



什么是结构体

在实际问题中，一组数据往往具有不同的数据类型。例如，人口大普查时，我们需要记录每一位公民的姓名，年龄，性别，住址，身份证号码。这些信息分别要用字符型，整型，布尔型等来记录。

数据项	数据项定义
姓名	最长 10 个字符
年龄	一个非负整数
性别	0 表示男；1 表示女
家庭住址	最长 100 个字符
身份证号码	最长 18 个字符

有没有简单清晰的方式定义数据类型，从而描述每个人的数据信息呢？

我们可以用结构体的方式来进行数据描述：

```
1. //定义类型名叫 Person 的 struct 类型
2. struct Person
3. {
4.     char name[11]; //姓名
5.     int age; //年龄
6.     bool gender; //性别
7.     char address[101]; //地址
8.     char id[19]; //身份证号码
9. };
```

结构体的语法形式如下：

```
1. //其中 struct 是关键字
2. struct 结构体类型名
3. {
4.     成员表; //可以有多个成员
5. };
```



结构体和结构体变量的定义

结构体变量的定义有两种形式：

- 先定义结构体再定义结构体变量；
- 定义结构体类型的同时定义变量。

【先定义结构体再定义结构体变量】

```
1. struct 结构体类型名
2. {
3.     成员表; //可以有多个成员
4. };
5. //可以同时定义多个结构体变量
6. 结构体类型名 结构体变量表;
```

例如：

```
1. struct Person
2. {
3.     char name[11]; //姓名
4.     int age; //年龄
5.     bool gender; //性别
6.     char address[100]; //地址
7.     char id[20]; //身份证号码
8. };
9. Person a[10], b;
```

【定义结构体类型的同时定义变量】

```
1. struct 结构体类型名
2. {
3.     成员表; //可以有多个成员
4. } 结构体变量表; //可以同时定义多个结构体变量
```

例如：

```
1. struct Person
2. {
3.     char name[11]; //姓名
4.     int age; //年龄
5.     bool gender; //性别
```



```
6.     char address[100]; //地址
7.     char id[20]; //身份证号码
8. } a[10], b; //定义 a 数组变量, b 变量
```

【总结】

定义结构体变量时，结构体变量名和结构体类型名不能相同。

定义结构体时，系统不分配实际内存。

定义结构体变量时，系统才分配内存。

【数组与结构体】

数组用来存储一组相同类型数据项的变量。

结构体是用户自定义的数据类型，允许存储不同类型的数据项。

数组和结构体结合，就可以写出功能非常强大的程序。



结构体变量的输入、输出、赋值

【使用方式】

结构体变量各个成员的引用的一般形式为：

1. 结构体变量名.成员名

【问题描述】

输入 1 个学生的姓名和语文、数学的得分，并计算出总分，然后输出。

【输入格式】

1 行，1 个姓名，2 个整数。姓名由不超过 10 个的小写字母组成，整数范围是[0…100]。

【输出格式】

1 行，每行格式：姓名 语文 数学 总分。

【输入样例】

```
Jianming 70 99
```

【输出样例】

```
Jianming 70 99 169
```

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. //定义类型名为 Student 的 struct 类型
4. struct Student
5. {
6.     char name[11];
7.     int chinese;
8.     int math;
9.     int total;
10. };
11. int main()
12. {
13.     Student a;
14.     //输入结构体数据
```



```
15.    cin >> a.name >> a.chinese >> a.math;
16.    //结构体数据项赋值
17.    a.total = a.chinese + a.math;
18.    //输出结构体数据
19.    cout << a.name << ' ' << a.chinese << ' ' << a.math << ' ' << a.total <<
    endl;
20.    return 0;
21. }
```



结构体数组变量的输入、输出、赋值

【问题描述】

输入 3 个学生的姓名和语文、数学的得分，并计算出总分，然后输出。

【输入格式】

3 行。每行 1 个姓名，2 个整数。姓名由不超过 10 个的小写字母组成，整数范围是[0…100]。

【输出格式】

3 行，每行格式：姓名 语文 数学 总分。

【输入样例】

```
Jianming 70 99
Yuhong 83 87
wei 65 89
```

【输出样例】

```
Jianming 70 99 169
Yuhong 83 86 170
wei 65 89 154
```

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. //定义 struct 的类型，类型名: Student
4. struct Student
5. {
6.     char name[11];
7.     int chinese;
8.     int math;
9.     int total;
10. };
11. int main()
12. {
13.     Student a[3];
14.     //对结构体中成员的赋值、取值。
15.     for (int i=0; i<3; i++)
16.     {
```



```
17.         cin >> a[i].name >> a[i].chinese >> a[i].math;
18.         a[i].total = a[i].chinese + a[i].math;
19.     }
20.     //输出
21.     for (int i=0; i<3; i++)
22.     {
23.         cout << a[i].name << ' ' << a[i].chinese << ' ' << a[i].math << ' '
            << a[i].total << endl;
24.     }
25.     return 0;
26. }
```



结构体的嵌套

成员运算符“.”在存取成员数值时使用，其优先级最高，并具有左结合性。在处理包含结构体的结构体时，可记作：

```
1. a.b.c
```

这说明结构体变量 a 有结构体成员 b；结构体变量 b 有成员 c。

【参考程序】

```
1. #include<iostream>
2. #include<cstring>
3. using namespace std;
4. struct Date
5. {
6.     int year;
7.     int month;
8.     int day;
9. };
10. struct Person
11. {
12.     char name[11];
13.     Date birthday;
14. };
15. int main()
16. {
17.     Person p;
18.     strcpy(p.name, "Jianming");
19.     p.birthday.year = 1977;
20.     p.birthday.month = 12;
21.     p.birthday.day = 1;
22.     cout << p.name << ' ' << p.birthday.year << ' ' << p.birthday.month << ' '
        << p.birthday.day << endl;
23.     return 0;
24. }
```




结构体变量的初始化

结构体变量的初始化和数组的初始化类似。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. struct Student
4. {
5.     char name[11];
6.     int chinese;
7.     int math;
8.     int total;
9. } a[2] = {
10.     {"Jianming", 90, 90, 180},
11.     {"Wei", 80, 90, 170}
12. };
13. int main()
14. {
15.     Student b = {"Yuhong", 90, 80, 170};
16.     cout << b.name << ' ' << b.chinese << ' ' << b.math << ' ' << b.total << endl;
17.     for (int i=0; i<2; i++)
18.     {
19.         cout << a[i].name << ' ' << a[i].chinese << ' ' << a[i].math << ' ' << a[i].total << endl;
20.     }
21.     return 0;
22. }
```



结构体变量的交换

结构体变量可以整体操作，进行交换。

【参考程序】

```
1. #include<iostream>
2. #include<algorithm>
3. using namespace std;
4. struct Student
5. {
6.     char name[11];
7.     int chinese;
8.     int math;
9.     int total;
10. } a[2] = {
11.     {"Jianming", 90, 90, 180},
12.     {"Wei", 80, 90, 170}
13. };
14. int main()
15. {
16.     swap(a[0], a[1]);
17.     for (int i=0; i<2; i++)
18.     {
19.         cout << a[i].name << ' ' << a[i].chinese << ' ' << a[i].math << ' '
20.         << a[i].total << endl;
21.     }
22.     return 0;
23. }
```