

# C++ class和struct到底有什么区别

C++ 中保留了C语言的 struct 关键字，并且加以扩充。在C语言中，struct 只能包含成员变量，不能包含成员函数。而在C++中，struct 类似于 class，既可以包含成员变量，又可以包含成员函数。

C++中的 struct 和 class 基本是通用的，唯有几个细节不同：

- 使用 class 时，类中的成员默认都是 private 属性的；而使用 struct 时，结构体中的成员默认都是 public 属性的。
- class 继承默认是 private 继承，而 struct 继承默认是 public 继承（《C++继承与派生》一章会讲解继承）。
- class 可以使用模板，而 struct 不能（《模板、字符串和异常》一章会讲解模板）。

C++ 没有抛弃C语言中的 struct 关键字，其意义就在于给C语言程序开发人员有一个归属感，并且能让C++编译器兼容以前用C语言开发出来的项目。

在编写C++代码时，我强烈建议使用 class 来定义类，而使用 struct 来定义结构体，这样做语义更加明确。

使用 struct 来定义类的一个反面教材：

```
01. #include <iostream>
02. using namespace std;
03.
04. struct Student{
05.     Student(char *name, int age, float score);
06.     void show();
07.
08.     char *m_name;
09.     int m_age;
10.     float m_score;
11. };
12.
13. Student::Student(char *name, int age, float score): m_name(name), m_age(age),
    m_score(score){ }
14. void Student::show(){
15.     cout<<m_name<<"的年龄是"<<m_age<<"，成绩是"<<m_score<<endl;
16. }
17.
18. int main(){
19.     Student stu("小明", 15, 92.5f);
20.     stu.show();
21.     Student *pstu = new Student("李华", 16, 96);
```

```
22.     pstu -> show();  
23.  
24.     return 0;  
25. }
```

运行结果：

小明的年龄是15，成绩是92.5

李华的年龄是16，成绩是96

这段代码可以通过编译，说明 struct 默认的成员都是 public 属性的，否则不能通过对象访问成员函数。如果将 struct 关键字替换为 class，那么就会编译报错。