以钟表为例。定义钟表数据和行为。

头文件 clock.h

```
#pragma once
1
2 class tagClock {
  private:
4
      //数据
      int nHour;
5
6
       int nMinute;
       int nSecond;
8
9
  public:
       //行为,函数声明
10
       int __stdcall SetTime(int nH, int nM, int nS);
11
12 };
```

实现头文件中定义的成员函数,源文件 clock.cpp

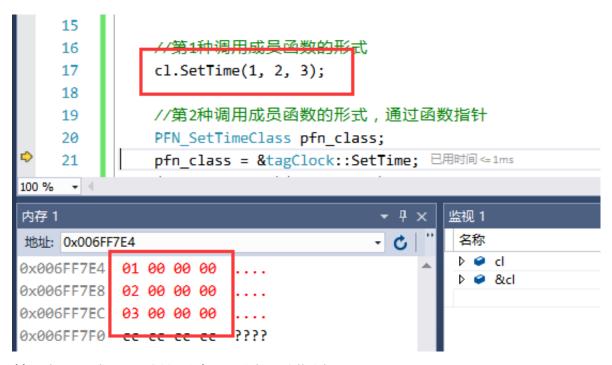
```
#include "stdafx.h"
  #include "clock.h"
2
3
  //函数定义,需要加上结构体名作用域
4
  int tagClock::SetTime(int nH, int nM, int nS) {
5
      //this指针表示当前调用该函数的对象的地址
6
7
      this->nHour = nH;
      nMinute = nM;
8
9
      nSecond = nS;
10
      return 0;
11
12 }
```

实现所有函数后,主函数中调用 main.cpp

```
1 #include "stdafx.h"
```

```
#include "clock.h"
3
  //定义类中的函数指针
4
5 typedef int(__stdcall tagClock::*PFN_SetTimeClass)(int nH, int
   nM, int nS);
  //参数传递:寄存器 栈
  // thiscall --> 用于类的成员函数的调用约定, 编译器内部使用
7
  // fastcall
8
  // cdecl
9
10 // stdcall
11
12 int main()
13 {
14
      tagClock cl;
15
16
      //cl.nHour = 1; //不可访问
17
      //第1种调用成员函数的形式
18
      cl.SetTime(1, 2, 3);
19
20
      //第2种调用成员函数的形式,通过函数指针
21
      PNF_SetTimeClass pfn_class;
22
      pfn_class = &tagClock::SetTime;
23
      (cl.*pfn_class)(11, 21, 31);
24
25
      //第3种调用成员函数的形式,通过对类对象的引用
26
27
      tagClock* pCL = &cl;
      //PNF_SetTimeClass pfn_class;
28
      //pfn_class = &tagClock::SetTime;
29
      (pCL->*pfn_class)(10, 20, 30);
30
31
32
      return 0;
33 }
```

• 第1种调用成员函数的形式



• 第2种调用成员函数的形式,通过函数指针

```
//第2种调用成员函数的形式,通过函数指针
    19
    20
              PFN_SetTimeClass pfn_class;
    21
              pfn class = &tagClock::SetTime;
    22
              (cl.*pfn_class)(11, 21, 31);
    23
    24
              //第3种调用成员函数的形式,通过对函数对象的引用
              tagClock* pCL = &cl; 已用时间<=1ms
    25
100 %
                                       ▼ 早 × 监视 1
地址: 0x006FE7F4
                                              名称
                                      - 0
                                                                                  {nHour=11 nMinute=21 nSecond=31 }
0x006FF7F4 0b 00 00 00
                                                                                  0x006ff7e4{nHour=11 nMinute=21 nSecond=31}
                                               0x006FF7F8 15 00 00 00
0x006FF71C 1f 00 00 00
           cc cc cc cc
```

• 第3种调用成员函数的形式,通过对类对象的引用

```
//第3种调用成员函数的形式,通过对类对象的引用
    24
    25
              tagClock* pCL = &cl;
              //PNF_SetTimeClass pfn_class;
    26
    27
              //pfn class = &tagClock::SetTime;
    28
              (pCL->*pfn_class)(10, 20, 30);
    29
              return 0; 已用时间<=1ms
    30
    31
00 %
内存 1
                                               监视 1
地址: 0x006FF<u>7E4</u>
                                       - 0
                                                                                    {nHour=10 nMinute=20 nSecond=30 }
                                                x006FF7E4
          0a 00 00 00
                                                14 00 00 00 .....
x006FF7E8
x006FF7EC
          1e 00 00 00
1x006FF7F0 cc cc cc cc ?????
1x006FF7F4 2a 11 84 ac
x006FF7F8 0c f8 6f 00 .?o.
```