

## CHƯƠNG 2. KIỂU DỮ LIỆU CÓ CẤU TRÚC

- **Khái niệm**
- **Khai báo**
- **Cách truy xuất thành phần bên trong cấu trúc**
- **Xử lý cấu trúc đơn**
- **Xử lý mảng có cấu trúc**

# Đặt vấn đề

- Thông tin của một **sinh viên** gồm:
  - Mã sinh viên (kiểu xâu ký tự)
  - Họ tên (kiểu xâu ký tự)
  - Năm sinh (kiểu int)
  - Giới tính (0: Nam, 1: Nữ) (kiểu int hoặc bool)
  - Điểm Văn (kiểu float)
  - Điểm Toán (kiểu float)
- Yêu cầu:
  - **Lưu** thông tin **n sinh viên**
  - **Truyền** thông tin **n sinh viên** vào **hàm**

# Khái niệm

- Một cấu trúc là một tập hợp các biến được tham chiếu thông qua một tên chung. Các biến tạo nên cấu trúc được gọi là các thuộc tính (hoặc các thành viên) của cấu trúc.
- Sự khác biệt giữa kiểu cấu trúc và kiểu mảng là: các phần tử của mảng là cùng kiểu còn các phần tử của kiểu cấu trúc có thể có kiểu khác nhau.
- Khai báo:

## Cách 1:

```
struct tên_struct{  
    khai báo các thuộc tính;  
};  
typedef struct tên_struct TênKDL;
```

## Cách 2:

```
typedef struct{  
    khai báo các thuộc tính;  
} TênKDL;
```

# Khái niệm

- **Ví dụ 1:** khai báo kiểu dữ liệu lưu thông tin của ngày:

## Cách 1:

```
struct ttDate{  
    char thu[10];  
    int ngay;  
    int thang;  
    int nam;  
};  
typedef struct ttDate Date;
```

## Cách 2:

```
typedef struct{  
    char thu[10];  
    int ngay;  
    int thang;  
    int nam;  
} Date;
```

# Truy cập các thuộc tính cấu trúc

## ➤ Biến kiểu cấu trúc

**TênKDL** tên\_biến;  
tên\_biến.tên\_thuộc\_tính;

Ví dụ: **Date** x;

x.ngay = 5;

## ➤ Biến con trỏ kiểu cấu trúc

**TênKDL** \*tên\_biến\_con\_trỏ;

Cách 1: tên\_biến\_con\_trỏ -> tên\_thuộc\_tính;

Cách 2: tên\_biến\_con\_trỏ.tên\_thuộc\_tính;

Ví dụ: **Date** \*x ;

x = (Date\*) malloc(sizeof(Date));

x -> ngay = 5 ;

x.ngay = 8;

**Ví dụ:** Viết chương trình nhập vào tọa độ hai điểm trong mặt phẳng và tìm tọa độ trung điểm của đoạn thẳng nối 2 điểm này.

```
#include <stdio.h>
#include <conio.h>

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

void nhap(Diem &d){
    printf("Hoanh do : ");
    scanf("%d", &d.x);
    printf("Tung do : ");
    scanf("%d", &d.y);
}
```

```
Diem trungdiem(Diem d1, Diem d2){
    Diem td;
    td.x = (d1.x + d2.x)/2;
    td.y = (d1.y + d2.y)/2;
    return td;
}

int main (){
    Diem a , b, td; //khai bao 3 diem a, b, td;
    printf("\nNhap vao toa do diem A: \n");
    nhap(a);
    printf("\nNhap vao toa do diem B: \n");
    nhap(b);
    td = trungdiem(a, b);
    printf("\n Trung diem cua doan thang AB:
        (%d, %d)", td.x, td.y);
    return 0;
}
```

# Khai báo đệ quy

## Cách 1:

```
struct tên_struct1{  
    khai báo các thuộc tính;  
};  
  
struct tên_struct2{  
    khai báo các thuộc tính;  
    struct tên_struct1 *tên_thuộc_tính_đệ_qui;  
};
```

## Ví dụ:

```
struct ttSinhVien{  
    char masv[10];  
    char hoten[30];  
    int ns;  
    bool gt;  
    float dv;  
    float dt;  
};  
  
struct ttLop{  
    char malop[10];  
    char tenlop[30];  
    int siso;  
    ttSinhVien sv[50];  
};  
  
typedef struct ttLop Lop;
```

# Khai báo đệ quy

## Cách 2:

```
typedef struct {  
    khai báo các thuộc tính;  
}TênKDL1;
```

```
typedef struct{  
    khai báo các thuộc tính;  
    TênKDL1 tên_thuộc_tính_đệ_qui;  
}TênKDL2;
```

## Ví dụ:

```
typedef struct{  
    char masv[10];  
    char hoten[30];  
    int ns;  
    bool gt;  
    float dv;  
    float dt;  
}sinhvien;  
typedef struct{  
    char malop[10];  
    char tenlop[30];  
    int siso;  
    sinhvien sv[50];  
}lop;
```



## Mảng cấu trúc

- **Cách khai báo** tương tự như mảng một chiều (Kiểu dữ liệu bây giờ là kiểu dữ liệu 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.
- **Nguyên tắc lập trình trên mảng 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.

# Ví dụ

10

**Ví dụ 2:** Viết chương trình nhập vào thông tin của một sinh viên gồm:

- Mã sinh viên (kiểu chuỗi ký tự dài 10)
  - Năm sinh (kiểu int)
  - Điểm Văn (kiểu float)
  - Họ tên (kiểu chuỗi ký tự dài 30)
  - Giới tính (0: Nam, 1: Nữ)
  - Điểm Toán (kiểu float)
1. Nhập danh sách n sinh viên cho một lớp học (với n được nhập từ bàn phím)
  2. Đưa danh sách sinh viên vừa nhập ra màn hình
  3. Nhập danh sách sinh viên từ tệp VD2.in
  4. Lưu danh sách sinh viên vừa nhập vào file có tên là VD2.OUT
  5. Tìm và in ra màn hình các sinh viên có điểm Văn >5 hoặc Toán >5
  6. Tìm và đưa ra màn hình các sinh viên có tổng điểm  $\geq 14$  và điểm Toán >5
  7. Đưa ra màn hình những sinh viên nam có tuổi lớn nhất
  8. Tìm và đưa ra các sinh viên có điểm trung bình cao nhất vào tệp VD2.OUT

# Ví dụ: Viết chương trình thực hiện ý 1 + 2

11

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
```

```
typedef struct{
    char masv[10];
    char hoten[30];
    bool gt;
    int ns;
    float dv;
    float dt;
}sinhvien;
```

```
//y1: Nhap danh sach sinh vien tu ban phim
void nhap(sinhvien *a, int &n){
    printf("\nNhap so luong sinh vien: ");scanf("%d", &n);
    for(int i=0; i<n;i++){
        printf("\nSinh vien thu %d\n", i+1);fflush(stdin);
        printf("Ma SV: ");gets(a[i].masv);
        printf("Ho ten: ");gets(a[i].hoten);
        printf("Nam sinh: ");scanf("%d", &a[i].ns);
        printf("Gioi tinh(0: nam, 1: nu): ");scanf("%d", &a[i].gt);
        printf("Diem van: ");scanf("%f", &a[i].dv);
        printf("Diem toan: ");scanf("%f", &a[i].dt);
    }
}
```

## Ví dụ: Viết chương trình thực hiện ý 1 + 2

//y2: Xuất danh sách sinh viên ra màn hình

```
void xuất(sinhvien *a, int n){  
    for(int i=0; i<n; i++){  
        printf("\n%-10s%-20s%6d", a[i].masv, a[i].hoten, a[i].ns);  
        if(a[i].gt==0)    printf(" Nam ");  
        else printf(" Nu  ");  
        printf("%8.2f%8.2f", a[i].dv, a[i].dt);  
    }  
}  
  
int main(){//Ham chinh  
    int n;  
    sinhvien *a;  
    a=(sinhvien*)malloc(20*sizeof(sinhvien));  
    nhap(a,n);  
    printf("\nDanh sach sinh vien:");  xuất(a,n);  
    free(a);  
    return 0;  
}
```

## Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

```
//Y3: Nhap danh sach sinh vien tu tep VD2.IN cach 1
void doc(sinhvien *a, int &n, FILE*f){
    f=fopen("VD2.IN", "rt");
    if(f==NULL){
        printf("Tep khong ton tai!"); exit(0);
    }
    fscanf(f, "%d\n", &n);
    for(int i=0; i<n;i++){
        fgets(a[i].masv, 10, f);
        fgets(a[i].hoten, 30, f);
        fscanf(f, "%d", &a[i].ns);
        fscanf(f, "%d", &a[i].gt);
        fscanf(f, "%f", &a[i].dv);
        fscanf(f, "%f\n", &a[i].dt);
    }
    fclose(f);
}
```

## Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

//Y3: Nhập danh sách sinh viên từ tệp VD2.IN cách 2

```
void doc2(sinhvien *a, int &n, FILE*f){
    f=fopen("VD2.IN", "rt");
    if(f==NULL){
        printf("Tệp không tồn tại!");
        exit(0);
    }
    fscanf(f, "%d\n", &n);
    int i,j;
    for(i=0; i<n;i++){
        fflush(stdin);
        fgets(a[i].masv,10,f);
        j=strlen(a[i].masv);
        a[i].masv[j-1]='\0';
```

```
        fgets(a[i].hoten,30,f);
        j=strlen(a[i].hoten);
        a[i].hoten[j-1]='\0';
```

```
        fscanf(f, "%d", &a[i].ns);
        fscanf(f, "%d", &a[i].gt);
        fscanf(f, "%f", &a[i].dv);
        fscanf(f, "%f\n", &a[i].dt);
```

```
    }
    fclose(f);
```

```
}
```

# Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

15

```
//Y4: Xuất ra tệp VD2.OUT
void ghi(sinhvien *a, int n, FILE *f){
    f=fopen("VD2.OUT","wt"); //Xóa dữ liệu đã có để ghi dữ liệu mới
    //f = fopen("VD2.OUT", "at"); //ghi dữ liệu vào cuối tệp
    for(int i=0; i<n; i++){
        fprintf(f,"\n%-10s%-20s%6d", a[i].masv, a[i].hoten, a[i].ns);
        if(a[i].gt==0)    fprintf(f," Nam ");
        else fprintf(f," Nu ");
        fprintf(f,"%8.2f%8.2f", a[i].dv, a[i].dt);
    }
    fclose(f);
}
```

## Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

```
//Y5: Xuất ra màn hình sinh viên có điểm văn >5 hoặc điểm toán >5
void xuấttheodiem1(sinhvien *a, int n){
    for(int i=0; i<n; i++){
        if(a[i].dv>5 || a[i].dt>5){
            printf("\n%-10s%-20s%6d", a[i].masv, a[i].hoten, a[i].ns);
            if(a[i].gt==0)    printf(" Nam ");
            else printf(" Nu ");
            printf("%8.2f%8.2f", a[i].dv, a[i].dt);
        }
    }
}

//Y6: Xuất ra màn hình sinh viên có tổng điểm >14 và điểm toán >5
void xuấttheodiem2(sinhvien *a, int n){
    for(int i=0; i<n; i++){
        if((a[i].dv+a[i].dt)>14 && a[i].dt>5){
            printf("\n%-10s%-20s%6d", a[i].masv, a[i].hoten, a[i].ns);
            if(a[i].gt==0)    printf(" Nam ");
            else printf(" Nu ");
            printf("%8.2f%8.2f", a[i].dv, a[i].dt);
        }
    }
}
```



## Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

17

//Y7: Xuất ra màn hình sinh viên nam có tuổi lớn nhất

```
void xuattheotuai(sinhvien *a, int n){
    int i, min;
    for(i=0; i<n; i++)
        if(a[i].gt==0){
            min=a[i].ns;
            break;
        }
    for(i=0; i<n; i++)
        if(a[i].gt==0 && a[i].ns<min) min=a[i].ns;
    for(i=0; i<n; i++)
        if(a[i].gt==0 && a[i].ns==min){
            printf("\n%-10s%-20s%6d", a[i].masv, a[i].hoten, a[i].ns);
            if(a[i].gt==0)    printf(" Nam ");
            else printf(" Nu  ");
            printf("%8.2f%8.2f", a[i].dv, a[i].dt);
        }
}
```

## Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

//Y8: Xuất ra màn hình sinh viên có điểm trung bình cao nhất (có nghĩa là có tổng điểm cao nhất)

```
void xuattheodiem3(sinhvien *a, int n){
    int i, max=a[0].dt+a[0].dv;
    for(i=1; i<n; i++)
        if((a[i].dt+a[i].dv)>max)
            max=a[i].dt+a[i].dv;
    for(i=0; i<n; i++)
        if((a[i].dt+a[i].dv)==max){
            printf("\n%-10s%-20s%6d", a[i].masv, a[i].hoten, a[i].ns);
            if(a[i].gt==0)    printf(" Nam ");
            else printf(" Nu  ");
            printf("%8.2f%8.2f", a[i].dv, a[i].dt);
        }
}
```

## Ví dụ: Bổ sung các ý tiếp theo và sửa hàm main

19

```
int main(){
    int n;
    sinhvien *a;
    a=(sinhvien*)malloc(20*sizeof(sinhvien));
    FILE *f;
    nhap(a,n); //doc(a,n,f);
    printf("\nDanh sach sinh vien:");  xuat(a,n);
    ghi(a,n,f); //ghi2(a,n,f);
    printf("\n\nDSSV co diem van >5 hoac diem toan >5:"); xuattheodiem1(a,n);
    printf("\n\nDSSV co tong diem >14 va diem toan >5:"); xuattheodiem2(a,n);
    printf("\n\nDSSV nam lon tuoi nhat:");  xuattheotuai(a,n);
    printf("\n\nDSSV co diem trung binh lon nhat:"); xuattheodiem3(a,n);
    free(a);
    return 0;
}
```

# Bài tập

**Bài 1:** Xây dựng các **cấu trúc Phim** với các thông tin cần quản lý gồm:

- Mã phim
- Tên đạo diễn
- Năm sản xuất
- Thể loại (1: Hình sự, 2: Tâm lý, 3: Hài)
- Tên phim
- Thời lượng (tính theo phút)
- Số lượng

1. Nhập danh sách n phim (với n được nhập từ bàn phím)
2. Lưu danh sách phim vừa nhập vào file có tên là BAI1.OUT
3. Tính tổng số lượng phim
4. Tìm và đưa ra màn hình danh sách các phim hài có năm sản xuất mới nhất
5. Nhập năm sản xuất, đưa ra màn hình danh sách các phim có năm sản xuất vừa nhập
6. Nhập một thể loại, đưa ra màn hình danh sách các phim thuộc thể loại này
7. Nhập tên một đạo diễn, in ra màn hình các bộ phim của đạo diễn này

# Bài tập

**Bài 2:** Xây dựng **cấu trúc Nhân viên** với các thông tin cần quản lý gồm:

- Họ tên
- Giới tính (0: Nam, 1: Nữ)
- Lương cơ bản
- Phạt
- Mã nhân viên
- Hệ số lương
- Thưởng

1. Nhập danh sách n nhân viên (với n được nhập từ bàn phím)
2. Lưu danh sách nhân viên vừa nhập vào file có tên là BAI2.OUT
3. Tìm và đưa ra danh sách nhân viên có hệ số lương cao nhất
4. Tìm và đưa ra danh sách các nhân viên nữ có thực lĩnh cao nhất, trong đó:  
 $\text{Thực lĩnh} = \text{Lương cơ bản} + \text{Thưởng} - \text{Phạt}$
5. Nhập vào một số, đưa ra những nhân viên có lương cơ bản lớn hơn hoặc bằng số vừa nhập
6. Nhập vào tên một nhân viên, đưa ra màn hình các thông tin về nhân viên này (nếu có)

**Bài 3:** Xây dựng **cấu trúc Mặt hàng** với các thông tin cần quản lý sau:

- Mã hàng
- Tên mặt hàng
- Số lượng
- Đơn giá
- Số lượng tồn
- Thời gian bảo hành (tính theo đơn vị tháng)

1. Nhập danh sách n mặt hàng với n được nhập từ bàn phím
2. Lưu danh sách vừa nhập vào file có tên là BAI3.OUT
3. Tìm và in ra màn hình mặt hàng có giá tiền cao nhất, thấp nhất
4. Tìm và in ra màn hình mặt hàng có số lượng tồn ít nhất, cao nhất

# Bài tập

23

**Bài 4:** Xây dựng **cấu trúc Vé tàu** với các thông tin cần quản lý sau:

- Mã vé
- Ngày khởi hành
- Ga đi
- Ga đến
- Loại tàu
- Loại chỗ ngồi (1: ngồi cứng, 2: ngồi mềm, 3: nằm cứng, 4: nằm mềm)
- Giá vé
- Phần trăm giảm giá

1. Nhập danh sách n vé tàu (với n được nhập từ bàn phím)
2. Lưu danh sách vé tàu vừa nhập vào file có tên là BAI4.OUT
3. In danh sách các vé tàu có ga đến là Huế, ga đi từ Thanh Hoá vào file có tên là BAI4.OUT
4. Nhập vào ngày đi và in ra màn hình danh sách các vé tàu có ga đến là Hà Nội đi vào ngày vừa nhập
5. Đếm số khách đi tàu loại chỗ ngồi là *nằm cứng*
6. Đếm số khách đi tàu loại chỗ ngồi là *ngồi mềm*
7. Nhập vào phần trăm giảm giá, thống kê và in ra danh sách vé có phần trăm giảm giá lớn hơn phần trăm vừa nhập

## Bài tập

**Bài 5:** Xây dựng **cấu trúc Điện** với các thông tin cần quản lý sau:

- Mã hoá đơn
- Kỳ thu
- Tên khách hàng
- Điện năng tiêu thụ (Kw)

1. Nhập danh sách n khách hàng với n được nhập từ bàn phím
2. Lưu danh sách vừa nhập vào file có tên là OUT.DAT
3. Tính tiền điện của các khách hàng và xuất ra màn hình hóa đơn của mỗi khách hàng, biết cách tính tiền điện như sau:

100 kw đầu tiên là 550 đồng/kw

50 kw tiếp theo là 900 đồng/kw

50 kw tiếp theo là 1200 đồng/kw

Số kw còn lại là 1500 đồng/kw

Thuế 10% trên tổng số tiền phải trả