

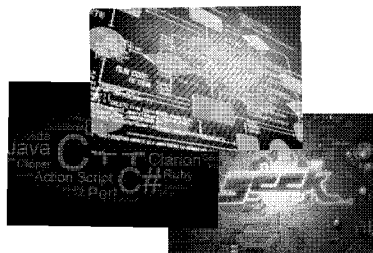


ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

NHẬP MÔN LẬP TRÌNH

CHƯƠNG VI

MẢNG VÀ CẤU TRÚC



Nguyễn Trọng Chính
chinhnt@uit.edu.vn

MẢNG VÀ CẤU TRÚC

- ❖ MẢNG CỐ ĐỊNH
- ❖ MỘT SỐ VẤN ĐỀ LẬP TRÌNH VỚI MẢNG
- ❖ CHUỖI KÝ TỰ
- ❖ DỮ LIỆU CÓ CẤU TRÚC

DỮ LIỆU CÓ CẤU TRÚC

❖CẤU TRÚC

Cấu trúc là một kiểu dữ liệu kết hợp từ các kiểu dữ liệu nguyên thủy do người lập trình định nghĩa để thuận tiện trong việc quản lý dữ liệu và lập trình.

Cấu trúc được khai báo như sau:

```
struct TênCấuTrúc {  
    Kiểu Thànhphần_1;  
    ....  
    Kiểu Thànhphần_n;  
};
```

Nếu Thànhphần_i là kiểu mảng thì thêm [số_ptử]

3

DỮ LIỆU CÓ CẤU TRÚC

❖CẤU TRÚC

Ví dụ

```
struct Hocsinh {  
    char hoten[31];  
    char socmnd[10];  
    int namsinh;  
};
```

4

DỮ LIỆU CÓ CẤU TRÚC

❖CẤU TRÚC

Cú pháp khai báo biến kiểu cấu trúc:

```
struct TênCấuTrúc TênBiến;
```

```
struct TênCấuTrúc TênMảng[số_pt];
```

Khởi tạo giá trị ban đầu cho biểu cấu trúc:

```
struct TênCấuTrúc TênBiến = {gtri_trường_1,  
...,  
gtri_trường_n};
```

Ví dụ:

```
struct Hocsinh nam;
```

```
struct Hocsinh lopA[30];
```

5

DỮ LIỆU CÓ CẤU TRÚC

❖CÁC PHÉP TOÁN TRÊN CẤU TRÚC

Phép gán “=”: Copy giá trị tương ứng từng thành phần giữa hai biến cấu trúc.

Ví dụ:

```
struct Hocsinh a={"Tran Nam", "2722113326", 1994},b;
```

```
b = a; //b={"Tran Nam", "2722113326", 1994}
```

Phép truy xuất thành phần “.”: Cho phép xử lý từng thành phần thông tin của một biến cấu trúc.

Ví dụ

```
printf(b.hoten); // Tran Nam
```

```
printf("%d", b.namsinh); // 1994
```

6

DỮ LIỆU CÓ CẤU TRÚC

❖ Ví dụ: Xây dựng cấu trúc hình tròn, nhập thông tin hai hình tròn và cho biết chúng có giao nhau không (chỉ xét hai đường tròn không chứa tâm của nhau).

7

DỮ LIỆU CÓ CẤU TRÚC

```
float sqr(float x) { return x * x;}  
void main() {  
    struct Hinhtron {  
        float x, y, r;  
    };  
    struct Hinhtron o1, o2;  
    scanf("%f %f %f", &o1.x, &o1.y, &o1.r);  
    scanf("%f %f %f", &o2.x, &o2.y, &o2.r);  
    if (sqr(o1.x - o2.x) + sqr(o1.y - o2.y) <= sqr(o1.r + o2.r))  
        printf("giao nhau");  
    else  
        printf("khong giao nhau");  
}
```

8

DỮ LIỆU CÓ CẤU TRÚC

❖TRƯỜNG BIT

Dùng để chỉ định số lượng bit của một thành phần. Trường bit được bắt đầu bằng dấu ":" và số nguyên chỉ số lượng bit của thành phần ngay sau tên thành phần. Trường bit chỉ áp dụng cho kiểu số nguyên.

Ví dụ:

```
struct Byte {  
    unsigned char b7:1, b6:1, b5:1, b4:1, b3:1, b2:1,  
    b1:1, b0:1;  
}
```

9

DỮ LIỆU CÓ CẤU TRÚC

❖Khai báo kiểu

Khai báo kiểu nhằm thu gọn cách khai báo biến. Khai báo kiểu được thực hiện nhờ cú pháp sau:

typedef Kiểu TênKiểuMới;

Ví dụ:

```
typedef char Hoten[31];  
typedef struct _Hocsinh {  
    Hoten hoten;  
    char socmnd[11];  
    int namsinh;  
} Hocsinh;
```

10

DỮ LIỆU CÓ CẤU TRÚC

❖ KHAI BÁO KIỂU

Cú pháp của C++ có thể khai báo kiểu struct trực tiếp như sau:

```
struct Hocsinh {  
    Hoten hoten;  
    char socmnd[11];  
    int namsinh;  
};
```

11

DỮ LIỆU CÓ CẤU TRÚC

❖ TẦM VỰC CỦA KIỂU DỮ LIỆU ĐƯỢC KHAI BÁO

Kiểu dữ liệu được khai báo trong một khối lệnh có tác dụng từ cuối dòng lệnh khai báo đến cuối khối lệnh chứa nó.

Nếu kiểu dữ liệu được khai báo không nằm trong một khối lệnh nào, thì phạm vi của nó bắt đầu từ cuối dòng lệnh khai báo nó.

12

DỮ LIỆU CÓ CẤU TRÚC

❖CẤU TRÚC HỢP

Là kiểu dữ liệu tương tự như struct nhưng các thành phần cùng chia sẻ một vùng nhớ. Kích thước của kiểu cấu trúc hợp là kích thước của thành phần lớn nhất. Cấu trúc hợp được khai báo như sau:

```
union TênCấuTrúc {  
    Kiểu Thànhphần_1;  
    ...  
    Kiểu Thànhphần_n;  
};
```

13

DỮ LIỆU CÓ CẤU TRÚC

❖CẤU TRÚC HỢP

Ví dụ

```
Union Soquanly {  
    char cmnd[10];  
    char hochieu[15];  
};
```

14

DỮ LIỆU CÓ CẤU TRÚC

❖CẤU TRÚC HỢP

Cú pháp khai báo biến kiểu cấu trúc hợp:

union TênCấuTrúc TênBiến;

union TênCấuTrúc TênMảng[số_pt];

Khởi tạo giá trị ban đầu cho biến cấu trúc hợp phải có kích thước không quá kích thước thành phần đầu tiên:

union TênCấuTrúc TênBiến {giá_trị};

Ví dụ:

union Soquanly a = {"272163248"};

union Soquanly ds[10];

15

DỮ LIỆU CÓ CẤU TRÚC

❖PHÉP TOÁN TRÊN CẤU TRÚC HỢP

Cấu trúc hợp có phép gán và truy xuất thành phần của biến cấu trúc hợp như đối với cấu trúc.

16

DỮ LIỆU CÓ CẤU TRÚC

❖BÀI TẬP

- 1) Viết chương trình nhập vào một số nguyên từ trong đoạn $[0,255]$ và in ra biểu diễn nhị phân của nó.
- 2) Định nghĩa cấu trúc Diem gồm các thành phần: kiểm tra 1 tiết, kiểm tra 15 phút, kiểm tra miệng và trung bình. Viết chương trình nhập và in thông tin điểm trong đó trung bình = $(\text{điểm 1 tiết} * 2 + \text{điểm 15p} + \text{điểm miệng}) / 4$. Trong đó việc nhập và in thông tin được viết thành hàm.

17

DỮ LIỆU CÓ CẤU TRÚC

❖BÀI TẬP

- 3) Định nghĩa cấu trúc Hocsinh gồm các thành phần: họ tên, năm sinh, địa chỉ, điểm trung bình, điểm môn Toán, điểm môn Văn và điểm môn Anh văn. Viết chương trình nhập vào một danh sách n học sinh ($0 < n < 10$) gồm thông tin và điểm các môn, sắp xếp danh sách theo điểm trung bình giảm dần và in danh sách đó gồm các thông tin họ tên, năm sinh, điểm trung bình. Trong đó, điểm trung bình được tính bằng trung bình cộng của điểm trung bình các môn, điểm trung bình mỗi môn được tính như bài tập 2.

18

DỮ LIỆU CÓ CẤU TRÚC

❖BÀI TẬP

- 4) Định nghĩa cấu trúc DiemHK gồm các thành phần là điểm các môn toán, lý, hóa, văn, anh văn và điểm trung bình học kỳ. Điểm các môn có kiểu Diem như trong bài tập 2. Định nghĩa cấu trúc Hocsinh gồm họ tên, năm sinh, địa chỉ, lớp, điểm học kỳ 1, điểm học kỳ và tổng kết trong đó điểm học kỳ 1 và 2 có kiểu DiemHK. Viết chương trình nhập thông tin cho một lớp gồm n học sinh ($0 < n < 10$), tính điểm tổng kết, sắp xếp danh sách học sinh theo điểm tổng kết giảm dần và in ra danh sách đó gồm họ tên, năm sinh, điểm tổng kết.

19