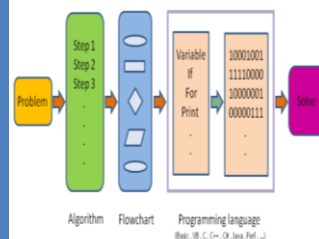




# NHẬP MÔN LẬP TRÌNH

## Chương 8 KIỂU CẤU TRÚC

GV: Lê Thị Bích Hằng  
Trường Đại học Nha Trang



### KIỂU CẤU TRÚC Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến  
Truy xuất các thành phần của cấu trúc  
Mảng cấu trúc  
Một số bài toán cơ bản về cấu trúc  
Tài liệu tham khảo

#### ▪ Đặt vấn đề

- Thông tin **một** SV
  - MSSV : kiểu chuỗi
  - Tên SV : kiểu chuỗi
  - Ngày tháng năm sinh : kiểu chuỗi
  - Phái : kiểu ký tự
  - Điểm Toán, Lý, Hóa : kiểu số thực
- Yêu cầu
  - **Lưu** thông tin **một** SV?
  - **Truyền** thông tin **một** SV vào hàm?

⇒ Khai báo các biến để lưu trữ thông tin 1 SV

```
char mssv[7]; // "0012078"
char hoten[30]; // "Nguyen Van A"
char ntms[8]; // "29/12/82"
char phai; // 'n'
float toan, ly, hoa; // 8.5 9.0 10.0
```

⇒ Tạo các biến để lưu trữ thông tin 1 SV

```
void XuatSV(char mssv[], char hoten[], char ntms[], char phai, float toan, float ly, float hoa);
```



## KIỂU CẤU TRÚC

### Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến  
Truy xuất các thành phần của cấu trúc  
Mảng cấu trúc  
Một số bài toán cơ bản về cấu trúc  
Tài liệu tham khảo

#### ▪ Đặt vấn đề

- Yêu cầu
  - Lưu thông tin **n** SV?
  - Truyền thông tin **n** SV vào hàm?

#### ▪ Nhận xét

- Đặt tên biến khó khăn và khó quản lý
- Truyền tham số cho hàm quá nhiều
- Tìm kiếm, sắp xếp, sao chép,... khó khăn
- Tốn nhiều bộ nhớ
- ...

#### ▪ Ý tưởng

- Gộp những thông tin của cùng 1 SV thành một kiểu dữ liệu mới => Kiểu **struct**

## KIỂU CẤU TRÚC

### Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến  
Truy xuất các thành phần của cấu trúc  
Mảng cấu trúc  
Một số bài toán cơ bản về cấu trúc  
Tài liệu tham khảo

#### ▪ Định nghĩa kiểu cấu trúc

- Cú pháp (Dạng 1):

```
struct Tên_cấu_trúc
{
    Kiểu_dữ_liệu Trường_1;
    Kiểu_dữ_liệu Trường_2;
    .....
    Kiểu_dữ_liệu Trường_n;
};
```

- **Tên\_cấu\_trúc**: tên kiểu cấu trúc.
- **Kiểu\_dữ\_liệu Trường<sub>i</sub>** ( $i = 1..n$ ): mỗi trường trong cấu trúc có dữ liệu thuộc kiểu nào đó, tên của trường phải là một tên được đặt theo quy tắc đặt tên của định danh.

- Ví dụ:

```
struct ngay
{
    unsigned char ngay_thu;
    unsigned char thang;
    unsigned char nam;
};
```

```
struct NhanVien
{
    int maNV;
    char ten[40];
    char dia_chi[30];
    double bac_luong;
    ngay ngay_sinh;
    ngay ngay_vao_co_quan;
};
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

**Định nghĩa và khai báo biến**

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Định nghĩa kiểu cấu trúc

- Cú pháp (Dạng 2):

```
typedef struct
{
    Kiểu_dữ_liệu Trường_1;
    Kiểu_dữ_liệu Trường_2;
    .....
    Kiểu_dữ_liệu Trường_n;
}Tên_cấu_trúc;
```

- Ví dụ:

```
typedef struct
{
    unsigned char ngay_thu;
    unsigned char thang;
    unsigned char nam;
} ngay;
```

Thành phần của cấu trúc là cấu trúc khác

```
typedef struct
{
    int maNV;
    char ten[40];
    char dia_chi[30];
    double bac_luong;
    ngay ngay_sinh;
    ngay ngay_vao_co_quan;
} NhanVien;
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

**Định nghĩa và khai báo biến**

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Khai báo biến kiểu cấu trúc

- Đối với cấu trúc được định nghĩa theo dạng 1:

```
struct Tên_cấu_trúc Biến_1 [, Biến_2...];
```

- Ví dụ:

```
struct NhanVien nv1, nv2;
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Khai báo biến kiểu cấu trúc

- Đối với cấu trúc được định nghĩa theo dạng 2:

```
Tên_cấu_trúc Biến_1 [, Biến_2...];
```

- Ví dụ:

```
NhanVien nv1, nv2;
```

- Các biến cấu trúc sau khi được khai báo sẽ được cấp phát bộ nhớ cho tất cả các thành phần của biến.



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Khởi tạo cấu trúc

- Việc khởi tạo cấu trúc có thể được thực hiện trong lúc khai báo biến cấu trúc.

- Các trường của cấu trúc được khởi tạo được đặt giữa 2 dấu { và }, chúng được phân cách nhau bởi dấu phẩy (,).

- Ví dụ:

```
struct ngay ngay_sinh = {15, 11, 1985};
```

⇒ Xác định một cấu trúc (kiểu *ngay*) có tên là *ngay\_sinh* và khởi đầu cho các thành phần của nó.

⇒ Nội dung của *ngay\_sinh.ngay\_thu* là 15, của *ngay\_sinh.thang* là 11, của *ngay\_sinh.nam* là 1985.



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Truy xuất thành phần kiểu cấu trúc

- Không thể truy xuất trực tiếp dữ liệu kiểu cấu trúc.
- Việc xử lý một cấu trúc phải được thực hiện thông qua các thành phần cơ bản của nó bằng cách sử dụng toán tử `.` (*dot operator*).
- Cú pháp:

```
Biến_cấu_trúc.Tên_trường
```

- Ví dụ:

```
printf("%5d %s", nv1.maNV, nv1.ho_ten);
```

```
printf("%d", nv1.ngay_sinh.nam);
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Gán dữ liệu kiểu cấu trúc

- Cú pháp (Dạng 1):

```
Biến_cấu_trúc_đích = Biến_cấu_trúc_nguồn;
```

- Ví dụ:

```
typedef struct
{
    int x;
    int y;
}DIEM;

DIEM diem1 = {29, 15};
DIEM diem2;

diem2 = diem1;
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Gán dữ liệu kiểu cấu trúc

- Cú pháp (Dạng 2):

```
Biến_cấu_trúc_đích.Tên_thành_phần = Giá_trị;
```

- Ví dụ:

```
typedef struct
{
    int x;
    int y;
}DIEM;

DIEM diem1 = {29, 15};
DIEM diem2;

diem2.x = diem1.x;
diem2.y = diem1.y * 2;
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Lưu ý

- Có thể sử dụng phép toán lấy địa chỉ đối với thành phần cấu trúc để nhập số liệu trực tiếp vào thành phần cấu trúc.

- Ví dụ: `scanf("%d", &nv1.ngay_sinh.nam);`

- Đối với các thành phần không phải số nguyên, câu lệnh dạng trên có thể cho kết quả không chính xác. Vì vậy trước tiên ta nên nhập số liệu vào một biến trung gian, sau đó mới gán cho thành phần cấu trúc.

- Ví dụ:

```
typedef struct
{
    float x;
    float y;
}DIEM;

DIEM diem1;
float temp;
scanf("%f", &temp);
d1.x = temp;
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

### Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

## ▪ Khai báo mảng cấu trúc

- Cú pháp:

```
struct Tên_kiểu_cấu_trúc Tên_mảng[Số_phần_tử];
```

- Ví dụ:

```
struct NhanVien nv1, nv2;
struct NhanVien dsnv1[10], dsnv2[7];
```



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

### Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

## ▪ Gán dữ liệu cho mảng cấu trúc

- Có thể thực hiện phép gán trên các biến và phần tử mảng cấu trúc cùng kiểu theo các cách sau:
  - Gán hai biến cấu trúc cho nhau
  - Gán biến cho phần tử mảng cấu trúc
  - Gán phần tử mảng cho biến
  - Gán hai phần tử mảng cho nhau



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

### ▪ Một số bài toán cơ bản

- Nhập và hiển thị thông tin một nhân viên.
- Nhập và hiển thị thông tin n nhân viên.
- Sắp xếp danh sách nhân viên giảm dần theo tên.
- ...



## KIỂU CẤU TRÚC

Giới thiệu kiểu cấu trúc

Định nghĩa và khai báo biến

Truy xuất các thành phần của cấu trúc

Mảng cấu trúc

Một số bài toán cơ bản về cấu trúc

Tài liệu tham khảo

- [1] Quách Tuấn Ngọc, *Ngôn ngữ lập trình C*, NXB Giáo dục, 2003
- [2] Trần Đan Thư, *Giáo trình lập trình C tập 1&2*, NXB ĐHQG TP. Hồ Chí Minh
- [3] Trần Đan Thư, *Nhập môn lập trình*, NXB KH&KT, 2019
- [4] Phạm Văn Ất, *Kỹ thuật lập trình C căn bản và nâng cao*, NXB Hồng Đức 2009
- [5] B. W. Kernighan and D. M. Ritchie, *C Programming Language, 2<sup>nd</sup> edition*, Prentice Hall, 1998
- [6] Lê Thị Bích Hằng, *Tài liệu tham khảo Nhập môn lập trình*, Lưu hành nội bộ
- [7] Đặng Bình Phương, *Bài giảng Nhập môn lập trình*