

结 构 体

本课为自主学习内容

问题

- 在程序里存储1000学生的详细资料（姓名、年龄、性别、成绩、……），怎么办？

某学校学生成绩表

学号	姓名	性别	入学时间	计算机	英语	数学	音乐
1	令狐冲	男	1999	90.5	83	72	82
2	林平之	男	1999	78	92	88.5	78
3	岳灵珊	女	1999	89.5	72	98	66
4	任莹莹	女	1999	78	95	87	90.5
5						
6						

用多个数组来存储

```
int    Id[1000];           /* 存学号*/
string Name[1000];         /* 字符串类型，存名字*/
char   Gen[1000];          /* 存性别*/
int     Time[1000];         /* 存入学时间*/
float   Computer[1000];     /* 存计算机课的成绩*/
float   English[1000];      /* 存英语课的成绩*/
float   Math[1000];         /* 存数学课的成绩*/
float   Music[1000];        /* 存音乐课的成绩*/
```

数组的解决方法

缺点：数组多，零散，使用麻烦，容易出错。

结构体的解决方法

■ struct STUDENT

```
{  
    int    ID;           /* 学生的学号 */  
    string Name;         /* 学生的姓名 */  
    char   Gen;          /* 学生的性别 */  
    int    Time;         /* 学生的入学时间 */  
    float  Computer;     /* 学生的计算机成绩 */  
    float  English;      /* 学生的英语成绩 */  
    float  Math;         /* 学生的数学成绩 */  
    float  Music;        /* 学生的音乐成绩 */  
};
```

再申明一个结构体类型的数据即可：

```
STUDENT stu[1001];
```

■ 结构体(structure):

- 把关系紧密且逻辑相关的多种不同类型的变量组织到统一的名字之下，也称复合数据类型

1. 结构体类型定义(申明)格式



在定义结构体时，在大括号中的内容也称为“成员表列”或“域表”。其中，每个成员名的命名规则与变量名命名相同；数据类型可以是基本变量类型和数组类型，也可以是指针变量类型，或者是另一个结构体类型，用分号“;”作为结束符。整个结构的定义也用分号作为结束符。

- 例如，一个学生的姓名(name)、学号(ID)、性别(Gen)、等信息，构成了这个学生档案的基本资料，通过使用**结构体类型**就可以将它们作为一个整体来处理。

```
struct STUDENT
{
    int      ID;          /* 学生的学号 */
    char      Name[10];   /* 学生的姓名, 结构体成员可以是数组 */
    string    Gen;        /* 学生的性别 */
    int       Time;       /* 学生的入学时间 */
    float     Computer;   /* 学生的计算机成绩 */
    float     English;    /* 学生的英语成绩 */
    float     Math;       /* 学生的数学成绩 */
    float     Music;      /* 学生的音乐成绩 */
};
```

上面只是对一种结构体类型的数据进行了申明，若要使用它，需要申明结构体类型的变量：

STUDENT s1, s2; //STUDENT就相当于一种你自己定义的数据类型
这里就申明了两个STUDENT类型的变量，s1和s2。

结构体变量的使用

访问结构体变量的成员，需要使用“.”运算符

格式：结构体变量名.成员名 （“.”是成员运算符，优先级很高）

例如：

```
STUDENT s1,s2;  
cin >> s1.Name >> s1.ID >> s1.Math;  
scanf("%d%f",&s1.Time,&s1.English);  
s2.Music = 99.5;  
s2.Math = 88;  
s2.Time = 2015;  
cout << s1.Name;  
printf("%d",s2.Time);
```


结构体的一些注意事项：

(1) 不能将一个结构体类型变量作为一个整体加以使用，而只能对结构体类型变量中的各个成员分别引用。

例如，对前面声明的结构体类型变量s1,s2下列引用都是错误的：

```
cout<<s1;  cout<<s2;  
cin>>s1;  cin>>s2;
```

只能用“.”逐个使用其成员,例如: `cout<<s1.Name; cin>>s2.Math;`

(2) **同一结构体类型的结构体变量之间允许相互赋值**，而不同结构体类型的结构体变量之间不允许相互赋值，即使两者包含有同样的成员。

例如: **s1=s2;**

上面语句就相当于：

```
s1.ID = s2.ID;  
s1.Name = s2.Name;  
s1.Gen = s2.Gen;  
s1.Time = s2.Time;  
s1.Math = s2.Math;  
s1.English = s2.English;  
s1.Compuer = s2.Computer;  
s1.Music = s2.Music;
```

结构体的一些注意事项：

(3) 如果成员本身又属另一个结构体类型，则要用若干个成员运算符，一级一级地找到最低的一级成员。例如：

```
struct AA
{
    float x;
    int y[10];
    char z;
};
```

```
struct BB
{
    int c;
    AA d;
};
```

```
BB t;
cin>>t.d.x;
for(i=0;i<=9;i++) cin>>t.d.y[i];
cout<<t.d.z;
cout<<t.d.y[5];
```

结构体数组 每个数组元素都是一个结构体类型的数据

定义:

```
struct XueSheng
{
    int num;
    string name;
    float score;
};
XueSheng stu[50]; //定义了一个名为stu的数组，其中每个元素都是XueSheng类型
```

使用:

- 可访问结构体数组元素的成员: `stu[1].name` `stu[5].num`
- 可进行同一个结构体类型变量(数组元素)的整体赋值
`stu[7]=stu[3];`
- 要对结构体数组元素的单个成员进行输入输出
`for(i=1;i<=20;i++) cin>> stu[i].num >> stu[i].name >> stu[i].score;`

例：输入 $n(n \leq 100)$ 个学生的学号，姓名和成绩(每个学生的成绩都不同)，
输出成绩最好的学生的信息。

```
struct student
{
    int num;
    string name;
    double score;
};

student st[101] , Best;           // 声明了一个结构体数组st和一个结构体变量Best

void main()
{
    int i,n;
    Best.score = -1;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>> st[i].num >> st[i].name >> st[i].score;
        if(st[i].score>Best.score) Best=st[i];
    }
    cout<<Best.num<<'    '<<Best.name<<'    '<<Best.score<<endl;
}
```

例：输入 $n(n \leq 10000)$ 个学生的学号，姓名和成绩，按成绩由高到低输出每个学生的信息。

```
struct XueSheng
{
    int num;
    string name;
    double score;
};

XueSheng xs[10001];
```

```
bool cmp(XueSheng a, XueSheng b)
{
    return a.score > b.score;
}
```

```
void main()
{
    int i, n;
    cin >> n;
    for (i = 1; i <= n; i++)
        cin >> xs[i].num >> xs[i].name >> xs[i].score;
    sort(xs + 1, xs + 1 + n, cmp); // 用sort函数排序，排序规则由比较函数cmp决定
    for (i = 1; i <= n; i++)
        cout << xs[i].num << ' ' << xs[i].name << ' ' << xs[i].score << endl;
}
```

例2：输入n($n \leq 10000$)个学生的学号，姓名，数学和语文成绩，按数学成绩由高到低输出每个学生的信息。如果数学成绩相同，按语文成绩由高到低排序。

```
struct XueSheng
{
    int num;
    string name;
    double math;
    double chinese;
};
```

```
XueSheng  xs[10001];
```

```
void main()
```

```
{
    int i,n;
    cin>>n;
    for(i=1;i<=n;i++)
        cin>>xs[i].num>>xs[i].name>>xs[i].score;
    sort(xs+1,xs+1+n,cmp);
    for(i=1;i<=n;i++)
        cout<<xs[i].num<<'    '<<xs[i].name<<'    '<<xs[i].score<<endl;
}
```

```
bool cmp(XueSheng a,XueSheng b)
{
    if(a.math==b.math) return a.chinese>b.chinese;
    else return a.math>b.math;
}
```