

```
1 // TestCoursework.cpp : 定义控制台应用程序的入口点。
2 //
3
4 #include <stdio.h>
5
6 class tagClock {
7
8 private:
9     //数据
10    int nHour;
11    int nMinute;
12    int nSecond;
13
14 public:
15     //行为, 函数声明, 内联
16    int SetTime(int nH, int nM, int nS) {
17        nHour = nH;
18        nMinute = nM;
19        nSecond = nS;
20
21        return 0;
22    }
23 };
24
25 //类中的函数指针
26 typedef int(tagClock::*PFN_SetTimeClass)(int nH, int nM, int nS);
27
28 int main()
29 {
30     tagClock cl1;
31     tagClock cl2;
32
33     PFN_SetTimeClass pfn1 = cl1.SetTime;
34     PFN_SetTimeClass pfn2 = cl2.SetTime;
35
36     // 地址是一样的, 表示同一个类的对象的成员函数是共用的
37     // (1) 数据是独立的
38     // (2) 成员函数是共用的
39     cl1.SetTime(1, 2, 3);
40     cl2.SetTime(4, 5, 6);
```

```

41
42     printf("pfn1 = %p\r\n", pfn1);
43     printf("pfn2 = %p\r\n", pfn2);
44     return 0;
45 }

```

- 函数指针都是指向的同一个类的成员函数，成员函数共用

The screenshot shows a debugger window with the following code:

```

PFN_SetTimeClass pfn1 = cl1.SetTime;
PFN_SetTimeClass pfn2 = cl2.SetTime;

```

Below the code is a memory dump window titled "Address: 0x0019FF24". The dump shows hexadecimal values for various memory addresses. Two specific values are highlighted with red boxes and labels:

- At address 0019FF14, the value is 05 10 40 00, labeled "pfn1".
- At address 0019FF24, the value is 05 10 40 00, labeled "pfn2".

- 数据独立

The screenshot shows a debugger window with the following code:

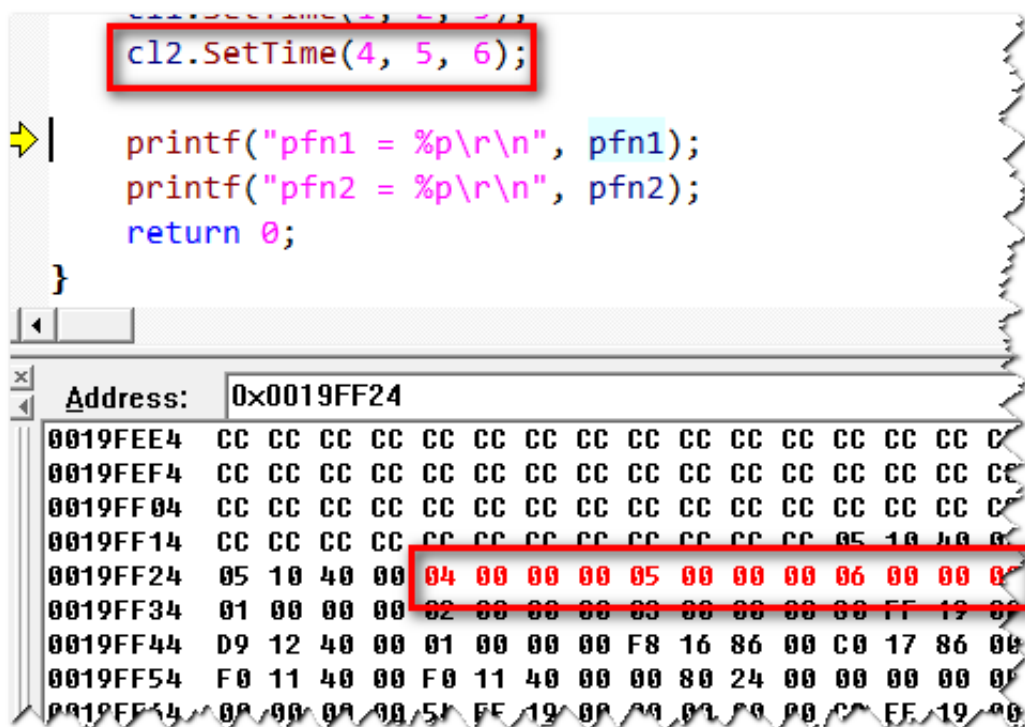
```

// (4) 成员函数是不用()
cl1.SetTime(1, 2, 3);
cl2.SetTime(4, 5, 6);

```

Below the code is a memory dump window titled "Address: 0x0019FF24". The dump shows hexadecimal values for various memory addresses. Two specific values are highlighted with red boxes:

- At address 0019FF14, the value is 05 10 40 00.
- At address 0019FF24, the value is 01 00 00 00 02 00 00 00 03 00 00 00.



Visual Studio 系列编译不通过。