

# C++ const成员变量和成员函数（常成员函数）

在类中，如果你不希望某些数据被修改，可以使用 `const` 关键字加以限定。`const` 可以用来修饰成员变量和成员函数。

## const成员变量

`const` 成员变量的用法和普通 `const` 变量的用法相似，只需要在声明时加上 `const` 关键字。初始化 `const` 成员变量只有一种方法，就是通过构造函数的初始化列表，这点在前面已经讲到了，请猛击[《C++初始化列表》](#)回顾。

## const成员函数（常成员函数）

`const` 成员函数可以使用类中的所有成员变量，但是不能修改它们的值，这种措施主要还是为了保护数据而设置的。`const` 成员函数也称为[常成员函数](#)。

我们通常将 `get` 函数设置为常成员函数。读取成员变量的函数的名字通常以 `get` 开头，后跟成员变量的名字，所以通常将它们称为 `get` 函数。

常成员函数需要在声明和定义的时候在[函数头部的结尾](#)加上 `const` 关键字，请看下面的例子：

```
01. class Student{
02. public:
03.     Student(char *name, int age, float score);
04.     void show();
05.     //声明常成员函数
06.     char *getname() const;
07.     int getage() const;
08.     float getscore() const;
09. private:
10.     char *m_name;
11.     int m_age;
12.     float m_score;
13. };
14.
15. Student::Student(char *name, int age, float score): m_name(name), m_age(age),
    m_score(score){ }
16. void Student::show(){
17.     cout<<m_name<<"的年龄是"<<m_age<<"，成绩是"<<m_score<<endl;
18. }
19. //定义常成员函数
20. char * Student::getname() const{
```

```
21.     return m_name;
22. }
23. int Student::getage() const{
24.     return m_age;
25. }
26. float Student::getscore() const{
27.     return m_score;
28. }
```

getname()、getage()、getscore() 三个函数的功能都很简单，仅仅是为了获取成员变量的值，没有任何修改成员变量的企图，所以我们加了 const 限制，这是一种保险的做法，同时也使得语义更加明显。

需要强调的是，必须在成员函数的声明和定义处同时加上 const 关键字。char \*getname() const 和 char \*getname() 是两个不同的函数原型，如果只在一个地方加 const 会导致声明和定义处的函数原型冲突。

最后再来区分一下 const 的位置：

- 函数开头的 const 用来修饰函数的返回值，表示返回值是 const 类型，也就是不能被修改，例如 const char \* getname()。
- 函数头部的结尾加上 const 表示常成员函数，这种函数只能读取成员变量的值，而不能修改成员变量的值，例如 char \* getname() const。