

## 本章知识点小结

内容	实例	备注
定义结构体类型	<pre>struct student {     long studentID;     char studentName[10];     char studentSex;     struct date birthday;     int score[4]; };</pre>	用户自定义的构造数据类型。结构体是将不同类型的数据成员组织在一起形成的数据结构，适合于对关系紧密、逻辑相关、具有相同或者不同属性的数据进行处理。
定义共用体类型	<pre>union maritalState {     int single;     struct marriedState married;     struct divorceState divorce; };</pre>	用户自定义的构造数据类型。共用体是将逻辑相关、情形互斥的不同类型的数据组织在一起形成的数据结构，每一时刻只有一个数据成员起作用。
定义枚举类型	<pre>enum response{no, yes, none};</pre>	
用 typedef 定义数据类型	<pre>typedef struct student STUDENT;</pre>	为已有的类型定义一个新的名字
指向结构体变量的指针	<pre>STUDENT stu1; STUDENT *pt = &amp;stu1;</pre>	
指向结构体数组的指针	<pre>STUDENT stu[30]; STUDENT *pt = stu;</pre>	
成员选择运算符	<pre>stu1.birthday.year = 1991;</pre>	用于访问结构体变量的成员
指向运算符	<pre>pt-&gt;birthday.year = 1991;</pre>	用于访问结构体指针指向的结构体的成员
结构体变量作函数参数	<pre>void Func(struct date d);</pre>	向函数传递结构体变量属于按值调用
结构体指针作函数参数	<pre>void Func(struct date *pt);</pre>	向函数传递结构体指针属于按地址调用
结构体数组作函数参数	<pre>void Input(STUDENT stu[],int n,int m);</pre>	向函数传递结构体数组属于按地址调用

## 本章常见错误小结

常见错误实例	常见错误描述	错误类型
<pre>struct date {     int year;     int month;     int day; }</pre>	定义结构体或者共用体类型时，忘记在最后的}后面加分号	编译错误
	将一种类型的结构体变量对另一种类型的结构体变量进行赋值	编译错误
	对两个结构体或者共用体进行比较操作	编译错误
	在结构体指向运算符的两个组成符号-和>之间插入空格，或写成→	编译错误
	直接使用结构体的成员变量名访问结构体变量的成员	编译错误
	使用指向运算符访问结构体变量的成员	编译错误
	使用成员选择运算符访问结构体指针指向的结构体的成员	编译错误
	误以为结构体实际所占内存的字节数是结构体每个成员所占内存字节数的“总和”	理解错误
	误以为结构只能包含一种数据类型	理解错误
	误以为不同结构体的成员名字不能相同	理解错误
	误以为用 typedef 可以定义一种新的数据类型	理解错误