

KỸ THUẬT LẬP TRÌNH

Chương 6. Kiểu cấu trúc

thuong.bui@ut.edu.vn

thuong.bui@ut.edu.vn



6.1 Khai báo và khởi tạo

Khai báo cấu trúc

```
struct Tên_Cấu_trúc {  
    Kiểu Thành_phần_1;  
    Kiểu Thành_phần_2;  
    ...  
} Danh_sách_biến;
```

Ví dụ

```
struct Date {  
    int day;  
    int month;  
    int year;  
} day1, day2 = {1, 5, 2025};
```

- *Tên cấu trúc: được đặt theo quy tắc định danh*
- *Mỗi thành phần giống như một biến riêng của kiểu, nó gồm kiểu và tên thành phần.*
- *Phần tên của kiểu cấu trúc và phần danh sách biến có thể có hoặc không.*
- *Các kiểu cấu trúc được phép khai báo lồng nhau.*
- *Một biến có kiểu cấu trúc sẽ được phân bố bộ nhớ sao cho các thực hiện của nó được sắp liên tục theo thứ tự xuất hiện trong khai báo.*

6.1 Khai báo và khởi tạo

Khai báo biến cấu trúc

struct Tên cấu trúc Danh sách biến;

Hoặc:

Tên cấu trúc Danh sách biến ;

khai báo kèm khởi tạo:

Tên cấu trúc Biến = { giá trị khởi tạo};

Ví dụ

```
struct Date {  
    int day;  
    int month;  
    int year;  
};
```

```
struct Student {  
    char name[30];  
    Date birthday;  
    float score;  
};
```

```
Date day3, holiday = {1,2,2025};
```

```
Student s1 = {"Nguyen Van A", {12,8,2006}, 8.5};
```

6.2 Truy cập vào các thành phần

- Đối với biến thường:
Tên_biến.Tên_thành_phần
- Đối với biến con trỏ :
Tên_biến -> Tên_thành_phần
- Đối với biến mảng: Truy nhập thành phần mảng rồi đến thành phần cấu trúc.
Tên_mảng[chỉ số].Tên_thành_phần
- Đối với cấu trúc lồng nhau: Truy nhập thành phần ngoài rồi đến thành phần của cấu trúc bên trong, sử dụng các phép toán **.** hoặc **->** một cách thích hợp.

Ví dụ

```
struct Date {
    int day;
    int month;
    int year;
};

struct Student {
    char name[30];
    Date birthday;
    float score;
};

Student s1 = {"Nguyen Van A", {1,1,2000}, 8.5};
cout<<"Thong tin sinh vien:\n";
cout<<"Ho ten: "<<s1.name<<endl;
cout<<"Sinh nhat: "<<s1.birthday.day<<"/";
cout<<s1.birthday.month<<"/"<<s1.birthday.year<<endl;
cout<<"Diem: "<<s1.score<<endl;
```

```
Thong tin sinh vien:
Ho ten: Nguyen Van A
Sinh nhat: 1/1/2000
Diem: 8.5
```

6.3 Con trỏ với cấu trúc

Một con trỏ cấu trúc cũng giống như con trỏ trỏ đến các kiểu dữ liệu khác, có nghĩa nó chứa địa chỉ của một biến cấu trúc hoặc một vùng nhớ có kiểu cấu trúc nào đó.

Truy nhập thuộc tính của con trỏ cấu trúc

Cách 1:

Tên biến con trỏ -> Tên thuộc tính;

Cách 2:

(*Tên biến con trỏ). Tên thuộc tính;

Ví dụ

```
struct Date {
    int day;
    int month;
    int year;
};
struct Student {
    char name[30];
    Date birthday;
    float score;
};
Student s1 = {"Nguyen Van A", {1,1,2000}, 8.5};
Student *s2 = &s1;
cout<<"Thong tin sinh vien:\n";
cout<<"Ho ten: "<<s2->name<<endl;
cout<<"Sinh nhat: "<<s2->birthday.day<<"/";
cout<<(*s2).birthday.month<<"/";
cout<<(*s2).birthday.year<<endl;
cout<<"Diem: "<<(*s2).score<<endl;
```

6.3 Con trỏ với cấu trúc

Cấp phát bộ nhớ động:

Tên biến con trỏ = new Kiểu cấu trúc;

Giải phóng vùng nhớ:

delete Tên biến con trỏ;

Ví dụ

```
Student *p = new Student;
cout << "Enter name: ";
cin.getline(p->name,30);
cout << "Enter birthday: ";
cin >> p->birthday.day;
cin >> p->birthday.month;
cin >> p->birthday.year;
cout << "Enter score: ";
cin >> p->score;
cout << "Name: " << p->name << endl;
cout << "Birthday: " << p->birthday.day;
cout << "/" << p->birthday.month;
cout << "/" << p->birthday.year << endl;
cout << "Score: " << p->score << endl;
delete p;
```

```
Enter name: Le Vu
Enter birthday: 12 5 1987
Enter score: 9
Name: Le Vu
Birthday: 12/5/1987
Score: 9
```

6.4 Mảng với cấu trúc

- Khai báo mảng tĩnh các cấu trúc:

Tên kiểu cấu trúc Tên biến mảng[Số phần tử];

- Khai báo mảng động các cấu trúc:

Tên kiểu cấu trúc *Tên biến;

- Cấp phát bộ nhớ:

Tên biến mảng = new Kiểu cấu trúc[Số lượng phần tử];

- Truy nhập đến phần tử của mảng cấu trúc:

Việc truy nhập đến các phần tử của mảng cấu trúc được thực hiện như truy cập đến phần tử của mảng thông thường.

```

#include<iostream>
#include<fstream>
using namespace std ;
struct product{
    int id;
    char name[20];
    float price;
};
int main(){
    int n; cout<<"Nhap so luong san pham: "; cin>>n;
    product *p=new product[n];
    for(int i=0;i<n;i++){
        cout<<"Nhap thông tin sản phẩm thứ "<<i+1<<endl;
        cout<<"Nhap id: "; cin>>p[i].id;
        cout<<"Nhap tên: "; cin.ignore(); cin.getline(p[i].name, 20);
        cout<<"Nhap giá: "; cin>>p[i].price;
    }
    cout<<"Thông tin các sản phẩm:"<<endl;
    cout<<"STT\tID\tTên\tGiá"<<endl;
    for(int i=0;i<n;i++){
        cout<<i+1<<"\t"; cout<<p[i].id<<"\t";
        cout<<p[i].name<<"\t"<<p[i].price<<endl;
    }
    delete[] p;
    system("pause");
}

```

Kết quả chạy chương trình

```

Nhap so luong san pham: 4
Nhap thông tin sản phẩm thứ 1
Nhap id: 21
Nhap tên: Du du
Nhap giá: 19000
Nhap thông tin sản phẩm thứ 2
Nhap id: 87
Nhap tên: Su su
Nhap giá: 12000
Nhap thông tin sản phẩm thứ 3
Nhap id: 48
Nhap tên: Tao do
Nhap giá: 44000
Nhap thông tin sản phẩm thứ 4
Nhap id: 59
Nhap tên: Xoai
Nhap giá: 38000
Thông tin các sản phẩm:
STT      ID      Tên      Giá
1         21      Du du    19000
2         87      Su su    12000
3         48      Tao do   44000
4         59      Xoai     38000

```


6.5 Hàm với cấu trúc

Đối của hàm là cấu trúc:

Một cấu trúc có thể được sử dụng để làm đối của hàm dưới các dạng sau đây:

- ❖ *Là một biến cấu trúc, khi đó tham đối thực sự là một cấu trúc.*
- ❖ *Là một con trỏ cấu trúc, tham đối thực sự là địa chỉ của một cấu trúc.*
- ❖ *Là một tham chiếu cấu trúc, tham đối thực sự là một cấu trúc.*
- ❖ *Là một mảng cấu trúc hình thức hoặc con trỏ mảng, tham đối thực sự là tên mảng cấu trúc.*

Giá trị hàm là cấu trúc:

Cũng tương tự như các kiểu dữ liệu cơ bản, giá trị trả lại của một hàm cũng có thể là các cấu trúc dưới các dạng sau:

- ❖ *Là một biến cấu trúc.*
- ❖ *Là một con trỏ cấu trúc.*
- ❖ *Là một tham chiếu cấu trúc.*

```

#include<iostream>
#include<fstream>
using namespace std ;
struct product{
    int id;
    char name[20];
    float price;
};
void nhap(product &p){
    cout<<"Nhap id: "; cin>>p.id;
    cout<<"Nhap ten: "; cin.ignore(); cin.getline(p.name,20);
    cout<<"Nhap gia: "; cin>>p.price;
}
product mostExpensive(product *p, int size){
    product max=p[0];
    for(int i=1;i<size;i++){
        if(p[i].price>max.price) max=p[i];
    }
    return max;
}
int main(){
    int n; cout<<"Nhap so luong san pham: "; cin>>n;
    product *p=new product[n];
    for(int i=0;i<n;i++){
        cout<<"Nhap thông tin sản phẩm thứ "<<i+1<<endl;
        nhap(p[i]);
    }
    cout<<"Thông tin sản phẩm có giá đắt nhất: "<<endl;
    product maxPrice = mostExpensive(p,n);
    cout<<"ID\tTen\tGia"<<endl;
    cout<<maxPrice.id<<"\t"<<maxPrice.name<<"\t"<<maxPrice.price<<endl;
    delete[] p;
    system("pause");
}

```

Kết quả chạy chương trình

```

Nhap so luong san pham: 3
Nhap thông tin sản phẩm thứ 1
Nhap id: 32
Nhap ten: Xoai
Nhap gia: 22000
Nhap thông tin sản phẩm thứ 2
Nhap id: 97
Nhap ten: Su Su
Nhap gia: 18000
Nhap thông tin sản phẩm thứ 3
Nhap id: 87
Nhap ten: Du Du
Nhap gia: 30000
Thông tin sản phẩm có giá đắt nhất:
ID      Ten      Gia
87      Du Du     30000

```

6.6 Câu lệnh typedef và hàm sizeof

- **Hàm sizeof()** trả lại kích thước của một biến hoặc kiểu.
- Lệnh **typedef** dùng để đặt lại tên kiểu dữ liệu được người sử dụng tạo mới:

typedef kiểu tên_kiểu;

Ví dụ

```
#include <iostream>
using namespace std;
int main() {
    typedef double sothuc;
    sothuc x;
    struct date{
        int ngay, thang, nam;
    } y;
    cout<< "Kich thuoc cua kieu sothuc double la: ";
    cout<<sizeof(x)<< " byte"<<endl;
    cout<< "Kich thuoc cua kieu cau truc date la: ";
    cout<<sizeof(y)<< " byte"<<endl;
    cout<< "Kich thuoc cua kieu so nguyen int la: ";
    cout<<sizeof(y.nam)<< " byte"<<endl;
    system("pause");
}
```

```
Kich thuoc cua kieu sothuc double la: 8 byte
Kich thuoc cua kieu cau truc date la: 12 byte
Kich thuoc cua kieu so nguyen int la: 4 byte
```

BÀI TẬP CHƯƠNG 6

Bài 1. Khai báo 1 cấu trúc sinh viên bao gồm các thành phần: Họ tên, ngày sinh (là 1 cấu trúc gồm ngày, tháng, năm), mã số sinh viên (kiểu int), điểm thi.

- a) Nhập/xuất dữ liệu cho n sinh viên.
- b) Hiển thị thông tin của những sinh viên có điểm thi > 5 .
- c) Hiển thị thông tin của những sinh viên sinh ngày 13.
- d) Sắp xếp danh sách các sinh viên theo MSSV tăng dần
- e) Sắp xếp danh sách các sinh viên trên theo thứ tự abc...xyz của tên sinh viên (*Ghi chú: Họ tên là Trần Linh Miêu thì sắp xếp dựa vào tên là Miêu*).

Bài 2. Cho biết ngày 1/1/2000 là thứ bảy. Hãy viết chương trình nhập vào ngày, tháng và năm (kiểu cấu trúc) bất kì và cho biết chính xác ngày hôm đó là thứ mấy?

Bài 3. Viết chương trình quản lý danh sách nhân viên bằng mảng cấu trúc. Mỗi nhân viên bao gồm các thông tin: Họ tên, ngày sinh, ngày vào làm, giới tính (dùng kiểu bool với true là nam và false là nữ), lương.

- a) Nhập xuất dữ liệu cho n nhân viên.
- b) Tìm nhân viên có thâm niên lâu nhất (thời gian vào làm lâu nhất).
- c) Tính tổng tiền lương cần thanh toán cho toàn bộ nhân viên.
- d) Hiển thị danh sách nhân viên Nam có tuổi < 30 .