

Cơ sở lập trình

Bài 8. Kiểu dữ liệu có cấu trúc (9 tiết)

Biên soạn TS. Trần Minh Thái
Giảng viên: Lê Thị Minh Nguyễn
Email: nguyentm@huflit.edu.vn

Mục tiêu

1. Khái niệm
2. Khai báo
3. Truy xuất thành phần
4. Mảng kiểu dữ liệu có cấu trúc và xử lý

Khái niệm

- Do người dùng định nghĩa bằng cách gom nhóm các kiểu dữ liệu (KDL) cơ bản có sẵn trong C thành một KDL phức hợp nhiều thành phần

- Khai báo

```
struct tenCauTruc
{
    khai báo các thuộc tính;
};
typedef struct tenCauTruc tenKDL;
```

3

VD khai báo KDL lưu thông tin của ngày

```
struct ttDate
{
    char thu[4];
    int ngay;
    int thang;
    int nam;
};
typedef struct ttDate Date;
```

4

VD khai báo KDL lưu trữ thông tin của một quyển sách

```
struct ttSach
{
    char tenSach[31];
    char nhaXB[21];
    int namXB;
    int soTrang;
    int giaBia;
};
typedef struct ttSach Sach;
```

5

Truy xuất thành phần cấu trúc

- Biến kiểu cấu trúc
tenKDL tenBien;
 tenBien.tenThuocTinh;
Date x;
 x.ngay = 5;
- Biến con trỏ kiểu cấu trúc
tenKDL *tenBien;
 tenBien->tenThuocTinh;
Date *x;
 x->ngay = 5;

6

VD: Viết chương trình nhập vào tọa độ hai điểm trong mặt phẳng và tính tổng hai tọa độ này

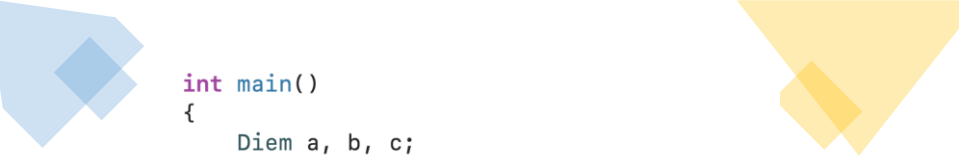
```
struct ttDiem
{
    int x;
    int y;
};
typedef struct ttDiem Diem;

void Nhap(Diem *d);
void Xuat(Diem d);
Diem Tong(Diem d1, Diem d2);
```

7

```
void Nhap(Diem *d)
{
    printf("Hoanh do x = ");
    scanf("%d", &(d->x));
    printf("Tung do y = ");
    scanf("%d", &(d->y));
}
void Xuat(Diem d)
{
    printf("(%d, %d)", d.x, d.y);
}
Diem Tong(Diem d1, Diem d2)
{
    Diem kq;
    kq.x=d1.x+d2.x;
    kq.y=d1.y+d2.y;
    return kq;
}
```





```

int main()
{
    Diem a, b, c;


    printf("Nhap toa do diem thu nhat:\n");
    Nhap(&a);
    printf("Nhap toa do diem thu hai:\n");
    Nhap(&b);

    printf("\nToa do diem thu nhat: ");
    Xuat(a);
    printf("\nToa do diem thu hai: ");
    Xuat(b);

    c=Tong(a, b);
    printf("\nToa do tong hai diem: ");
    Xuat(c);

    return 0;
}

```



Bài tập

7.1. Bổ sung thêm tính khoảng cách của 2 điểm a và b trong chương trình VD trên.

7.2. Viết chương trình nhập vào thông tin của một sinh viên (**SinhVien**) gồm:

- Mã số sinh viên (**maSo**) gồm có 10 ký tự
- Họ và tên (**hoTen**) có tối đa 30 ký tự
- Điểm giữa kỳ (**GK**)
- Điểm thực hành (**TH**)
- Điểm lý thuyết (**LT**)

Tính điểm tổng kết môn = $GK \cdot 10\% + TH \cdot 30\% + LT \cdot 60\%$

Bài tập

7.3. Cho thông tin của linh kiện máy tính (**LinhKien**) gồm:

- Mã số linh kiện (**maSo**): Chuỗi có 5 ký tự
- Tên linh kiện (**tenLK**): Chuỗi có tối đa 20 ký tự
- Xuất xứ (**xuatXu**): Chuỗi có tối đa 15 ký tự
- Đơn giá (**donGia**)
- Số lượng (**soLuong**)

Hãy khai báo cấu trúc dữ liệu và viết chương trình nhập xuất thông tin linh kiện trên.

11

Bài tập về nhà

BTVN 7.1. Viết chương trình nhập vào 2 thời gian t_1 và t_2 (thông tin thời gian gồm: giờ, phút và giây).

Tính khoảng cách giữa t_1 và t_2 (đơn vị tính bằng giây)

BTVN 7.2. Viết chương trình nhập vào thông tin của một sinh viên gồm:

- Mã số sinh viên: có 10 ký tự
- Họ và tên: có tối đa 30 ký tự
- Ngày tháng năm sinh
- Điểm trung bình

12

Sửa bài tập

Viết chương trình nhập vào thông tin của một sinh viên gồm:

- Mã số sinh viên: có 10 ký tự
- Họ và tên: có tối đa 30 ký tự
- Ngày tháng năm sinh
- Điểm trung bình

Các em xem và chuẩn bị trước

13

Mảng cấu trúc

Mảng dữ liệu có cấu trúc

- Cách khai báo tương tự như mảng một chiều (KDL là KDL có cấu trúc)
- Cách truy cập phần tử trong mảng cũng như truy cập trên mảng một chiều.
- Nhưng do từng phần tử có kiểu cấu trúc nên phải chỉ định rõ cần lấy thành phần nào, tức là phải truy cập đến thành phần cuối cùng có kiểu là dữ liệu cơ bản

15

Nguyên tắc xử lý mảng dữ liệu có cấu trúc

Do kiểu dữ liệu có cấu trúc thường chứa rất nhiều thành phần nên khi viết chương trình loại này ta cần lưu ý:

- *Xây dựng hàm xử lý cho một kiểu cấu trúc.*
- *Muốn xử lý cho mảng cấu trúc, ta gọi lại hàm xử lý cho một kiểu cấu trúc đã được xây dựng bằng cách dùng vòng lặp.*

16

VD: hàm
nhập và
xuất mảng
các phân số

```
#include<stdio.h>
#define MAX 100
struct ttPhanSo
{
    int tuSo;
    int mauSo;
};
typedef struct ttPhanSo PhanSo;

void NhapPS(PhanSo &ps);
void XuatPS(PhanSo ps);
void NhapMangPS(PhanSo a[], int n);
void XuatMangPS(PhanSo a[], int n);
```

17

Nhập vào mảng phân số

```
void NhapPS(PhanSo &ps)
{
    printf("Tu so = ");
    scanf("%d", &ps.tuSo);
    printf("Mau so = ");
    scanf("%d", &ps.mauSo);
}
void NhapMangPS(PhanSo a[], int n)
{
    for(int i=0; i<n; i++)
    {
        printf("Nhap phan so tai vi tri %d:\n", i);
        NhapPS(a[i]);
    }
}
```

18

Xuất mảng phân số

```
void XuatPS(PhanSo ps)
{
    printf("%d/%d", ps.tuSo, ps.mauSo);
}
void XuatMangPS(PhanSo a[], int n)
{
    for(int i=0; i<n; i++)
    {
        XuatPS(a[i]);
        printf(" ");
    }
}
```

19

Hàm main()

```
int main()
{
    int n;
    PhanSo a[MAX];

    printf("So luong phan so n = ");
    scanf("%d", &n);
    NhapMangPS(a, n);

    printf("Danh sach phan so vua nhap:\n");
    XuatMangPS(a, n);

    return 0;
}
```

20

Bài tập

Viết chương trình nhập vào mảng các phân số, cho biết:

1. Phân số có giá trị lớn nhất
2. Tìm phân số x trong mảng
3. Tính tổng các phân số trong mảng

21

Bài tập tổng hợp – Quản lý mặt hàng

Cho thông tin mặt hàng gồm:

- **Mã mặt hàng:** chuỗi gồm 5 ký tự
- **Tên mặt hàng:** chuỗi gồm tối đa 20 ký tự
- **Loại mặt hàng** (“Bánh kẹo”, “Nước giải khát”, “Cà phê”, ...)
- **Xuất xứ:** chuỗi gồm tối đa 10 ký tự
- **Số lượng**
- **Đơn giá**
- **Ngày sản xuất**

Viết chương trình gồm các chức năng sau:

1. Nhập và xuất danh sách các mặt hàng
2. In ra những mặt hàng có xuất xứ x
3. Sắp xếp danh sách theo thứ tự tăng dần của đơn giá
4. Sắp xếp danh sách theo thứ tự tăng dần của mã mặt hàng
5. Tìm mặt hàng có mã số x
6. Tính tổng thành tiền của các mặt hàng có loại là “Bánh kẹo”

Q&A

