

# Kiểu struct

---

- Khái niệm, mục đích sử dụng
- Cú pháp tổng quát
- Khởi tạo
- Sử dụng biến struct
- Ví dụ minh họa

# Khái niệm, mục đích sử dụng

- Struct là tập hợp của các biến cùng liên quan đến một đối tượng dưới một cái tên chung
- Các biến thành phần cấu thành nên struct có thể thuộc nhiều kiểu khác nhau
- Kiểu dữ liệu cấu trúc là một kiểu do người dùng tự định nghĩa, nó còn có tên khác là kiểu dữ liệu dẫn xuất hay kiểu dữ liệu tổng hợp
- Mục đích: biểu diễn các thông tin đặc trưng của một đối tượng cụ thể
- Ví dụ cấu trúc student chứa thông tin về tên, mã sv, ngành, học phí, điểm trung bình, khoa... của một sinh viên

# Cú pháp tổng quát

Cú pháp tổng quát tạo struct:

```
struct name {  
    data_type property_name;  
    data_type property_name;  
};
```

- Luôn bắt đầu với keyword *struct*
- Name: tên của cấu trúc muốn mô tả, đặt như quy tắc đặt tên biến
- Phần thân struct bao bởi cặp {} chứa các biến mô tả thông tin liên quan. Quy tắc đặt tên thành phần như đặt tên biến
- Luôn kết thúc struct bằng dấu chấm phẩy ;

# Ví dụ về struct

- Ví dụ một struct mô tả thông tin sinh viên:

```
struct student {  
    char id[20]; // ma sv  
    char name[30]; // ho ten  
    char address[40]; // dia chi  
    int age; // tuoi  
    float avgGrades; // diem trung binh  
}; // luon ket thuc struct voi dau ;
```

# Tạo biến struct

Để tạo biến struct ta có hai cách:

- Cách 1: khai báo khi cần sử dụng

```
struct struct_name struct_variable;
```

```
struct student oanh, nam; // khai bao nhieu bien tren 1 dong  
struct student thanh; // khai bao 1 bien tren 1 dong
```

# Tạo biến struct

- Cách 2: khai báo biến toàn cục khi định nghĩa struct:

```
struct name {  
    data_type property_name;  
    data_type property_name;  
} variable1, variable2...;  
  
struct student {  
    char id[20]; // ma sv  
    char name[30]; // ho ten  
    char address[40]; // dia chi  
    int age; // tuoi  
    float avgGrades; // diem trung binh  
} hoa, loan; // khai bao hai bien toan cuc cua struct
```



# Khởi tạo struct

- Ta có thể khởi tạo các thành phần tương ứng của một struct:

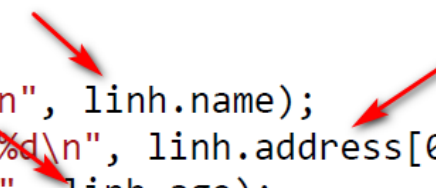
```
struct student linh = {  
    "B24DCCN123",           // khởi tạo cho id  
    "Nguyen Khanh Linh",    // khởi tạo cho ten  
    "Ha Noi",               // khởi tạo cho dia chi  
    20,                     // khởi tạo cho tuoi  
    3.25f                   // khởi tạo diem  
};
```

- Nếu có ít giá trị được khởi tạo hơn các biến thành phần thì phần còn lại sẽ nhận giá trị mặc định
- Biến struct toàn cục sẽ nhận giá trị NULL nếu ta không chủ động khởi tạo tường minh
- Có thể khởi tạo struct bằng cách gán cho nó một struct cùng kiểu

# Sử dụng biến struct

- Để truy cập đến các thành phần cấu thành của struct, ta dùng toán tử dấu chấm .
- Cú pháp tổng quát:  
*biến\_struct.tên\_thành\_phần*
- Ví dụ:

```
struct student linh = {  
    "B24DCCN123",           // khởi tạo cho id  
    "Nguyen Khanh Linh",    // khởi tạo cho ten  
};  
  
printf("Name: %s\n", linh.name);  
printf("Address: %d\n", linh.address[0]);  
printf("Age: %d\n", linh.age);
```





# Ví dụ nhập dữ liệu của struct

- Ví dụ sau sẽ nhập dữ liệu vào cho biến struct:

```
struct student linh;  
  
printf("%s", "Student ID: ");  
scanf("%19s", linh.id);  
  
printf("%s", "Full name: ");  
scanf("%*c%29[^\n]", linh.name);  
  
printf("%s", "Address: ");  
scanf("%*c%39[^\n]", linh.address);  
  
printf("%s", "Age: ");  
scanf("%d", &linh.age);  
  
printf("%s", "Average grades: ");  
scanf("%f", &linh.avgGrades);
```



# Tiếp theo

Truyền biến struct vào hàm