sách thư viện

Khai báo lớp **Sách** gồm các thuộc tính (**mã sách, Tên sách, Tác giả, Chuyên ngành, Năm xuất bản**) – trong đó Chuyên ngành có thể là: Khoa học tự nhiên, Văn học – Nghệ thuật, Điện tử Viễn thông, Công nghệ thông tin. Mã sách là một số nguyên có 5 chữ số, tự động tăng.

Khai báo lớp **Người** (**Họ tên, Địa chỉ, SDT**)

Khai báo lớp **Bạn đọc** kế thừa từ lớp **Người** và có thêm (**mã thẻ đọc, loại bạn đọc, số sách mượn, danh sách các mã sách đã mượn**), mã bạn đọc là một số nguyên có 5 chữ số, tự động tăng. Loại bạn đọc có thể là sinh viên, học viên cao học hoặc giáo viên.

Chương trình chính:

- Nhập vào danh sách (mảng) các cuốn sách hiện có.
- Nhập vào danh sách (mảng) bạn đọc.
- Hiển thị danh sách bạn đọc đã nhập.
- Nhập vào 1 mã bạn đọc bất kỳ, hiển thị ra màn hình số sách và danh sách các cuốn sách mà bạn đọc đó đã mượn.

Bài 8: Quản lý đăng ký học theo tín chỉ

Khai báo lớp **Môn học** gồm các thuộc tính (**mã môn học, Tên môn, Tổng số tiết, Loại môn học**) – trong đó Loại môn học có thể là: Đại cương, Cơ sở ngành, Chuyên ngành bắt buộc, Chuyên ngành tự chọn. Mã môn học là một số nguyên có 3 chữ số, tự động tăng.

Khai báo lớp **Người** (**Họ tên, Địa chỉ, Số ĐT**)

Khai báo lớp **Sinh viên đăng ký môn học** kế thừa từ lớp **Người** và có thêm (**mã sinh viên, lớp, số môn học đăng ký, danh sách các môn học đăng ký**), mã sinh viên là một số nguyên có 5 chữ số, tự động tăng.

Chương trình chính:

- Nhập vào danh sách (mảng) môn học có thể đăng ký
- Nhập danh sách (mảng) sinh viên đăng ký môn học.
- Sắp xếp mảng sinh viên theo thứ tự alphabet. Hiển thị danh sách đã sắp ra màn hình.

Bài 9: Quản lý bán hàng trong siêu thị

Khai báo lớp **Mặt hàng** gồm các thuộc tính (**mã hàng, Tên hàng, Nhóm hàng, Giá bán**) – trong đó Nhóm hàng có thể là: Hàng thời trang, Hàng tiêu dùng, Hàng điện máy, Hàng gia dụng. Mã hàng là một số nguyên có 4 chữ số, tự động tăng.

Khai báo lớp **Người** (**Họ tên, Địa chỉ, Số ĐT**)

Khai báo lớp **Khách hàng** kế thừa từ lớp **Người** và có thêm (**mã KH, loại KH, số mặt hàng đã mua, danh sách (mảng) các mặt hàng đã mua**), mã KH là một số nguyên có 5 chữ số, tự động tăng. Loại KH có thể là: Mua lẻ hoặc Mua buôn.

Chương trình chính:

- Nhập vào danh sách các mặt hàng đang bán trong siêu thị.
- Nhập danh sách khách hàng.
- Nhập vào 1 mã khách hàng, tìm kiếm xem khách hàng có mã tương ứng đã được lưu trong danh sách hay chưa?
 - Nếu tìm thấy hiển thị thông tin đầy đủ của khách hàng đó ra màn hình.
 - Nếu không thấy, yêu cầu người dùng nhập đầy đủ thông tin của khách hàng và lưu vào danh sách khách hàng đã có. Hiển thị lại danh sách sau khi thêm ra màn hình.

Bài 10: Quản lý nhân sự

Khai báo lớp **Phòng ban** (**mã phòng, tên phòng, mô tả, hệ số công việc**), mã phòng là một số nguyên có 3 chữ số, tự động tăng. Hệ số công việc là một số thực từ trong khoảng từ 1 đến 20, cho biết mức đánh giá chung về các công việc thuộc phòng đó đảm nhiệm.

Khai báo lớp **Nhân viên** (**mã nhân viên, mã phòng, tên nhân viên, bậc lương, số ngày công**) là bạn của lớp Phòng ban– trong đó Bậc lương là một số nguyên từ 1 đến 9. Mã nhân viên là một số nguyên có 4 chữ số, tự động tăng.

Chương trình chính:

- Nhập vào danh sách (mảng) các phòng ban.
- Nhập danh sách (mảng) các nhân viên.
- Hiển thị danh sách nhân viên đã nhập ra màn hình
- Nhập vào 1 mã nhân viên bất kỳ, kiểm tra xem nhân viên có mã tương ứng đã được lưu trong danh sách hay chưa?

+ Nếu tìm thấy, hiển thị ra màn hình các thông tin sau: Mã nhân viên, tên nhân viên, tên phòng, hệ số công việc, số ngày công.

Bài 11: Quản lý chia tiền thưởng cho cầu thủ bóng đá

Khai báo lớp **Vị trí thi đấu** (*mã vị trí, Tên vị trí, mức thưởng*), mã vị trí là một số nguyên có 3 chữ số, tự động tăng.

Khai báo lớp **Người** (*Họ tên, Địa chỉ, Số ĐT*)

Khai báo lớp **Cầu thủ** kế thừa từ lớp **Người** và bổ sung các thuộc tính (*mã cầu thủ, Mức lương, mã vị trí thi đấu*) – trong đó mức lương là giá trị đã ký theo hợp đồng. Mã cầu thủ là một số nguyên có 3 chữ số, tự động tăng. Lớp cầu thủ là bạn của lớp Vị trí thi đấu.

Chương trình chính:

- Nhập danh sách (mảng) các vị trí thi đấu trong đội bóng.
- Nhập danh sách cầu (mảng) cầu thủ. Hiển thị danh sách đã nhập ra màn hình
- Nhập vào 1 mã cầu thủ bất kỳ, kiểm tra xem cầu thủ có mã tương ứng đã được lưu trong danh sách hay chưa? Nếu tìm thấy, hiển thị ra màn hình mức thưởng của cầu thủ đó.

Bài 12: Quản lý dịch vụ viễn thông

Khai báo lớp **Dịch vụ** (*mã DV, tên DV, giá cước, đơn vị tính*) mã DV là một số nguyên có 3 chữ số, tự động tăng.

Khai báo lớp **Người** (*Họ tên, Địa chỉ, Số ĐT*)

Khai báo lớp **Khách hàng** kế thừa từ lớp **Người** và bổ sung các thuộc tính (*mã KH, Loại KH, số dịch vụ sử dụng, danh sách các mã dịch vụ sử dụng*) – với Loại khách hàng có thể là: cá nhân, đại diện đơn vị hành chính, đại diện đơn vị kinh doanh, mã KH là một số nguyên có 5 chữ số, tự động tăng.

Chương trình chính:

- Nhập vào danh sách các dịch vụ hiện có.
- Nhập danh sách (mảng) khách hàng. Hiển thị danh sách đã nhập ra màn hình.
- Sắp xếp danh sách khách hàng theo thứ tự tổng giá cước dịch vụ giảm dần.
(tổng giá cước = \sum (dịch vụ sử dụng * giá cước). Hiển thị danh sách đã sắp ra màn hình.

Bài 13: Quản lý điểm sinh viên

Khai báo lớp **Sinh viên** (*mã SV, Tên SV, Lớp*). Trong đó mã sinh viên là một số nguyên có 5 chữ số, tự động tăng. Lớp được chọn từ danh sách các lớp theo cách đặt tên tại HV CN BCVT.

Khai báo lớp **Môn học** (*mã môn, tên môn, số đvht, loại môn*). Trong đó, mã môn là một số nguyên có 3 chữ số, tự động tăng. Loại môn học có thể là Đại cương, Cơ sở ngành, Chuyên ngành.

Khai báo lớp **Bảng điểm** (*mã sinh viên, mã môn học, điểm môn học*) là **bạn** của lớp **Sinh viên** và lớp **Môn học** trong đó một sinh viên có một điểm cho mỗi môn học ($0 \leq \text{điểm} \leq 10$).

Chương trình chính:

- Nhập vào 1 danh sách (mảng) môn học.
- Nhập vào 1 danh sách sinh viên.
- Nhập bảng điểm cho n sinh viên.

Nhập vào 1 mã sinh viên bất kỳ, liệt kê tất cả các môn học và điểm từng môn học của sinh viên đó ra màn hình.