对象作为返回值, 也会有拷贝构造

1. 如果外部没有接受函数的对象,该返回的对象是个无名对象,当遇到该行的";"时,析构。

```
1 foo_rtn().foo_test() , printf("123\r\n");printf("456\r\n");
2 123
3 ~CTest() -->说明遇到;析构
4 456
```

```
/*
1
2
  对象作为参数
  Object as a parameter
4
  */
5
  #include "stdafx.h"
6
  #include <iostream>
8
  //创建一个测试类用于创建对象
9
10 class CTest {
11 public:
12
      //构造函数
       CTest(int n = 1)
13
           :m_nTest(n)
14
15
       {
           std::cout << "CTest()" << std::endl;</pre>
16
       }
17
18
       //拷贝构造
19
       CTest(CTest& obj)
20
       {
21
           std::cout << "CTest(CTest& obj)" << std::endl;</pre>
22
           //给对象成员赋值
23
           m_nTest = obj.m_nTest;
24
25
       }
26
       //=运算符重载
27
28
       CTest& operator=(CTest& obj)
```

```
29
       {
           std::cout << "CTest& operator=(CTest& obj)" << std::endl;</pre>
30
31
           m_nTest = obj.m_nTest;
32
       }
33
       //析构函数
34
       ~CTest()
35
       {
36
37
           std::cout << "~CTest()" << std::endl;</pre>
       }
38
39
       //常成员函数
40
41
       void foo_test() const
42
       {
           //m_nTest = 1; // 常成员函数不允许修改成员值
43
           std::cout << "m_nTest = " << m_nTest << std::endl;</pre>
44
       }
45
46
       //普通成员函数
47
       void foo_test1()
48
       {
49
           m_nTest = 2; //普通成员函数可以修改成员值
50
           std::cout << "m_nTest = " << m_nTest << std::endl;</pre>
51
       }
52
53
   private:
54
55
       int m_nTest;
56 };
57
58 //对象作为返回值,也会发生拷贝构造
59 CTest foo_rtn()
60 {
61
       CTest t(1);
       return t;
62
63 }
64
65 int main()
66 {
       foo_rtn();
67
68
       return 0;
69
70 }
```

```
File Edit Selection View Go Debug Terminal Help
                                                                                                                                                                                                         test.cpp - 08copy_construct - Visual Studio Code
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          ė II
                    G→ Copy.cpp G→ TestArg.cpp G→ test.cpp X
   3
                                                                                                                                                                                                                                                                                                                                                                                                                                                                  ET Comment and Com
                                         .
对象作为参数
   Q
                                         Object as a parameter
   99
                                         #include "stdafx.h"
                                         #include <iostream>
   8
                                         //创建一个测试类用于创建对象
   11
                                         public:
                                                  //构造函数
CTest(int n = 1)
                                                           :m_nTest(n)
                          14
15
16
17
                                                  18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
                                                    //拷贝构造
                                                    CTest(CTest& obj)
                                                           std::cout << "CTest(CTest& obj)" << std::endl;
                                                                //给对象成员赋值
                                                          m_nTest = obj.m_nTest;
                                                     //=运算符重载
                                                    CTest& operator=(CTest& obj)
                                                  {
    std::cout << "CTest& operator=(CTest& obj)" << std::endl;
    m_nTest = obj.m_nTest;
                                                    //析构函数
                                                    ~CTest()
                                                  36
37
38
39
40
41
42
                                                    //常成员函数
                                                    void foo_test() const
                                                  43
                         44
45
                         46
47
48
                                                    //普通成员函数
                                                    void foo_test1()
                          49
                                                         m_nTest = 2; //普通成员函数可以修改成员值
std::cout << "m_nTest = " << m_nTest << std::endl;
                         50
51
                         52
53
54
                         55
56
                                                 int m_nTest;
                                         };
                         57
                                         (名)

分分象作为返回值,也会发生拷贝构造

CTest foo_rtn()
                         58
59
                         60
61
                                        return t;
                         62
                                                                                                                                                                                                    过程分析:
                         64
                                                                                                                              output:
CTest()
                                                                                                                                                                                                   1 调用函数foo_trn(),
1-1 CTest t(1),调用类的构造函数,输出CTest()
1-2 return t,返回这个类的对象,调用拷贝构造,输出CTest(CTest& obj)
1-3 返回,交还控制权,调用析构函数,释放t,输出-CTest()
2 程序执行完毕,调用析构函数,输出-CTest()
                         65
66
67
                                         int main()
                                                                                                                              CTest(CTest& obj)
~CTest()
~CTest()
                                             1 foo_rtn();
                          68
69
                                             2 return 0;
                         70
71
72
                                                                                                                                                                                                                                                                                                                                                                                                                    激活 Windows
                                                                                                                                                                                                                                                                                                                                                                                                                   转到"设置"以激活 Windows.
```