## 1.结构体的定义与使用

结构体是一种构造数据类型,把不同类型的数据组合成一个整体

结构体的定义形式：

struct 结构体名

{

结构体所包含的变量或数组

};

结构体是一种集合，它里面包含了多个变量或数组，它们的类型可以相同，也可以不同，每个这样的变量或数组都称为结构体的成员（Member）。请看下面的一个例子：

struct Student  
{  
 char name[20]; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在学习小组* float score; *//成绩*};

注意：大括号后面的分号;不能少，这是一条完整的语句。

Student为结构体名，它包含了 5 个成员，分别是 name、num、age、group、score。结构体成员的定义方式与变量和数组的定义方式相同，只是不能初始化。

结构体也是一种数据类型，它由程序员自己定义，可以包含多个其他类型的数据。

像 int、float、char 等是由C语言本身提供的数据类型，不能再进行分拆，我们称之为基本数据类型；而结构体可以包含多个基本类型的数据，也可以包含其他的结构体，我们将它称为复杂数据类型或构造数据类型。

先定义结构体类型，再定义结构体变量

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

定义结构体类型的同时定义结构体变量

struct Data  
{  
 int day ;  
 int month;  
 int year  
}time1,time2;

直接定义结构体变量

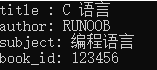
struct  
{  
 char name[20];  
 char sex;  
 int num;  
} person1,person2;

2.结构体变量的初始化

和其它类型变量一样，对结构体变量可以在定义时指定初始值。

#include <stdio.h>  
struct Books  
{  
 char title[50];  
 char author[50];  
 char subject[100];  
 int book\_id;  
} book = {"C 语言", "RUNOOB", "编程语言", 123456};  
  
int main()  
{  
 printf("title : %s\nauthor: %s\nsubject: %s\nbook\_id: %d\n", book.title, book.author, book.subject, book.book\_id);  
}

输出结果：



3.访问结构成员

为了访问结构的成员，我们使用成员访问运算符（.）。

引用形式：<结构体类型变量名>.<成员名>

注意：结构体变量不能整体引用，只能引用变量成员

( 一般我们是将结构体放在主函数外面 ,这里做个反例子 )

成员访问运算符是结构变量名称和我们要访问的结构成员之间的一个句号。可以使用 struct 关键字来定义结构类型的变量。下面的实例演示了结构的用法：

#include <stdio.h>  
  
struct  
{  
 char \*name; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在小组* float score; *//成绩*} stu1;  
int main()  
{  
 *//给结构体成员赋值* stu1.name = "Tom";  
 stu1.num = 12;  
 stu1.age = 18;  
 stu1.group = 'A';  
 stu1.score = 136.5;  
 *//读取结构体成员的值* printf("%s的学号是%d，年龄是%d，在%c组，今年的成绩是%.1f！\n",stu1.name, stu1.num, stu1.age, stu1.group, stu1.score);  
 return 0;  
}

运行结果：



除了可以对成员进行逐一赋值，也可以在定义时整体赋值，例如：

struct  
{  
 char \*name; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在小组* float score; *//成绩*} stu1, stu2 = { "Tom", 12, 18, 'A', 136.5 };

不过整体赋值仅限于定义结构体变量的时候，在使用过程中只能对成员逐一赋值，这和数组的赋值非常类似。

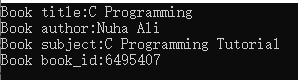
需要注意的是，结构体是一种自定义的数据类型，是创建变量的模板，不占用内存空间；结构体变量才包含了实实在在的数据，需要内存空间来存储。

4.结构作为函数参数

可以把结构作为函数参数，传参方式与其他类型的变量或指针类似。例如：

#include <stdio.h>  
#include<string.h>  
  
struct Books  
{  
 char title[50];  
 char author[50];  
 char subject[100];  
 int book\_id;  
};  
  
void printBook(struct Books *book*)  
{  
 printf("Book title:%s\n", *book*.title);  
 printf("Book author:%s\n", *book*.author);  
 printf("Book subject:%s\n", *book*.subject);  
 printf("Book book\_id:%d\n", *book*.book\_id);  
}  
int main()  
{  
 struct Books *Book1*;*//声明Book1,类型为Books* struct Books *Book2*;  
  
 */\* Book1 详述 \*/* strcpy(*Book1*.title, "C Programming");  
 strcpy(*Book1*.author, "Nuha Ali");  
 strcpy(*Book1*.subject, "C Programming Tutorial");  
 *Book1*.book\_id = 6495407;  
 printBook(*Book1*);  
 return 0;  
}

运行结果：



5.结构体数组

所谓结构体数组，是指数组中的每个元素都是一个结构体。在实际应用中，结构体数组常被用来表示一个拥有相同数据结构的群体，比如一个班的学生、一个车间的职工等。

定义结构体数组和定义结构体变量的方式类似，请看下面的例子：

struct Student  
{  
 char \*name; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在小组* float score; *//成绩*}class[5];

//表示一个班有5个人

结构体数组在定义的同时也可以初始化，例如：

struct Student  
{  
 char \*name; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在小组* float score; *//成绩*}class[5] =  
 {  
 {"Li ping", 5, 18, 'C', 145.0},  
 {"Zhang ping", 4, 19, 'A', 130.5},  
 {"He fang", 1, 18, 'A', 148.5},  
 {"Cheng ling", 2, 17, 'F', 139.0},  
 {"Wang ming", 3, 17, 'B', 144.5}  
 };

当对数组中全部元素赋值时，也可以不给出数组长度，如：

struct Student  
{  
 char \*name; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在小组* float score; *//成绩*}class[] = {  
 {"Li ping", 5, 18, 'C', 145.0},  
 {"Zhang ping", 4, 19, 'A', 130.5},  
 {"He fang", 1, 18, 'A', 148.5},  
 {"Cheng ling", 2, 17, 'F', 139.0},  
 {"Wang ming", 3, 17, 'B', 144.5}  
};

结构体数组的使用也很简单。例如，计算全班学生的总成绩、平均成绩和140分一下的人数：

#include <stdio.h>  
  
struct Student  
{  
 char name[20]; *//姓名* int num; *//学号* int age; *//年龄* char group; *//所在小组* float score; *//成绩*} classes[5] = {  
 {"Li ping", 5, 18, 'C', 145.0},  
 {"Zhang ping", 4, 19, 'A', 130.5},  
 {"He fang", 1, 18, 'A', 148.5},  
 {"Cheng ling", 2, 17, 'F', 139.0},  
 {"Wang ming", 3, 17, 'B', 144.5}  
};  
  
int main()  
{  
 int *i*, *num\_140* = 0;  
 float *sum* = 0;  
 for(*i*=0; *i*<5; *i*++)  
 {  
 *sum* += classes[*i*].score;  
 if(classes[*i*].score < 140) *num\_140*++;  
 }  
 printf(" sum=%.2f\n average=%.2f\n num\_140=%d\n", *sum*, *sum*/5, *num\_140*);  
 return 0;  
}

运行结果：

