

项目 5 “学生选课系统”课程信息数据分析

学习目标

- 1. 掌握结构体类型的含义与定义。
- 2. 掌握结构体变量的初始化与赋值规则。
- 3. 掌握结构体的引用规则。
- 4. 掌握结构体类型数据在实际问题中的应用。

项目内容分析——课程信息数据分析

在 C 语言中,系统提供的基本数据类型有时候不能满足用户的实际需求,允许用户根据需要定义数据类型,如一个学生的信息中,可以包含学号、姓名、年龄、成绩等信息,这时涉及字符串、整型、浮点型数据等多个不同类型的数据,无法通过数组(每个数组元素的数据类型是相同的)结构来实现,这时用户就可以自己定义一个数据类型,使得这种类型能够包含多个复杂的数据。这种用户自己定义的由不同数据类型组成的组合型数据结构称为结构体类型。

任务说明

通过项目 4 的课程信息存储展示发现,对于课程信息来说,应该包含课程编号和课程名称,如果仅通过两个二维的字符数组存储和展示信息过于复杂,且无法满足此类信息的存储和输出需求。本系统中的学生信息数据、教师信息数据等均为这种类型。本次任务对选课信息进行展示,学习结构体类型在实际中的应用,教师与学生的相关信息用法相同,以此类推。

课程信息展示效果如图 5-1 所示。

课程编号	课程名称	任课教师工号	任课教师姓名
1111	数据结构	02	李老師
2222	C语言程序设计	03	王老師
3333	高等数学	01	張老師
4444	大学英语	02	李老師
5555	Java程序设计	02	李老師
请按任意键继续. . .			

图 5-1 课程信息展示效果

知识点巩固

在前面学习了 C 语言的常用基本数据类型,这些类型都是 C 系统中已定义好的数据类

型,但有时处理的数据可能不仅仅包含基本的数据类型,例如在本系统中包含学生的基本数据,仅仅利用系统提供的数据类型不能满足,为此,C语言根据需要建立了自定义的数据类型——结构体类型。要完成该界面设计,需要掌握结构体相关知识。

任务 5.1 结构体定义及应用

结构体类型是C编程中用户自定义的可用的数据类型,它可以存储不同类型的数据项。它是由一系列具有相同类型或不同类型的数据项构成的数据集合,这些数据项称为结构体的成员。

1. 结构体的定义

关键字 struct 和结构体名组合成一种类型标识符,其地位如同通常的 int、char 等类型标识符,其用途就像 int 类型标识符标识整型变量一样,可以用来定义结构体变量。定义变量以后,该变量就可以像定义的其他变量一样使用了;成员又称为成员变量,它是结构体所包含的若干基本的结构类型,必须用“{}”括起来,并且要以分号结束,每个成员应表明具体的数据类型。

以学生为例,创建一个学生结构体类型。

【课堂案例 5-1】 定义一个学生结构体,包含学号、姓名、性别和年龄 4 个成员。下面介绍结构体变量的定义方法。

方法一:

```
//创建结构体模板 struct student
struct student
{
    int num;           //学生学号
    char name[10];     //学生姓名,因为学生姓名是一个字符串,所以这里定义了一个字符数组进行处理
    int age;           //学生年龄
    char sex[4];       //性别
};                    //注意这里的分号不能省略
```

此处,student 是结构体名,该名字是任意定义的,但是尽量起个有意义的名称。其相当于一个模板,可以使用这个模板来定义变量 stu1、stu2、stu3。定义学生变量的方式为:

```
struct student stu1,stu2,stu3;
```

方法二:

```
//定义 3 个结构体变量 stu1、stu2、stu3
struct
{
    int num;           //学生学号
    char name[10];     //学生姓名
    int age;           //学生年龄
    char sex[4];       //性别
}stu1, stu2, stu3;
```

定义结构体类型的同时定义三个变量,这种方式省略了结构体名称,但不能用该结构定义新的变量,所以在实际中应用不多。

方法三:在定义结构体名字的同时定义变量。

```
struct Student
{
    int num;                //学生学号
    char name[10];          //学生姓名
    int age;                //学生年龄
    char sex[4];            //性别
}stu1, stu2, stu3;
```

定义好结构体名字后,同时定义了 3 个学生变量 stu1、stu2、stu3,如果还需要,可以继续定义学生变量,定义方式同方法一的定义方式。

```
struct Student stu4;                //定义 stu4 为 Student 结构体类型
```

在实际应用中,为了简化结构体变量的定义,常常使用以下方式进行定义。

1) 使用 typedef

将结构体模板 struct Student 重新命名为 student。

```
typedef struct Student
{
    char name[20];          //学生姓名
    int num;                //学生学号
    int age;                //学生年龄
}student;
```

或者

```
typedef struct
{
    char name[20];          //学生姓名
    int num;                //学生学号
    int age;                //学生年龄
}student;
```

这样在定义结构体变量时,可以采用以下方法进行简化。

使用 student 创建 3 个结构体变量 stu1、stu2、stu3

```
student stu1, stu2, stu3;
```

此处使用 typedef 为结构体模板 struct Student 定义一个别名 student,使用 typedef 给结构体创建一个别名,这在实际编程中经常使用。

2) 使用 define 宏定义

使一个符号常量来表示一个结构类型。

```
#define Student struct student
Student{
char name[20];           //学生姓名
int num;                 //学生学号
int age;                 //学生年龄
};
Student stu1,stu2;       //定义两个学生变量
```

2. 结构体成员引用

在 C 语言中,使用“分隔符.成员”来获取结构体中的一个成员,一般格式为:

结构变量名.成员名

例如:

```
stu1.name;           //第一个学生的姓名
stu2.num;            //第二个学生的学号
```

获取成员后,就可以对该成员赋值了,以上面定义的 Student 结构体类型为例:

```
stu2.age=20;
```

当对 stu2.name 进行赋值时,注意不可以使用 stu2.name="ZhangSan",因为 name 为一个数组名,即字符数组的首元素地址值,不可以将字符串直接赋值给字符数组名,可以通过 strcpy(stu2.name,"ZhangSan")给 stu2.name 赋值,或通过终端使用 gets()或 scanf()输入。

【课堂案例 5-2】 学生结构体中成员的引用。

```
#include<stdio.h>
#define Student struct student
#include<string.h>
Student{
char name[20];           //学生姓名
int num;                 //学生学号
int age;                 //学生年龄
};
int main()
{
Student s1;
strcpy(s1.name,"张三"); //利用字符串函数 strcpy()给 s1.name 赋值
s1.num=12;
s1.age=13;
printf("学生的学号为:%d,年龄为:%d,姓名为:%s",s1.num,s1.age,s1.name);
}
```

运行结果:

学生的学号为:12,年龄为:13,姓名为:张三

【课堂案例 5-3】 定义一个包含学生学号、姓名、性别及 3 门课的成绩的结构体变量,从键盘输入具体数据,并输出学生的基本信息及平均成绩。

```
#include<stdio.h>
#include<string.h>
struct Student
{
    char num[20];
    char name[12];
    char sex[2];
    double score[3];
};
int main()
{
    struct Student stu;
    char x[12]="张三";
    float y[3]={85,75,75.5},sum=0;
    int i;
    strcpy(stu.num,"202011201");/* 如果成员名是一个变量名,那么引用的就是这个变量的内容;如果成员名是一个数组名,那么引用的就是这个数组的首地址 */
    /* stu.s="202011201";这样写是错误的,相当于通过数组名给数组赋值,数组名是一个整数常量 */
    strcpy(stu.name,x);
    strcpy(stu.sex,"f");
    for( i=0;i<3;i++)
    {
        stu.score[i]=y[i];
        sum=sum+stu.score[i];
    }
    printf("该学生的基本信息为:\n 学号:%s, 姓名:%s, 性别:%s, 平均成绩为:%f", stu.num,
    stu.name, stu.sex, sum/3);
    return 0;
}
```

运行结果:

该学生的基本信息为:
学号:202011201, 姓名:张三, 性别:f, 平均成绩为:78.500000



观看视频

任务 5.2 结构体数组及应用

根据实际需求,某个数组中需要存储一些结构体数据,例如定义一个学生数组,包含 10 个学生信息,学生信息中包含一系列的复杂信息,这时就需要用到结构体数组。

【课堂案例 5-4】 利用数组和结构体完成学生成绩问题。有 4 个学生,每个学生的数据包括学号、姓名、3 门课的成绩。从键盘输入 4 个学生数据,要求输出每门课的总平均成绩及最高分学生的数据(用结构体数组实现)。

```

#include<stdio.h>
#include<string.h>
#define N 4
struct Student
{
    char num[20];
    char name[12];
    double score[3];
};
int main()
{
    struct Student stu[N];
    int i, j;
    double sum, max;
    printf("请输入所有学生的信息\n");
    printf("学号    姓名    成绩 1    成绩 2    成绩 3    \n");
    for(i=0; i<N; i++)
        //输入 4 名学生的成绩
    {
        scanf("%s %s", stu[i].num, stu[i].name);
        for(j=0; j<3; j++)
            scanf("%lf", &stu[i].score[j]);    //输入 3 个成绩
    }
    //输出所有学生的基本信息
    printf("所有学生的信息如下:\n");
    printf("学号    姓名    成绩 1    成绩 2    成绩 3    \n");
    for(i=0; i<N; i++)
        //输出 4 名学生的基本信息
    {
        printf("%s\t%s\t", stu[i].num, stu[i].name);
        for(j=0; j<3; j++)
            printf("%.2f\t", stu[i].score[j]);
        printf("\n");
    }
    printf("\n");
    //计算每门课的平均成绩和最高分
    printf("\t\t\t 平均成绩\t 最高分\n");
    for(j=0; j<3; j++)
    {
        printf("第%d 门课成绩\t", j+1);
        sum=0;
        max=-1;
        //给 max 一个初值-1, 保证输入的成绩都比 max 大
        //计算每门课
        for(i=0; i<N; i++)
        {
            sum=sum+stu[i].score[j];
            if(max<stu[i].score[j])
                max=stu[i].score[j];
        }
        printf("%f\t%f\n", sum/N, max);
    }
}

```

运行结果：

请输入所有学生的信息

学号	姓名	成绩 1	成绩 2	成绩 3
01	张三	75	74	71
02	李四	74	70	85
03	王五	85	90	94
04	赵六	74	85	90

所有学生的信息如下：

学号	姓名	成绩 1	成绩 2	成绩 3
01	张三	75.00	74.00	71.00
02	李四	74.00	70.00	85.00
03	王五	85.00	90.00	94.00
04	赵六	74.00	85.00	90.00

	平均成绩	最高分
第 1 门课成绩	77.000000	85.000000
第 2 门课成绩	79.750000	90.000000
第 3 门课成绩	85.000000	94.000000

任务实现

完成课程信息的存储,并输出课程的信息情况。

运行结果如下：

请输入第 1 个学生的基本信息：

课程编号:1111

课程名称:数据结构

课程成绩:85

教师姓名:李莉

请输入第 2 个学生的基本信息：

课程编号:2222

课程名称:C 语言程序设计

课程成绩:93

教师姓名:王超

请输入第 3 个学生的基本信息：

课程编号:3333

课程名称:高等数学

课程成绩:82

教师姓名:李燕

请输入第 4 个学生的基本信息：

课程编号:4444

课程名称:大学英语

课程成绩:75

教师姓名:刘静

请输入第 5 个学生的基本信息：

课程编号:5555

课程名称:Java 程序设计

课程成绩:88

教师姓名:姚建

当前课程信息如下:

课程编号	课程名称	课程成绩	教师姓名
1111	数据结构	85.00	李莉
2222	C 语言程序设计	93.00	王超
3333	高等数学	82.00	李燕
4444	大学英语	75.00	刘静
5555	Java 程序设计	88.00	姚建

1. 实现课程信息的结构体定义

代码如下:

```
typedef struct course
{
    char courseno[10];           //课程编号
    char coursename[50];        //课程名称
    double course_score;         //课程成绩
    char teachername[10];       //任课教师信息,这里只展示教师姓名(学习了指针之后,再进行调整)
}Course;
```

2. 课程信息的录入与输出

代码如下:

```
#include<stdio.h>
#include<string.h>
int main()
{
    Course c1[5];
    int i;
    for(i=0;i<5;i++)
    {
        printf("请输入第%d个学生的基本信息:\n",i+1);
        printf("课程编号:");scanf("%s",c1[i].courseno);
        printf("课程名称:");scanf("%s",c1[i].coursename);
        printf("课程成绩:");scanf("%lf",&c1[i].course_score);
        printf("教师姓名:");scanf("%s",c1[i].teachername);
    }
    printf("当前课程信息如下:\n");
    printf("课程编号\t课程名称\t课程成绩\t教师姓名\n",i+1);
    for(i=0;i<5;i++)
    {
        printf("%s",c1[i].courseno);
        printf("\t\t%s",c1[i].coursename);
        printf("\t\t%.2f",c1[i].course_score);
        printf("\t\t%s",c1[i].teachername);
        printf("\n");
    }
    return 0;
}
```