

挑战30天C 入门极限：C\_C 中的结构体 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/134/2021\\_2022\\_\\_E6\\_8C\\_91\\_E6\\_88\\_9830\\_E5\\_A4\\_c97\\_134523.htm](https://www.100test.com/kao_ti2020/134/2021_2022__E6_8C_91_E6_88_9830_E5_A4_c97_134523.htm)

什么是结构体？简单的来说，结构体就是一个可以包含不同数据类型的一个结构，它是一种可以自己定义的数据类型，它的特点和数组主要有两点不同，首先结构体可以在一个结构中声明不同的数据类型，第二相同结构的结构体变量是可以相互赋值的，而数组是做不到的，因为数组是单一数据类型的数据集合，它本身不是数据类型(而结构体是)，数组名称是常量指针，所以不可以做为左值进行运算，所以数组之间就不能通过数组名称相互复制了，即使数据类型和数组大小完全相同。定义结构体使用struct修饰符，例如：`struct test { float a. int b. }`。上面的代码就定义了一个名为test的结构体，它的数据类型就是test，它包含两个成员a和b，成员a的数据类型为浮点型，成员b的数据类型为整型。由于结构体本身就是自定义的数据类型，定义结构体变量的方法和定义普通变量的方法一样。`test pn1`。这样就定义了一test结构体数据类型的结构体变量pn1，结构体成员的访问通过点操作符进行，`pn1.a=10`就对结构体变量pn1的成员a进行了赋值操作。注意:结构体生命的时候本身不占用任何内存空间，只有当你用你定义的结构体类型定义结构体变量的时候计算机才会分配内存。结构体，同样是可以定义指针的，那么结构体指针就叫做结构指针。结构指针通过->符号来访问成员，下面我们就以上所说的看一个完整的例子

```
#include iostream> #include string> using namespace std. struct test//定义一个名为test的结构体 { int a.//定义结构体成员a int
```

```
b.//定义结构体成员b }. void main() { test pn1.//定义结构体变量pn1 test pn2.//定义结构体变量pn2 pn2.a=10.//通过成员操作符.给结构体变量pn2中的成员a赋值 pn2.b=3.//通过成员操作符.给结构体变量pn2中的成员b赋值 pn1=pn2.//把pn2中所有的成员值复制给具有相同结构的结构体变量pn1 cout cout test *point.//定义结构指针 point=&pn2.//指针指向结构体变量pn2的内存地址 cout point->a=99.//通过结构指针修改结构体变量pn2成员a的值 cout coutab cin.get(). } 总之，结构体可以描述数组不能够清晰描述的结构，它具有数组所不具备的一些功能特性。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com
```