



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

## TIN HỌC ĐẠI CƯƠNG

### Bài 09. Kiểu cấu trúc

#### Nội dung

1. Khái niệm cấu trúc
2. Khai báo và sử dụng cấu trúc
3. Xử lý dữ liệu cấu trúc
4. Mảng cấu trúc

## Nội dung

1. Khái niệm cấu trúc
2. Khai báo và sử dụng cấu trúc
3. Xử lý dữ liệu cấu trúc
4. Mảng cấu trúc

3

## 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 môn Tin đại cương của sinh viên: Họ tên SV, MSSV, Điểm quá trình, Điểm thi.
  - Thông tin về cầu thủ: Họ tên, Tuổi, CLB, Số áo, Vị trí,...

4

## 2. Khai báo và sử dụng cấu trúc

### 2.1. Khai báo kiểu dữ liệu cấu trúc

### 2.2. Khai báo biến cấu trúc

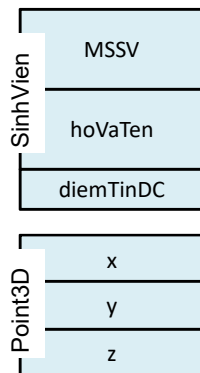
### 2.3. Định nghĩa kiểu dữ liệu với typedef

5

## 2.1. Khai báo kiểu dữ liệu cấu trúc

- Khai báo cấu trúc

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



- Ví dụ

```
struct SinhVien{  
  char MSSV[10];  
  char hoVaTen[30];  
  float diemTinDC;  
}  
struct Point3D{  
  float x;  
  float y;  
  float z;  
}
```

6

## 2.2. Khai báo biến cấu trúc

- Cú pháp:

```
struct Tên_Cấu_Trúc tenBien;
```

- Ví dụ:

– struct SinhVien a, b, c;

- Kết hợp khai báo

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

7

## 2.2. Khai báo biến cấu trúc

- Các cấu trúc có thể được khai báo lồng nhau

```
struct DiemThi {  
    float dToan, dLy, dHoa;  
};  
struct ThiSinh{  
    char SBD[10];  
    char hoTen[30];  
    struct DiemThi ketQua;  
} thiSinh1, thiSinh2;
```

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

8

## 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 ThiSinh{
    char SBD[10];
    char hoTen[30];
    struct [DiemThi]{
        float dToan, dLy, dHoa;
    } ketQua;
} thiSinh1, thiSinh2;
```

9

## 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_Cũ Tên_Mới;
hoặc typedef struct Tên_Cũ {
    <khai báo các trường dữ liệu>;
} Tên_Mới;
```

- Chú ý: cho phép đặt Tên\_Mới trùng Tên\_Cũ

10

## 2.3. Định nghĩa kiểu dữ liệu với typedef

- Ví dụ:

```
struct Point3D{
    float x, y, z;
}
struct Point3D M;
typedef struct Point3D Point3D;
Point3D N;

typedef struct {
    float x, y, z;
}Point3D;
Point3D M;
Point3D N;
```

11

## 2.3. Định nghĩa kiểu dữ liệu với typedef

- Ví dụ

```
typedef struct Point2D {
    float x, y;
}Point2D, Diem2Chieu, TenBatKi;
Point2D X;
Diem2Chieu Y;
TenBatKi Z;
```

=> Point2D, Diem2Chieu, TenBatKi là các tên cấu trúc, không phải tên biến

12

## 3. Xử lý dữ liệu cấu trúc

3.1. Truy cập các trường dữ liệu

3.2. Phép gán giữa các biến cấu trúc

13

## 3.1. Truy cập các trường dữ liệu

- Cú pháp

***tenBien.tenTruong***

- 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

14

### 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

15

### 3.1. Truy cập các trường dữ liệu

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

16



### 3.1. Truy cập các trường dữ liệu

```
printf("Ten diem: ");  
fflush(stdin);  
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();  
return 0;  
}
```

17

### 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:

```
tenBien1 = tenBien2;
```

18

## 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.  
b?c

19

## 3.2. Phép gán giữa các biến cấu trúc

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

## 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();
return 0;
}
```

21

## 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...
- Khai báo:  
***struct Tên\_Cấu\_Trúc tenMang [kích\_thước];***

22

## 4. Mảng cấu trúc

- Ví dụ
  1. Khai báo một cấu trúc gồm tên và điểm thi Tin đại cương
  2. Khai báo mảng tên là sv thuộc kiểu trên
  3. Nhập từ bàn phím số n là số sinh viên trong lớp
  4. Nhập thông tin n sinh viên và lưu vào mảng sv
  5. Sắp mảng mảng theo thứ tự tăng dần của điểm/tên
  6. Hiện thị ra màn hình danh sách các sinh viên có điểm  $\geq 8$

23

## Chương trình quản lý thông tin thí sinh

Viết chương trình quản lý thông tin thí sinh có các chức năng sau:

- Các thông tin cần quản lí:
  - Họ
  - Tên
  - SBD
  - Điểm 3 môn
  - Điểm tổng
- Số lượng tối đa trong danh sách : 100

24

## Chương trình quản lý thông tin thí sinh

- Nhập thông tin thí sinh từ bàn phím
- Sắp xếp danh sách theo thứ tự ABC của tên thí sinh
- Hiện thị danh sách thí sinh theo định dạng sau

STT	Họ và tên	SBD	Môn Toán	Môn Lý	Môn Hóa	Tổng
-----	-----------	-----	----------	--------	---------	------

- Tìm thông tin thí sinh theo tên
- Tìm thông tin thí sinh theo SBD
- Sửa thông tin của một thí sinh

25

## Chương trình quản lý đặt vé rạp phim

- Mỗi khách hàng đặt vé cần cung cấp các thông tin sau:
  - Họ và tên : Xâu ký tự
  - Số điện thoại : Xâu ký tự
  - Số vé trẻ em : số nguyên
  - Số vé người lớn : số nguyên
  - Tên phim : Xâu ký tự
  - Phòng chiếu : số nguyên
- Số lượng phần tử tối đa : 100

26

## Chương trình quản lý đặt vé rạp phim

- Nhập thông tin đặt vé từ khách và tính tổng số tiền khách cần trả. Quy định giá vé:
  - Trẻ em : 20.000 VND
  - Người lớn : 40.000 VND
- Tìm kiếm thông tin đặt vé theo tên khách
- Tính tổng doanh thu của rạp phim
- Tính tổng doanh thu của từng phòng chiếu
- Tính tổng doanh thu của từng phim

27

## Bài tập

Viết chương trình quản lý thông tin các đội bóng tham dự giải đấu World Cup 2018

1. Khai báo mảng cấu trúc lưu trữ thông tin
2. Nhập dữ liệu các đội bóng
3. Hiện thị đầy đủ danh sách các đội bóng
4. Hiện thị các đội bóng trong một bảng thi đấu. Tên bảng thi đấu do người dùng nhập từ bàn phím. (Gợi ý: bài toán tìm kiếm)

28

## Thảo luận



29