



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TIN HỌC ĐẠI CƯƠNG

Phần 3. Lập trình C

Bài 10. Cấu trúc

Nội dung

10.1. Khái niệm cấu trúc

10.2. Khai báo và sử dụng cấu trúc

10.3. Xử lý dữ liệu cấu trúc

10.4. Mảng cấu trúc

Nội dung

10.1. Khái niệm cấu trúc

10.2. Khai báo và sử dụng cấu trúc

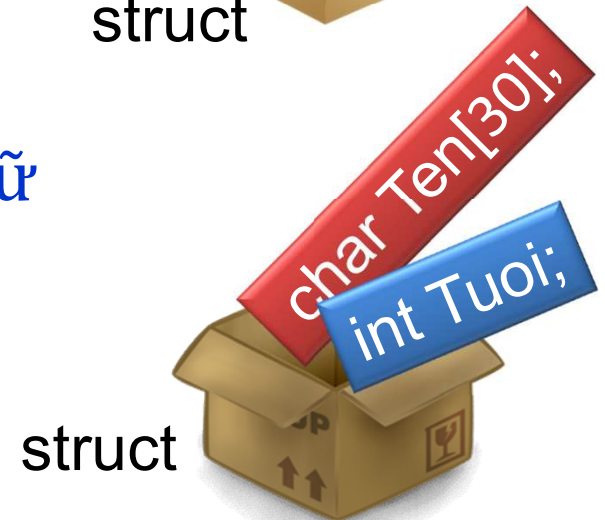
10.3. Xử lý dữ liệu cấu trúc

10.4. Mảng cấu trúc

10.1. Khái niệm cấu trúc

* Kiểu dữ liệu cấu trúc (**struct**)

- Là kiểu dữ liệu phức hợp, bao gồm **nhiều thành phần** có thể thuộc các kiểu dữ liệu khác nhau
- Các thành phần: gọi là trường dữ liệu (**field**)



* Ví dụ

- Thông tin về kết quả học tập của sinh viên:
TenSV, MaSV, Diem.
- Thông tin về cầu thủ: Ten, Tuoi, CLB, SoAo, Vitri,...

10.1. Khái niệm cấu trúc

DANH SÁCH THÍ SINH THAM DỰ THI VIẾT VÀO CÔNG TY TNHH DENSO VIỆT NAM

TT	Mã TD	Vị trí	Phòng thi	Họ tên	Ngày sinh	Giới tính	Trường	Chuyên ngành
1	MA-NE14	Nhân viên phòng Bảo dưỡng	HB	Trần Hoàng Long	23/01/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Điện Công Nghiệp & Bảo Dưỡng
2	MA-NE14	Nhân viên phòng Bảo dưỡng	HB	Vũ Văn Vĩnh	21/09/1991	Nam	Cao đẳng nghề Bách Khoa Hà Nội	điện tử viễn thông
3	PRO-NE14	Nhân viên phòng Sản xuất	HB	Nguyễn Văn Đức	25/07/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ Khí
4	PRO-NE14	Nhân viên phòng Sản xuất	HB	Nguyễn Trung Dũng	01/09/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Công nghệ ô tô.
5	PRO-NE14	Nhân viên phòng Sản xuất	HB	Vũ Minh Nhật	05/10/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Chế tạo máy
6	PRO-NE14	Nhân viên phòng Sản xuất	HB	Nguyễn Tiến Thắng	01/06/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Chế tạo máy
7	PRO-NE14	Nhân viên phòng Sản xuất	D5-406	Lê Chiều Dương	27/03/1985	Nam	Cao đẳng nghề Bách Khoa Hà Nội	
8	PRO-NE14	Nhân viên phòng Sản xuất	D5-406	Nguyễn Thế Hải	10/03/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	CƠ KHÍ CHẾ TẠO
9	PRO-NE14	Nhân viên phòng Sản xuất	D5-406	Nguyễn Đức Hải	03/06/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Điện công nghiệp
10	PRO-NE14	Nhân viên phòng Sản xuất	D5-406	Phạm Quang Khuê	26/09/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	cơ điện tử
11	PRO-NE14	Nhân viên phòng Sản xuất	D5-406	Ngô Đức Linh	12/12/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ Điện Tử
12	PRO-NE14	Nhân viên phòng Sản xuất	D5-406	Đặng Hải Long	29/09/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ Khí
13	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-503	Đặng Thanh Hoạch	08/02/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cắt gọt kim loại
14	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-504	Phạm Quang Khuê	26/09/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ điện tử
15	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-504	Ngô Đức Linh	12/12/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ điện tử
16	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-504	Phan Văn Lợi	15/07/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cắt Gọt Kim Loại (Chế Tạo Máy)
17	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-504	Lê Hoài Luân	06/11/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Điện công nghiệp
18	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-504	Hồ Sỹ Năm	12/05/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ Khí (Cơ Điện Tử)
19	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-504	Hồ Sỹ Năm	12/05/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ khí (cơ điện tử)
20	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-505	Đỗ Văn Thông	15/07/1991	Nam	Cao đẳng nghề Bách Khoa Hà Nội	CƠ KHÍ
21	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-505	Nguyễn Xuân Trường	26/07/1992	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ điện tử
22	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-505	Hoàng Đức Tú	14/12/1991	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Điện Công nghiệp
23	MA-NE14	Nhân viên phòng Bảo dưỡng	D5-505	Đỗ Đại Vương	29/03/1993	Nam	Cao đẳng nghề Bách Khoa Hà Nội	Cơ điện tử

Cột – Trường dữ liệu
Dòng – Bản ghi

10.2. Khai báo và sử dụng cấu trúc

10.2.1. Khai báo kiểu dữ liệu cấu trúc

10.2.2. Khai báo biến cấu trúc

10.2.3. Định nghĩa kiểu dữ liệu với typedef

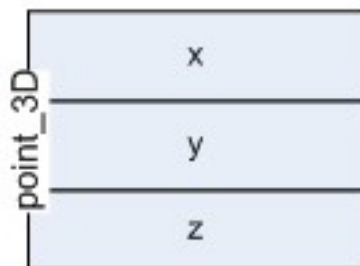
10.2.1. Khai báo kiểu dữ liệu cấu trúc

- Khai báo cấu trúc**

```
struct tên_cấu_trúc {  
    <khai báo các trường dữ liệu>  
};
```



struct sinh_vien



struct point_3D

- Ví dụ**

```
struct sinh_vien {  
    char ma_so[10];  
    char ho_va_ten[30];  
    float diem_tinDC;  
};
```

```
struct point_3D {  
    float x;  
    float y;  
    float z;  
};
```


10.2.2. Khai báo biến cấu trúc

- Cú pháp:

```
struct tên_cấu_trúc tên_biến_cấu_trúc;
```

- Ví dụ:

```
struct sinh_vien a, b, c;
```

- Kết hợp khai báo

```
struct [tên_cấu_trúc] {  
    <khai báo các trường dữ liệu>;  
} tên_biến_cấu_trúc;
```



10.2.2. Khai báo biến cấu trúc

- Các cấu trúc có thể được khai báo lồng nhau

```
struct diem_thi {  
    float dToan, dLy, dHoa;  
};
```

```
struct thi_sinh {  
    char ma_so[10];  
    char ho_va_ten[30];  
    struct diem_thi ket_qua;  
} thi_sinh_1, thi_sinh_2;
```



10.2.2. Khai báo biến cấu trúc

- Có thể khai báo trực tiếp các trường dữ liệu của một cấu trúc bên trong cấu trúc khác

```
struct thi_sinh {  
    char SBD[10];  
    char ho_va_ten[30];  
    struct diem_thi {  
        float dToan, dLy, dHoa;  
    } ket_qua;  
} thi_sinh_1, thi_sinh_2;
```

10.2.3. Định nghĩa kiểu dữ liệu với typedef

- Mục đích
 - Đặt **tên mới** cho **kiểu dữ liệu** cấu trúc
 - Giúp khai báo biến “quen thuộc” và ít sai hơn
- Cú pháp

```
typedef [struct tên_cấu_trúc] [tên_mới];
```

hoặc

```
typedef struct tên_cấu_trúc {  
    <khai báo các trường dữ liệu>;  
} danh_sách_các_tên_mới;
```
- Chú ý: cho phép đặt **tên_mới** trùng **tên_cấu_trúc**

10.2.3. Định nghĩa kiểu dữ liệu với typedef

- Ví dụ:

```
struct point_3D {  
    float x, y, z;  
};
```

```
struct point_3D M;
```

```
typedef struct point_3D  
point_3D;
```

```
point_3D N;
```

```
typedef struct {  
    float x, y, z;  
} point_3D;  
point_3D M;  
point_3D N;
```

10.2.3. Định nghĩa kiểu dữ liệu với typedef

- Ví dụ

```
typedef struct point_2D {  
    float x, y;  
} point_2D, diem_2_chieu, ten_bat_ki;  
point_2D X;  
diem_2_chieu Y;  
ten_bat_ki Z;
```

⇒ **point_2D, diem_2_chieu, ten_bat_ki** là các tên cấu trúc,
không phải tên biến

⇒ **X, Y, Z** là các biến cấu trúc

10.3. Xử lý dữ liệu cấu trúc

10.3.1. Truy cập các trường dữ liệu

10.3.2. Phép gán giữa các biến cấu trúc

10.3.1. Truy cập các trường dữ liệu

- Cú pháp
`tên_biến_cấu_trúc.tên_trường`
- Lưu ý
 - Dấu “.” là toán tử truy cập vào trường dữ liệu trong cấu trúc
 - Nếu trường dữ liệu là một cấu trúc => sử dụng tiếp dấu “.” để truy cập vào thành phần mức sâu hơn
 - Trường dữ liệu được sử dụng như các biến thông thường để nhập, xuất dữ liệu và các thao tác tính toán khác.

10.3.1. Truy cập các trường dữ liệu

Ví dụ:

Xây dựng một cấu trúc biểu diễn điểm trong không gian 2 chiều. Nhập giá trị cho một biến kiểu cấu trúc này, sau đó hiển thị giá trị các trường dữ liệu của biến này ra màn hình.

- Cấu trúc: tên điểm, tọa độ x, tọa độ y
- Nhập, hiển thị từng trường của biến cấu trúc như các biến dữ liệu khác

10.3.1. Truy cập các trường dữ liệu

```
#include <stdio.h>
#include <conio.h>
typedef struct {
    char ten[5];
    int x, y;
} toado;
void main() {
    toado t;
    printf("Nhap thong tin toa do\n");
```

10.3.1. Truy cập các trường dữ liệu

```
printf("Ten diem: ") ; gets(t.ten) ;  
printf("Toa do x: ") ; scanf("%d", &t.x) ;  
printf("Toa do y: ") ; scanf("%d", &t.y) ;  
printf("Gia tri cac truong\n") ;  
printf("%-5s%3d%3d\n", t.ten, t.x, t.y) ;  
getch() ;  
}
```

Bài tập

Lập trình nhập vào 2 điểm trong không gian 3 chiều.
Tính khoảng cách giữa 2 điểm đó.

10.3.2. Phép gán giữa các biến cấu trúc

- Muốn sao chép dữ liệu từ biến cấu trúc này sang biến cấu trúc khác cùng kiểu
 - gán lần lượt từng trường trong hai biến cấu trúc
=> “thủ công”
 - C cung cấp phép gán hai biến cấu trúc cùng kiểu:

`biến_cấu_trúc_1 = biến_cấu_trúc_2;`

10.3.2. Phép gán giữa các biến cấu trúc

- Ví dụ
 - Xây dựng cấu trúc gồm họ tên và điểm TĐC của sinh viên
 - a, b, c là 3 biến cấu trúc.
 - Nhập giá trị cho biến a.
 - Gán $b=a$, còn gán từng trường của a cho c.

10.3.2. Phép gán giữa các biến cấu trúc

```
#include <stdio.h>
#include <conio.h>
typedef struct {
    char hoten[20];
    int diem;
} sinhvien;
void main() {
    sinhvien a, b, c;
    printf("Nhap thong tin sinh vien\n");
    printf("Ho ten: "); gets(a.hoten);
    printf("Diem:"); scanf("%d", &a.diem);
```


10.3.2. Phép gán giữa các biến cấu trúc

```
b = a;
```

```
strcpy(c.hoten, a.hoten);
```

```
c.diem = a.diem;
```

```
printf("Bien a: ");
```

```
printf("%-20s%3d\n", a.hoten, a.diem);
```

```
printf("Bien b: ");
```

```
printf("%-20s%3d\n", b.hoten, b.diem);
```

```
printf("Bien c: ");
```

```
printf("%-20s%3d\n", c.hoten, c.diem);
```

```
getch();
```

```
}
```

10.4. Mảng cấu trúc

- Là tập hợp các phần tử có cùng kiểu dữ liệu là kiểu cấu trúc
- Mục đích:
 - Lưu trữ một tập hợp các phần tử có cùng kiểu.
 - Mỗi phần tử là một tập hợp các thành phần có thể khác nhau: thông tin các sinh viên trong lớp, đội bóng...

7.4. Mảng cấu trúc

- **Cú pháp:**

```
struct ten_cau_truc ten_mang_ctruc[so_ptu_toi_da];
```

- **Ví dụ:**

```
struct sinh_vien
{
    char ho_ten[20];
    float diem_thi;
};
struct sinh_vien lop[80];
```

10.4. Mảng cấu trúc

Ví dụ:

- Khai báo một cấu trúc gồm họ tên và điểm thi Tin đại cương
- Khai báo mảng tên là Tin1 gồm 60 phần tử có kiểu trên
- Nhập thông tin cho mảng trên và hiển thị kết quả.

Bài tập

Viết 1 chương trình thực hiện những công việc sau:

- Yêu cầu người dùng nhập vào một số nguyên dương n với $5 \leq n \leq 20$ (có kiểm tra tính hợp lệ của giá trị được nhập vào, nếu giá trị n nhập vào không thỏa mãn điều kiện thì yêu cầu nhập lại)
- Yêu cầu người dùng nhập vào thông tin của n sinh viên gồm những mục sau
 - Họ và tên: có kiểu dữ liệu là chuỗi ký tự gồm không quá 30 ký tự
 - Lớp: chuỗi ký tự có độ dài không quá 5 ký tự
 - Điểm thi Tin đại cương: là một số nguyên có giá trị từ 0 đến 10

Bài tập

Viết 1 chương trình thực hiện những công việc sau (tiếp):

- Đưa ra màn hình danh sách các sinh viên cùng thông tin mà người dùng vừa nhập vào.
- Yêu cầu người dùng nhập vào từ bàn phím một số thực. Đưa ra màn hình danh sách các sinh viên có điểm thi Tin đại cương nhỏ hơn giá trị số thực vừa nhập vào.
- Đưa ra màn hình danh sách sinh viên được sắp xếp theo chiều giảm dần của điểm thi Tin đại cương.