

# 第九章 构造体与链表

北京邮电大学出版社

# 内容提要

- 构造体类型的定义
- 构造体类型变量
- 构造体类型数组
- 构造体类型指针
- 构造体与函数
- 链表



# §11.1 概述

## ■ 问题定义：

有时需要将不同类型的数据组合成一种有机

的整体，以便于引用。如：

一种学生有学号/姓名/性别/年龄/地址等属性

```
int num; char name[20]; char sex; int age; char addr[30];
```

Num	name	sex	age	addr
-----	------	-----	-----	------

100101	Li Fun	M	18	Beijing
--------	--------	---	----	---------

图11-1

# 构造体类型的定义

- 构造体

- 构造体是一种构造数据类型
- 用途：把不同类型的数据组合成一种整体---自定义数据类型
- 构造体类型定义

```
struct [构造体名]
{
    类型标识符  组员名1;
    类型标识符  组员名2;
    .....
};
```

**struct**是关键字  
,  
不能省略

正当标识符  
可省:无名构造体

组员类型能够是  
基本型或构造型

# 构造体类型的定义

- 定义构造体类型，描述下列数据

- (1) 学生情况：包括学生的学号、姓名、性别、年龄、C语言课程成绩：

```
struct student
```

```
{int no;           /*学号*/
```

```
  char name [10] ; /*姓名*/
```

```
  char sex;       /*性别*/
```

```
  int age;        /*年龄*/
```

```
  float score;   /*C成绩*/
```

```
}; 注：‘;’不能省
```

构造体类型定义描述构造的组织形式, 不分配内存

如考虑10门课程成绩，加上总成绩与平均成绩，可作如下定义：

```
struct student  
{int no;          /*学号  
*/  
    char name [10] ; /*姓名*/  
    char sex;      /*性别*/  
    int age;       /*年龄*/  
    float score [10] ; /*10门课程成绩  
*/  
    float tcj, acj; /*总成绩， 平均成绩  
*/  
};
```

- 定义构造体类型，描述下列数据
  - (2) 个人数据：包括姓名、性别、年龄、身高、体重、住址：

```
struct person  
{char name [20] ; /*姓名*/  
char sex; /*性别*/  
int age; /*年龄  
*/  
float height; /*身高*/  
float weight; /*体重*/  
char addr [50] ; /*住址*/  
};
```

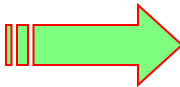
(3) 日期构造体类型涉及年、月、日:

```
struct date
{int year; /*年*/
month; /*月*/
day; /*日*/
};
```

(4) 如职员信息构造体类型:

```
struct person
{
char name[20]; /*姓名*/
char address[40]; /*地址*/
float salary; /*工资*/
float cost; /*扣款*/
struct date hiredate; /*聘任日期*/
};
```

构造体类型能够嵌套定义即一种构造体类型中的某些组员又是其他构造体类型





# 构造体类型变量的定义(1)

- 先定义构造体类型，再定义构造体变量

✦ 一般形式：

```
struct 构造体名
{
    类型标识符 组员名;
```

```
例 struct student
{
    int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
};
struct student stu1,stu2;
```

```
例
#define STUDENT struct student

STUDENT
{
    int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
};
STUDENT stu1,stu2;
```

# 构造体类型变量的定义(2)

- 定义构造体类型的同步定义构造体变量

✦ 一般形式:

```
例 struct student
{   int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
}stu1,stu2;
```

```
struct 构造体名
{
    类型标识符  组员名;
    类型标识符  组员名;
    .....
}变量名表列;
```

# 构造体类型变量的定义(3)

- 直接定义构造体变量

✦ 一般形式:

```
例 struct
{   int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
}stu1,stu2;
```

```
struct
{
    类型标识符  组员名;
    类型标识符  组员名;
    .....
}变量名表列;
```

用**无名构造体**直接定义变量，**只能一次**，不能在再需要时使用所定义的构造体类型。

# 构造体类型声明的阐明

## 阐明

✦ 构造体类型与构造体变量概念不同

• 类型:不分配内存;

变量:分配内存

• 类型:不能赋值、存取、运算;

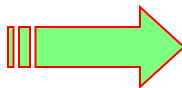
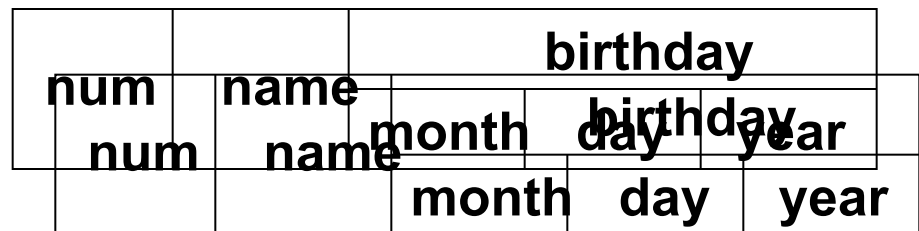
变量:能够

✦ 对于构造体中的组员，能够单独使用，它的作用与地位相当于一般变量。

✦ 构造体可嵌套，其组员也能够是一种构造体变量。

```
例 struct student
{
    int num;
    char name[20];
    struct date
    {
        int month;
        int day;
        int year;
    } birthday;
};
stu;
}st
```

变量名可相同，两者不代表同一对象，不会混



# 构造体变量的使用(1)

- 由构造体变量名引用其组员

✦ 构造体变量不能整体引用,只能引用变量组员

引用方式: 构造体变量名.组员名

✦ 能够将一种构造体变量整体引用,另一种构造体变量

例 struct student  
{ int num;

例 struct student  
{ int num;  
char name[20];  
char sex;  
int age;  
float score;  
char addr[30];  
}stu1,stu2;

例 struct stu  
{ int num;  
char nam  
struct da  
{ int mo  
int day  
int year;  
}birthday;  
}stu1,stu2;

例 st  
{  
{

stu1.num=10;

组员(分量)运算符

优先级: 1

结合性: 从左向右

stu1.score=85.5;

s,"%c %d %f %s\n", stu1); (x)

stu1.score+=stu2.score;

stu1.age++;  
in", 'M', 19, 87.5, "DaLian"); (x)

birthday

stu2=stu1;

(✓)

month

day

year

# 构造体变量的初始化（1）

- 形式一：

```
struct 构造体名
{
    类型标识符  组员名;
    类型标识符  组员名;
    .....
};
struct 构造体名 构造体变量={初始数据};
```

```
例 struct student
{
    int num;
    char name[20];
    char sex;
    int age;
    char addr[30];
};
struct student stu1={112,“Wang Lin”,‘M’,19, “200 Beijing Road”};
```

# 构造体变量的初始化（2）

- 形式二:

```
struct 构造体名
{
    类型标识符  组员名;
    类型标识符  组员名;
    .....
}构造体变量={初始数据};
```

```
例 struct student
{
    int num;
    char name[20];
    char sex;
    int age;
    char addr[30];
}stu1={112,“Wang Lin”,‘M’,19, “200 Beijing Road”};
```

# 构造体变量的初始化 (3)

- 形式三:

```
struct  
{  
    类型标识符  组员名;  
    类型标识符  组员名;  
    .....  
}构造体变量={初始数据};
```

```
例 struct  
{    int num;  
    char name[20];  
    char sex;  
    int age;  
    char addr[30];  
}stu1={112, "Wang Lin", 'M', 19, "200 Beijing Road"};
```



**例 9.1 求某同学上学期8门课程的总成绩与平均成绩。**

程序如下：

```
main ( )  
{ int i;  
  struct st  
  {char xm[8];  
    float cj[9];  
    float tcj, acj;  
  } stu;
```

```
scanf("%s", stu.xm) ; /*输入*/  
for(i=1; i<=8;i++)  
    scanf("%f", &stu.cj[i]);  
stu.tcj=0.0;          /*求总成绩*/  
for(i=1; i<=8;i++)  
    stu.tcj+=stu.cj[i];  
stu.acj=stu.tcj/8;   /*求平均成绩*/  
printf("%s的总成绩=%6.2f,平均成绩  
=%6.2f",stu.xm,stu.tcj,stu.acj) ;  
}
```

输入数据: CHEN 80 86 79 98 88 72 96 66

运营成果: CHEN的总成绩=577.00, 平均成绩= 72.13

# 构造体变量的输入与输出

- C语言不能把构造体变量作为一种整体进行输入或输出，应该按组员变量输入输出。
- 例如，若有一种构造体变量：

**struct**

**{char name[15];**

**char addr[20];**

**long num;**

**}stud={"wang dawei", "125 Beijing  
  Road", 3021118};**

因为变量stud包括两个字符串数据和一种长整型数据，所以输出stud变量，应该使用如下方式：

```
printf(“%s,%s,%ld\n”,stud.name,stud.addr,stud.num);
```

输入stud变量的各组员值，则用：

```
scanf(“%s%s%ld”,stud.name,stud.addr,&stud.num);
```

# 构造体数组

一种构造体变量中能够存储一组数据（如一种学生的学号、姓名、成绩等数据）。假如有 10 个学生的数据需要参加运算，显然应该用数组，这就是结构体数组。构造体数组与此前简介过的数值型数组不同之处在于每个数组元素都是一种构造体类型的数据，它们都分别涉及各个组员（分量）项。

# 构造体类型数组

- 构造体数组的定义

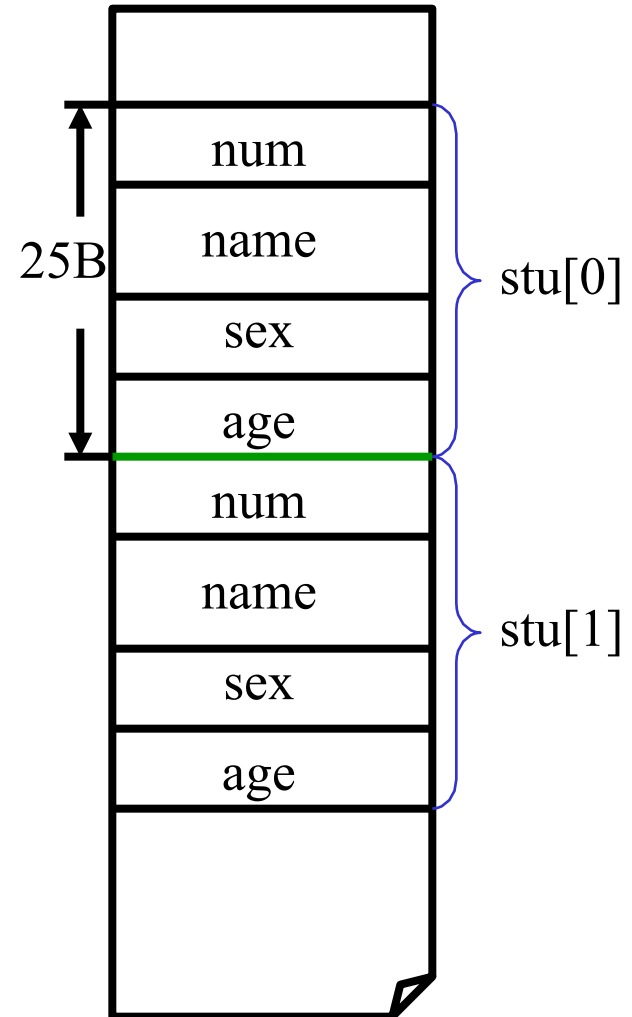
✦ 三种形式:

形式一:

形式二:

形式三:

```
struct
{
    int num;
    char name[20];
    char sex;
    int age;
}stu[2];
```



# 构造体数组初始化与引用

- 构造体数组初始化

## 构造体数组引用

引用方式: 构造体数组名[下标].组员名

```
struct student  
{  
    int num;  
    char name[20];  
};
```

```
struct student  
{  
    int num;  
    char name[20];  
    char sex;  
    int age;  
};  
str[3];  
stu[1].age++;  
str[0] = { "Jiang Lin", 'M', 20}, 0,  
          { "Jiang", 'M', 19},  
          { "Yan", 'F', 19}};  
strcpy(stu[0].name, "ZhaoDa");
```

全部初始化时维数可省

引用构造体数组元素有如下规则：

(1)引用某一元素的一种组员。students[i].num

(2)能够将一种构造体数组元素赋给同一构造体类型数组中的另一种元素，或赋给同一类型的变量。例如：

```
struct student students[3],student1;
```

```
Student1=students[0];
```

```
Students[2]=students[1];
```

```
Students[1]=student1;
```

(3)不能把构造体数组元素作为一种整体直接进行输入或输出，只能以单个组员对象进行输入输出。

```
scanf(“%s”,students[0].name);
```

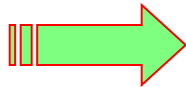
```
printf(“%ld”,students[0].num);
```



## 例 统计候选人选票

```
struct person
{ char name[20];
  int count;
}leader[3]={"Li",0,"Zhang",0,"Wang",0};
main()
{ int i,j; char leader_name[20];
  for(i=1;i<=10;i++)
  { scanf("%s",leader_name);
    for(j=0;j<3;j++)
      if(strcmp(leader_name,leader[j].name)==0)
        leader[j].count++;
  }
  for(i=0;i<3;i++)
    printf("%5s:%d\n",leader[i].name,leader[i].count);
}
```

name	count
Li	0
Zhang	0
Wang	0



# 指向构造体类型数据的指针

一种构造体变量的指针就是该变量所占据的内存段的起始地址。能够设一种指针变量，用来指向一种构造体变量，此时该指针变量的值是构造体变量的起始地址。指针变量也能够用来指向构造体数组中的元素。

# 指向构造体变量的指针

- 指向构造体变量的指针

```
main()
{ struct student
  { long int num;
    char name[20];
    char sex;
```

地址

(\*构造例  
例

```
int float score;
struct student stu_1,*p=&n;
p=&stu_1;
p->num=101; ⇔ n=101;
stu_1.num=89.5;
```

```
struct student stu1,stu2;
struct student *p=&stu1;
stu1.num=101; ⇔ (*p).num=101
```

```
strcpy(stu_1.name,"Li Lin");
p->sex='M';
p->score=89.5;
printf("\nNo:%ld\nname:%s\nsex:%c\nscore:%f\n",
      (*p).num,p->name,stu_1.sex,p->score);
}
```



## 总结：构造体组员变量引用方式

①构造体变量.组员名

②(\*p).组员名

③p->组员名

其中，->称为指向运算符

请分析下列几种运算：

①p->n

②p->n++

③++p->n

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/108130136120006116>