

Chương 01.7: Kiểu cấu trúc và kiểu liệt kê

I. Kiểu cấu trúc (struct)

II. Kiểu liệt kê (enum)

I. Kiểu cấu trúc

1. Khái niệm về kiểu cấu trúc
2. Khai báo kiểu cấu trúc
3. Khai báo biến cấu trúc
4. Truy nhập các thành phần của cấu trúc
5. Khởi tạo biến cấu trúc
6. Phép gán biến cấu trúc
7. Mảng cấu trúc

1. Khái niệm về kiểu cấu trúc

- ✧ Ngoài các kiểu dữ liệu có sẵn trong C++ người lập trình còn có thể tạo ra những kiểu dữ liệu của riêng mình. Trong chương này chúng ta nghiên cứu hai kiểu hay dùng là kiểu cấu trúc và kiểu liệt kê.
- ✧ Một cấu trúc là một nhóm các phần tử có thể có kiểu dữ liệu khác nhau. Các phần tử này gọi là các thành phần của cấu trúc. Kiểu cấu trúc trong C++ tương đương với kiểu bản ghi trong Pascal.

2. Khai báo kiểu cấu trúc

✧ Khai báo cấu trúc là mô tả về các thành phần của cấu trúc. Cú pháp như sau:

```

    Từ khoá
    ↙
struct Tên_kiểu_cấu_trúc
{
    Kiểu_1 Tên_thành_phần_1;
    Kiểu_2 Tên_thành_phần_2;
    . . . .
};
    ← Dấu chấm phẩy kết thúc
    khai báo kiểu cấu trúc

```

Các thành phần của cấu trúc

2. Khai báo kiểu cấu trúc (tiếp)

- ✧ Ví dụ: Để lưu trữ thông tin về nhân sự của phòng tổ chức với các thông tin về họ tên, ngày sinh, địa chỉ, lương ta khai báo một kiểu cấu trúc như sau:

```
struct nhansu
{
    char hoten[30];
    char ngaysinh[10];
    char diachi[40];
    float luong;
};
```

2. Khai báo kiểu cấu trúc (tiếp)

- ✧ Sau khi khai báo kiểu cấu trúc ta có thể dùng tên kiểu cấu trúc như tên các kiểu dữ liệu cơ bản.
- ✧ Kiểu của các thành phần của cấu trúc có thể là kiểu cấu trúc, tức là trong cấu trúc có thể chứa cấu trúc khác. Ví dụ:

```
struct ngaythang
```

```
{  
    int ngay,thang,nam;  
};
```

```
struct nhansu
```

```
{  
    char hoten[30];  
    ngaythang ngaysinh;  
    char diachi[40];  
    float luong;  
};
```

3. Khai báo biến cấu trúc

- ✧ Việc khai báo kiểu cấu trúc không tạo ra vùng nhớ chứa cấu trúc mà chỉ mô tả về cấu trúc xem có những gì.
- ✧ Muốn có vùng nhớ chứa cấu trúc ta phải khai báo biến cấu trúc. Cú pháp:
Tên_kiểu_cấu_trúc Tên_biến_cấu_trúc;

Ví dụ:

nhansu ng1,ng2;

4. Truy nhập các thành phần cấu trúc

✧ Để truy nhập các thành phần của cấu trúc ta dùng toán tử chấm. Cú pháp:

Tên_biến_cấu_trúc.Tên_thành_phần

Ví dụ:

```
struct thisinh
```

```
{
```

```
    char SBD[15];
```

```
    float toan,ly,hoa;
```

```
};
```

```
//Khai bao bien cau truc
```

```
thisinh ts;
```

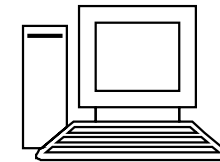
```
//Nhap du lieu cho thi sinh
```

```
cout<<"So bao danh: "; cin>>ts.SBD;
```

```
cout<<"Diem Toan: "; cin>>ts.toan;
```

```
cout<<"Diem Ly: "; cin>>ts.ly;
```

```
cout<<"Diem Hoa: "; cin>>ts.hoa;
```



5. Khởi tạo biến cấu trúc

- ✧ Khi khai báo biến cấu trúc ta có thể khởi tạo giá trị cho các thành phần của cấu trúc như khởi tạo cho các phần tử của mảng.

Ví dụ:

```
//Khai bao kieu cau truc
struct thisinh
{
    char SBD[15];
    float toan,ly,hoa;
};
//Khai bao va khoi tao bien cau truc
thisinh ts={"NNHA23456", 7, 8, 9};
```

6. Phép gán biến cấu trúc

✧ Ta có thể gán một biến cấu trúc cho một biến cấu trúc cùng kiểu. Ví dụ:

```
//Khai bao kieu cau truc
struct thisinh
{
    char SBD[15];
    float toan,ly,hoa;
};
//Khai bao bien cau truc
thisinh ts1={"NNHA23456",7,8,9};
thisinh ts2;
ts2=ts1;
```

7. Mảng cấu trúc

- ✧ Sau khi khai báo kiểu cấu trúc thì tên kiểu cấu trúc được dùng như các kiểu dữ liệu khác. Chẳng hạn, dùng cấu trúc làm kiểu phần tử của mảng.

Ví dụ:

```
//Khai bao kieu cau truc
struct thisinh
{
    char SBD[15];
    float toan,ly,hoa;
};
//Khai bao bien cau truc
thisinh ds[100];
strcpy(ds[0].SBD,"NNHA23456");
ds[0].toan=8;
ds[0].ly=8;
ds[0].hoa=9;
```

II. Kiểu liệt kê

- ✧ Kiểu liệt kê là kiểu dữ liệu do người lập trình tự định nghĩa bằng cách liệt kê tất cả các giá trị. Các giá trị của kiểu liệt kê là các tên tự đặt.
- ✧ Để định nghĩa kiểu liệt kê ta dùng từ khóa enum theo cú pháp sau:

```
enum Tên_kiểu_liệt_kê {Danh sách các tên tự đặt};
```

Ví dụ: enum boolean {TRUE, FALSE};

```
enum mausac {Xanh, Do, Tim, Vang};
```

```
enum days_of_week {Sun, Mon, Tue, Wed, Thu, Fri, Sat};
```

II. Kiểu liệt kê (tiếp)

✧ Sau khi khai báo kiểu liệt kê ta có thể khai báo các biến kiểu liệt kê như các biến kiểu khác:

Tên_kiểu_liệt_kê Danh_sách_các_biến;

Ví dụ: Giả sử các kiểu liệt kê đã được khai báo ở trên, ta khai báo các biến liệt kê:

days_of_week day1, day2;

✧ Để đưa giá trị vào biến liệt kê ta dùng lệnh gán:

Ví dụ: day1 = Mon; day2 = Sat;

✧ Ta không dùng được lệnh cout và cin với các biến kiểu liệt kê.

II. Kiểu liệt kê (tiếp)

✧ Các giá trị kiểu liệt kê được lưu trữ như các số nguyên kiểu int, giá trị tên đầu tiên là 0, giá trị tên tiếp theo là 1,...

Ví dụ: Với kiểu liệt kê `days_of_week` ở trên thì Sun có giá trị 0, Mon có giá trị 1, Tue có giá trị 3,...

✧ Ta có thể thay đổi giá trị số của các giá trị tên

- Cho các giá trị tên có giá trị số bắt đầu từ một số khác 0

Ví dụ: `enum mausac {Xanh=5, Do, Tim, Vang};`

Với khai báo này Xanh có giá trị 5, Do có giá trị 6, Tim có giá trị 7, Vàng có giá trị 8.

- Cho

Bài tập

Viết chương trình nhập vào một danh sách có n sinh viên, mỗi sinh viên có các thông tin về họ tên, lớp, điểm TBC. Sắp xếp danh sách sinh viên theo điểm TBC giảm dần.