

BÀI TẬP THỰC HÀNH NNLT C++

MỤC TIÊU 3: CLASS

Bài tập 1: Xây dựng lớp Car (xe hơi) có các thành viên sau:

- Các thuộc tính public:
 - o brand: nhãn hiệu, kiểu string.
 - o weight: khối lượng xe lúc không tải.
 - o seats: số chỗ ngồi, kiểu int.
- Các phương thức:
 - o show(): in các thuộc tính ra màn hình.
 - o hàm khởi tạo không tham số thực hiện gán brand="N/A", weight= 0, seat= 0 và load= 0

Viết hàm main() khai báo 3 đối tượng car1, car2, car3 kiểu Car, nhập dữ liệu cho 3 đối tượng đó, và gọi hàm show() để in thông tin 3 xe đó ra màn hình.

Bài tập 2:

a. Để bảo vệ dữ liệu, hãy thay đổi tính chất của các thành viên, từ public thành private. Sau đó chạy thử chương trình.

b. Sẽ gặp lỗi "Car::Car() is private". Đó là do **hàm khởi tạo không được có tính chất private**. Hãy đổi tính chất của hàm khởi tạo Car::Car() về public. Chạy thử chương trình.

c. Sẽ gặp lỗi "Car::brand is private". Lý do là trong hàm main() có lệnh **truy cập đến một thuộc tính private của object**. Giải quyết bằng cách thêm vào class Car các hàm public đọc và ghi thuộc tính private (thường gọi là các hàm set/get). Từ bên ngoài (tức trong hàm main()), chỉ có thể truy cập vào các thuộc tính private của object thông qua các hàm set/get này.

```
public:
    string getBrand() {
        return brand;
    }
    void setWeight(int w) {
        weight = w;
    }
    int getWeight() {
        return weight;
    }
    void setSeats(int s) {
        seats = s;
    }
    int getSeats() {
        return seats;
    }
}
```

Trong hàm main():

`cin.ignore(265, '\n');` //Xóa bộ đệm bàn phím, xóa 256 ký tự hoặc xóa đến khi gặp ký tự xuống dòng. Thao tác này giúp làm sạch bộ đệm trước khi nhập dữ liệu từ bàn phím.

```
cout <<"Brand: "; getline(cin, strtmp); car1.setBrand(strtmp);  
cout <<"Weight: "; cin >>inttmp; car1.setWeight(inttmp);  
cout <<"Seats: "; cin >>inttmp; car1.setSeats(inttmp);
```

Tương tự như trên, hàm `show()` cũng phải được chuyển thành `public`.

Bài tập 3: Một công ty du lịch có nhiều xe dùng để chở khách, để quản lý được nhiều xe, ta phải dùng mảng. Viết chương trình như bài 2, nhưng dùng mảng để quản lý nhiều xe chứ không khai báo 3 xe rời rạc. Số lượng xe được nhập khi chạy chương trình.

Chú ý: Sử dụng con trỏ để thao tác mảng.

Bài tập 4: Công ty muốn quản lý số lượng du khách đi trên xe và mã số của từng du khách.

Giải pháp: Thêm các thuộc tính `private`:

- `travelers` kiểu `int`: số lượng du khách
 - `travelers_list`: Mảng chứa các Id của hành khách, Id có dạng hỗn hợp chữ và số, ví dụ "KH001"
- ```
string* travelers_list;
```

Thêm các hàm `public`:

- `setTravelers()`: Tham số là số `int` chỉ số lượng hành khách. Trong hàm này ta có thể thêm vào lệnh nhập chi tiết các mã số du khách
- `getTravelers()`: trả về `travelers`
- `showTravelers_List()`: liệt kê danh sách du khách trên xe

**Chú ý:**

- Vì thuộc tính `travelers_List` (danh sách du khách) là con trỏ, nên trước khi nhập chi tiết các du khách, ta phải xin bộ nhớ cho nó.
- Khi thuộc tính của class là con trỏ, nếu ta xin bộ nhớ cho nó thì ta phải tự xóa bộ nhớ của nó. Nhưng, lệnh xóa đối tượng sẽ không xóa bộ nhớ cho các thuộc tính của đối tượng. Cho nên ta phải chủ động xóa loại bộ nhớ này. Ta sẽ xây dựng hàm hủy (destructor) tường minh để làm điều này.

```
Car::~~Car() {
 delete [] travelers_list;
}
```

**Bài tập 5:** Xây dựng lớp `Thisinh` (Thí Sinh) gồm các thành viên sau:

- Các thuộc tính `private`:
  - o `hoten`: Họ và tên, kiểu `string`
  - o `somonthi`: Số môn thi, kiểu `int`

- diem: Con trỏ chỉ đến mảng chứa điểm các môn, mỗi phần tử có kiểu float
- trungbinh: Điểm trung bình cộng các môn, kiểu float
- Các phương thức public:
  - Các hàm set/get cho các thuộc tính private. Hàm set cho số môn thi cho phép nhập lần lượt điểm của tất cả các môn.
  - Hàm tinh\_tb(): Tính điểm trung bình (trungbinh).
  - Hàm show(): Hiện thị thông tin thí sinh, có liệt kê chi tiết điểm các môn.
  - Hàm khởi tạo không tham số khởi tạo hoten= "N/A".
  - Hàm hủy gọi lệnh trả bộ nhớ của mảng diem.

Dùng lớp Thisinh, viết chương trình nhập một số n. Sau đó khởi tạo và nhập vào một mảng n đối tượng ts (Thí Sinh). In ra danh sách các thí sinh trúng tuyển biết rằng thí sinh trúng tuyển có điểm trung bình  $\geq 5$  và không có môn nào bị 0 điểm.

**Bài tập 6:** Xây dựng lớp hìnhchunhat (hình chữ nhật) có các thành viên sau:

- Các thuộc tính private:
  - dai, rong: Chiều dài và chiều rộng, kiểu int.
- Các phương thức public:
  - Các hàm set/get cho các thuộc tính private.
  - Hàm dientich(): Tính và in ra diện tích hình chữ nhật
  - Hàm chuvi(): Tính và in ra chu vi hình chữ nhật
  - Hàm khởi tạo không tham số gán dai= -1, rong= -1.

Viết chương trình sử dụng lớp hìnhchunhat: Khai báo, nhập các thuộc tính, in ra diện tích và chu vi hình chữ nhật.

**Bài tập 7:** Xây dựng lớp ptb2 (phương trình bậc 2) có các thành viên sau:

- Các thuộc tính private:
  - a, b, c: Ba hệ số của phương trình bậc 2, kiểu float.
- Các phương thức public:
  - Các hàm set/get cho các thuộc tính private.
  - Hàm giaipt(): Giải phương trình và in ra kết quả.
  - Hàm khởi tạo không tham số gán a= 0, b= 0, c= 0.

Viết chương trình sử dụng lớp ptb2: Khai báo, nhập các hệ số, giải phương trình và in ra kết quả.

**Bài tập 8:** Xây dựng lớp diem (điểm trong mặt phẳng) có các thành viên sau:

- Các thuộc tính private:
  - name: Tên điểm, kiểu string;
  - x, y: Hoành độ và tung độ của điểm, kiểu int.
- Các phương thức public:
  - Các hàm set/get cho các thuộc tính private.

- Hàm `distance()`: Tính khoảng cách giữa điểm một điểm bất kỳ và điểm đang xét, tham số của hàm này cũng là một đối tượng của lớp `diem`. Hàm này trả về kiểu `float`.
- Hàm `show()` hiển thị thông tin điểm.
- Hàm khởi tạo không tham số gán  $x = 0, y = 0$ .

Viết chương trình sử dụng lớp `diem`: Cho phép nhập vào số  $n$ , khởi tạo mảng  $n$  điểm. Tìm điểm trọng tâm, biết rằng **điểm trọng tâm là điểm có tổng khoảng cách đến tất cả các điểm khác là nhỏ nhất**.