

1.3.5结构体

- 结构(structure) 属于构造类型，它是由多种类型的数据成分组成的数据类型。
- 区别以前所介绍过的数据类型，结构的具体构成需要用户自己定义，即定义一个结构类型。
- 结构中的每个数据项称为成员。
- 成员的类型可以是基本数据类型，也可以是一个诸如结构这样的构造类型（嵌套==>无限）。

结构体定义的一般形式为：

```
1 struct <结构体名> {  
2     <数据类型> <成员1>;  
3     <数据类型> <成员2>;  
4     <数据类型> <成员n>;  
5 };
```

结构体变量的声明形式如下:

```
1 <结构体名> <结构体变量名>;
```

结构体变量的使用:

```
1 <结构体变量名>.<成员名>  
2 <结构体变量指针> -> <成员名>
```


例子

```
1 //定义结构体  
2 struct staffer {  
3     int ID;  
4     char name[20];  
5     bool sex;  
6     float aalary;
```


```
1 #include<bits/stdc++.h>  
2 using namespace std;  
3  
4 //定义结构体  
5 struct staffer {  
6     int ID;  
7     char name[20];  
8     bool sex;  
9     float salary;  
10 };  
11  
12 void eg2_11_0() {  
13     staffer employee1 = {110105, "LiMing", 1, 3809.80};  
14     staffer employee2 = employee1;  
15  
16     strcpy(employee2.name, "WangPing");  
17     employee2.ID = 110106;
```

```
18 cout << employee1 << endl; // 正常会报错，无法直接输出结构体成员
19 cout << employee2 << endl;
20
21 staffer* pStaff = &employee2;
22 pStaff->salary += 1000;
23 cout << employee1 << endl;
24 cout << employee2 << endl;
25 }
26
27 int main() {
28     eg2_11_0();
29     return 0;
30 }
```

结构体：例子*



```
ostream& operator<<(ostream& os, staffer staff1) {
    os << "(";
    os << staff1.ID << ", " << staff1.name << ", ";
    os << staff1.sex << ", " << staff1.salary;
    os << ")";
    return os;
}
```



```
cout << employee1 << endl;
(110105, LiMing, 1, 3809.8)
```