```
/*
1
  对象作为参数
  Object as a parameter
   */
4
5
   #include "stdafx.h"
   #include <iostream>
7
8
   //创建一个测试类用于创建对象
10 class CTest {
11 public:
       //构造函数
12
13
       CTest(int n = 1)
14
           :m_nTest(n)
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
24
           m_nTest = obj.m_nTest;
25
       }
26
       //析构函数
27
       ~CTest()
28
29
       {
           std::cout << "~CTest()" << std::endl;</pre>
30
31
       }
32
       //常成员函数
33
       void foo_test() const
34
35
       {
           //m_nTest = 1; // 常成员函数不允许修改成员值
36
37
           std::cout << "m_nTest = " << m_nTest << std::endl;</pre>
       }
38
39
       //普通成员函数
40
```

```
41
       void foo_test1()
42
       {
          m_nTest = 2; //普通成员函数可以修改成员值
43
          std::cout << "m_nTest = " << m_nTest << std::endl;</pre>
44
       }
45
46
47 private:
       int m_nTest;
48
49 };
50
51 //引用对象作为函数参数(const T&),调用常成员函数
52 void foo_argRef(const CTest& tRefArg)
53 {
54
       tRefArg.foo_test();
       //tRefArg.foo_test1(); //被调用的函数也必须是 const 常成员函数
55
56 }
57
58 int main()
59 {
      //CTest t = 1; //隐式转换调用构造函数
60
       CTest t; //创建一个对象,调用CTest构造函数
61
       foo_argRef(t);
62
63
       return 0;
64
65 }
66 /*
67 output:
68 CTest()
69 \text{ m_nTest} = 1
70 ~CTest()
71 */
72
```

```
File Edit Selection View Go Debug Terminal Help test.cpp - 08copy_construct - Visual Studio Code
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               \frac{1}{2} \left( \frac{1}{2} \right) \right) \right) \right) \right)}{(1 \right)} \right) \right)} \right) \right)}{(1 \right)} \right)} \right)} \right) \right)} \right) \right) \right) \right) \right) \right) \right) \\
                          a
                                                  ,
对象作为参数
     م
                                                  Object as a parameter
    6
                                                  #include "stdafx.h"
   (8)
                                                  //创建一个测试类用干创建对象
                                                 class CTest {
public:
//构造函数
    11
                                12
                                14
                                                                       :m_nTest(n)
                                15
16
17
18
19
20
                                                                       std::cout << "CTest()" << std::endl;
                                                               CTest(CTest& obj)
                                                              {
    std::cout << "CTest(CTest& obj)" << std::endl;
    //给对象成员账值
                                22
                               23
24
25
26
27
28
29
30
                                                                      m_nTest = obj.m_nTest;
                                                                //析构函数
                                                                ~CTest()
                                                              31
32
                                                                //常成员函数
                                33
34
35
36
37
38
                                                                 void foo_test() const
                                                              39
40
                                                                //普通成员函数
                               41
                                                                void foo_test1()
                                42
43
                                                                    m_nTest = 2; //普通成员函数可以修改成员值
std::cout << "m_nTest = " << m_nTest << std::endl;
                                44
                               46
                               47
48
                                                            int m_nTest;
                               49
                               50
51
                                                  //引用对象作为函数参数(const-T&),调用学成员函数
                               52
                                                  void foo_argRef const CTest& tRefArg
                                                 tRefArg.foo_test();
                                54
                                                        //tRefArg.foo_test1(); //被调用的函数也必须是 const 常成员函数
                                58
59
                                                                                                                                                                                                                                 过程分析:
1 CTest t;创建对象,调用类的构造函数,输出CTest()
2 foo_argRef(t),函数定义,由于是对CTest这个类对象的引用,编译器优化不在调用拷贝构造。
2-1 直接调用函数foo_test():
这里要注意,const引用的类对象,调用的成员函数也必须是常成员函数const
2-2 输出m_nTest=1,返回2-1,这里也要注意,引用的类对象,不再发生析构
3 调用析构函数,释放t
                                                 60
61
                               62
                                                   foo_argRef(t);
                                                  3 return 0;
                               65
66
67
                                                  output:
                                                 CTest()
m_nTest = 1
                                70
                                                  ~CTest()
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             激活 Windows
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             转到"设置"以激活 Windows
```