示例代码：

```
// 3. 多重继承有虚函数.cpp : 定义控制台应用程序的入口点。
//
#include "stdafx.h"
#include <iostream>
using namespace std;

// 床类
class CBed {
public:
    CBed() {
        printf("CBed::CBed()\r\n");
        m_nBed = 1;
    }

    ~CBed() {
        printf("CBed::~CBed()\r\n");
        m_nBed = 0;
    }

    virtual void sleep() {
        printf("CBed::sleep()\r\n");
    }

    void sleep2() {
        printf("CBed::sleep2()\r\n");
    }

private:
    int m_nBed;
};

// 沙发类
class CSofa {
public:
    CSofa() {
        printf("CSofa::CSofa()\r\n");
        m_nSofa = 2;
    }
```

```
39
40      ~CSofa() {
41          printf("CSofa::~CSofa()\r\n");
42          m_nSofa = 0;
43      }
44
45      virtual void sit() {
46          printf("CSofa::sit()\r\n");
47      }
48
49      void sit2() {
50          printf("CSofa::sit2()\r\n");
51      }
52
53  private:
54      int m_nSofa;
55  };
56
57  // 沙发床多重继承，既继承了沙发的特点，又继承了床的特点。
58  class CSofaBed : public CSofa, public CBed {
59  public:
60      CSofaBed() {
61          printf("CSofaBed::CSofaBed()\r\n");
62          m_nSofaBed = 3;
63      }
64
65      virtual void sleep() {
66          printf("CSofaBed::sleep()\r\n");
67      }
68
69      virtual void sit() {
70          printf("CSofaBed::sit()\r\n");
71      }
72
73      void test() {
74          printf("CSofaBed::test()\r\n");
75      }
76
77  private:
78      int m_nSofaBed;
79  };
80
```

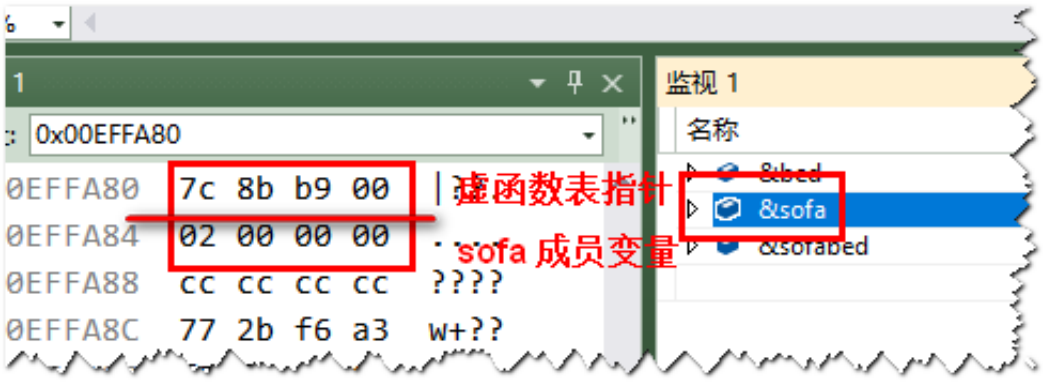
```
81 int main()
82 {
83     CSofa sofa;
84     CBed bed;
85     CSofaBed sofabed;
86     //对象大小
87     cout << sizeof(CSofa) << endl; // 8
88     cout << sizeof(CBed) << endl; // 8
89     cout << sizeof(sofabed) << endl; // 20
90 }
```

对象大小

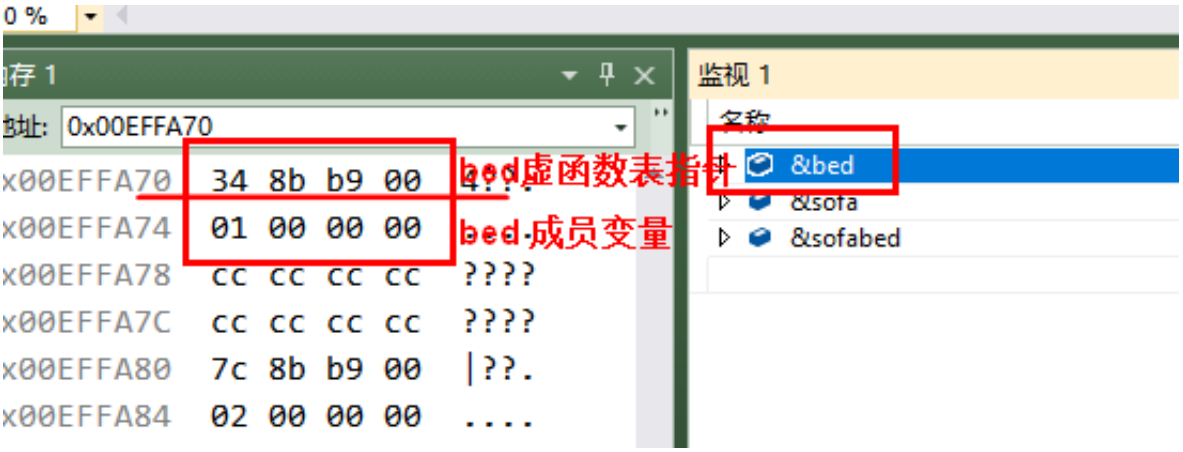
```
1     //对象大小
2     cout << sizeof(CSofa) << endl; // 8
3     cout << sizeof(CBed) << endl; // 8
4     cout << sizeof(sofabed) << endl; // 20
```

对象内存分析

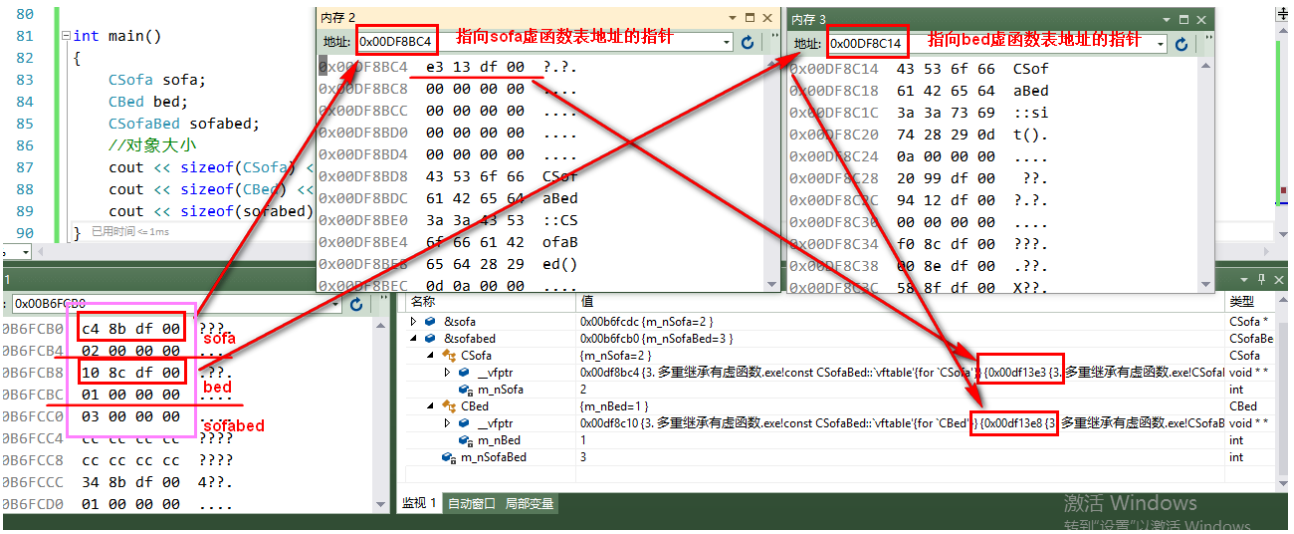
`CSofa sofa;`



`CBed bed;`




```
CSofaBed sofabed;
```



函数调用

参考虚函数调用部分。