菱形继承无虚继承方式消除了不数据冗余，于是就引进了虚继承的概念。
关键字virtual

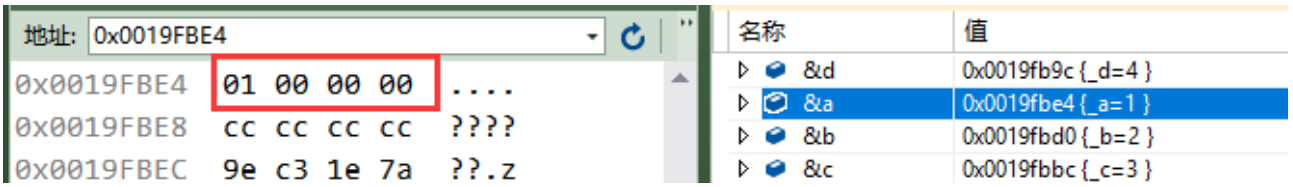
示例代码：

```
// 5_1 菱形继承有虚继承无虚函数.cpp
//

#include "stdafx.h"
#include <iostream>
using namespace std;

class A {
public:
    A() {
        _a = 1;
    };
public:
    int _a;
};

class B : virtual public A {
public:
    B() {
        _b = 2;
    }
public:
    int _b;
};

class C : virtual public A {
public:
    C() {
        _c = 3;
    }
public:
    int _c;
};

class D :public B, public C {
```

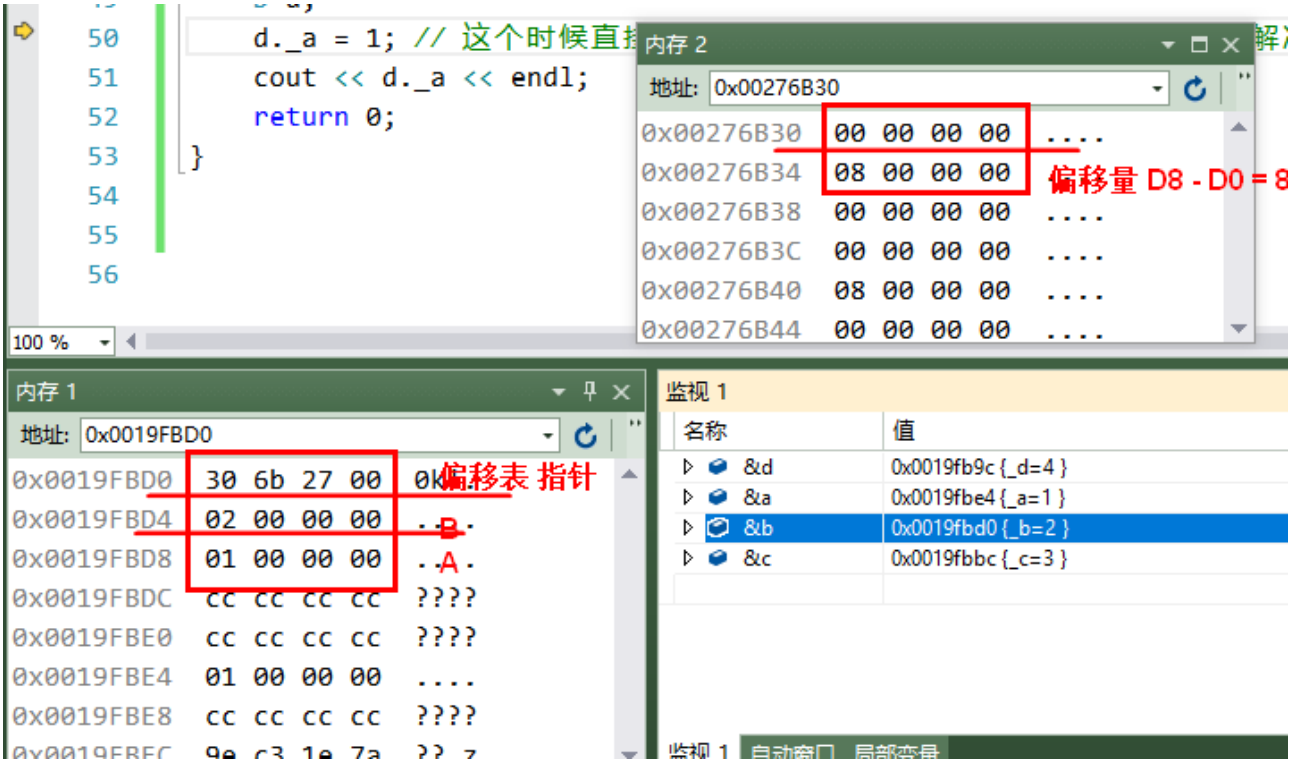
```
public:
    D() {
        _d = 4;
    }
public:
    int _d;
};

int main()
{
    A a;
    B b;
    C c;
    D d;
    d._a = 1; // 这个时候直接访问就可以了，d对象里面只有一个_a了，解决了数据冗余问题
    cout << d._a << endl;
    return 0;
}
```

对象内存分析

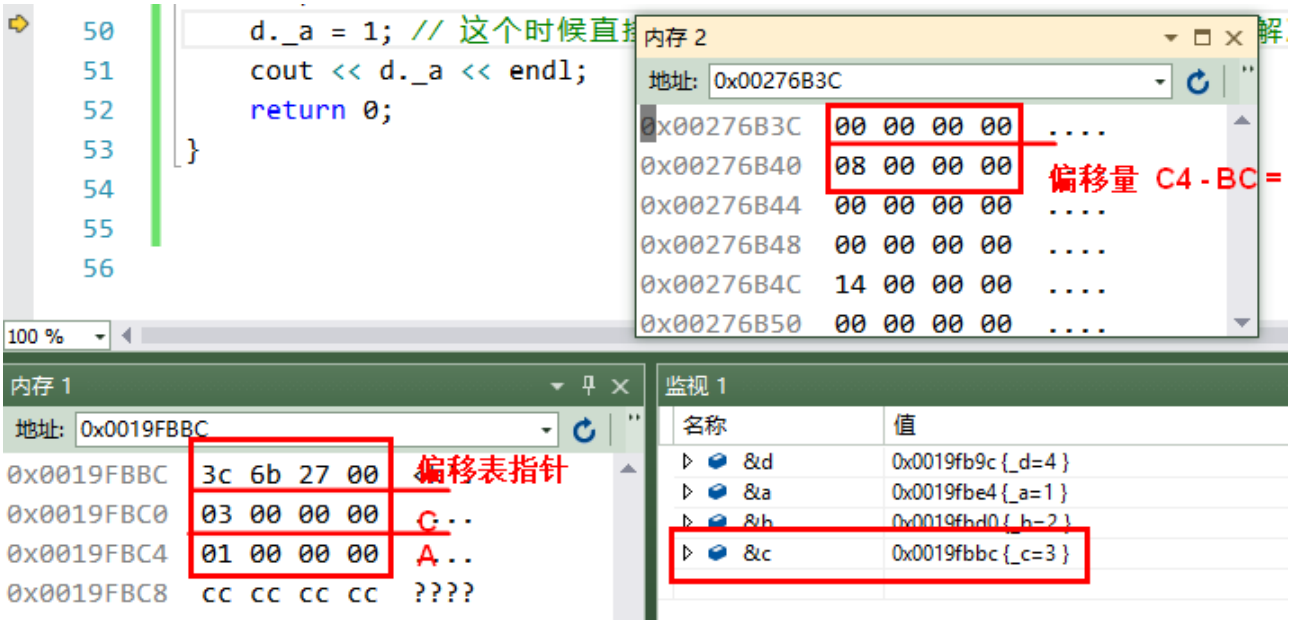
A a



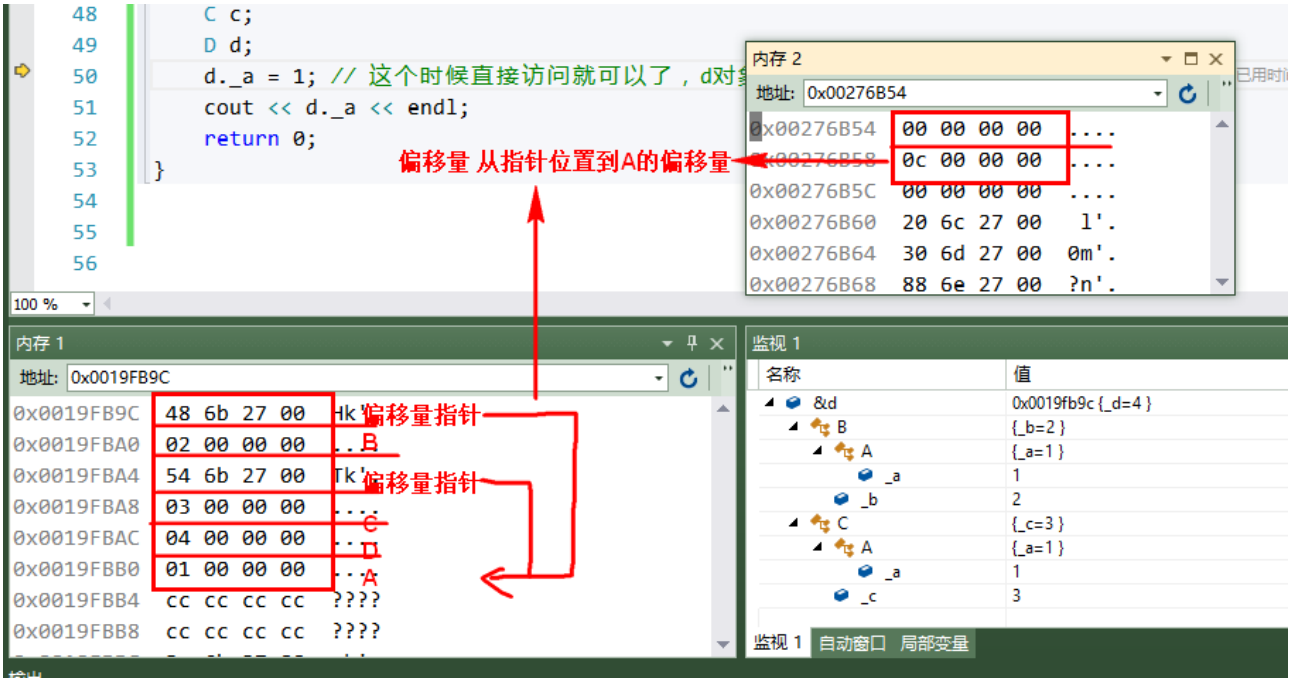
B b


50 d._a = 1; // 这个时候直接
51 cout << d._a << endl;
52 return 0;
53 }
54
55
56
100 %
内存 2
地址: 0x00276B30
0x00276B30 00 00 00 00 ....
0x00276B34 08 00 00 00 ....
0x00276B38 00 00 00 00 ....
0x00276B3C 00 00 00 00 ....
0x00276B40 08 00 00 00 ....
0x00276B44 00 00 00 00 ....
偏移量 D8 - D0 = 8
内存 1
地址: 0x0019FBD0
0x0019FBD0 30 6b 27 00 0k'.
0x0019FBD4 02 00 00 00 ....
0x0019FBD8 01 00 00 00 ....
0x0019FBDC cc cc cc cc ????
0x0019FBE0 cc cc cc cc ????
0x0019FBE4 01 00 00 00 ....
0x0019FBE8 cc cc cc cc ????
偏移表 指针
B
A
监视 1
名称
值
&d 0x0019fb9c {_d=4 }
&a 0x0019fbe4 {_a=1 }
&b 0x0019fbd0 {_b=2 }
&c 0x0019fbbc {_c=3 }
监视 1 自动窗口 局部变量

C c


50 d._a = 1; // 这个时候直接
51 cout << d._a << endl;
52 return 0;
53 }
54
55
56
100 %
内存 2
地址: 0x00276B3C
0x00276B3C 00 00 00 00 ....
0x00276B40 08 00 00 00 ....
0x00276B44 00 00 00 00 ....
0x00276B48 00 00 00 00 ....
0x00276B4C 14 00 00 00 ....
0x00276B50 00 00 00 00 ....
偏移量 C4 - BC =
内存 1
地址: 0x0019FBBC
0x0019FBBC 3c 6b 27 00 <k'.
0x0019FBC0 03 00 00 00 ....
0x0019FBC4 01 00 00 00 ....
0x0019FBC8 cc cc cc cc ????
偏移表指针
C
A
监视 1
名称
值
&d 0x0019fb9c {_d=4 }
&a 0x0019fbe4 {_a=1 }
&b 0x0019fbd0 {_b=2 }
&c 0x0019fbbc {_c=3 }

D d



我们把虚继承中的这个有关偏移量的表叫做虚基表

注意：

至于为什么不是将偏移量存放在指定地址，

而是在指定地址的后4个字节，

是因为这4个字节是为多态预留的

（在多态中我们还会来分析这个模型）

一般不到万不得已不要使用菱形继承虚继承，

因为菱形继承虚继承也带来了性能上的损耗

（毕竟多开了地址来存放偏移量）。