派生类

调用顺序

```
// 3 A B C extends.cpp : 定义控制台应用程序的入口点。
//

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

class CTop
{
private:
    int m_a;

public:
    CTop(int a);
    ~CTop();
};

class CMiddle:public CTop
{
private:
    int m_b;

public:
    CMiddle(int a, int b);
    ~CMiddle();
};

class CBottom:public CMiddle
{
private:
    int m_c;

public:
    CBottom(int a, int b, int c);
    ~CBottom();
```

```
};

CTop::CTop(int a = 0)
{
    this->m_a = a;
    cout << "a.." << m_a << endl;
    cout << "CTop constructed\r\n";
}

CTop::~CTop()
{
    this->m_a = 0;
    cout << "a.." << m_a << endl;
    cout << "~CTop deconstructed\r\n";
}

CMiddle::CMiddle(int a = 0, int b = 0) :CTop(a) //初始化成员列表，执行CTop类中的构造函数
{
    this->m_b = b;
    cout << "b.." << m_b << endl;
    cout << "CMiddle constructed\r\n";
}

CMiddle::~CMiddle()
{
    this->m_b = 0;
    cout << "b.." << m_b << endl;
    cout << "~CMiddle constructed\r\n";
}

CBottom::CBottom(int a, int b, int c):CMiddle(a,b)//初始化成员列表，执行CMiddle类中的构造函数
{
    this->m_c = c;
    cout << "c.." << m_c << endl;
    cout << "CBottom constructed\r\n";
}

CBottom::~CBottom()
{
    this->m_c = 0;
```

```
    cout << "c.." << m_c << endl;
    cout << "~CBottom constructed\r\n";
}


int main()
{
    CBottom bottom(1, 2, 3);

    return 0;
}
/*
output:

a..1
CTop constructed
b..2
CMiddle constructed
c..3
CBottom constructed
c..0
~CBottom constructed
b..0
~CMiddle constructed
a..0
~CTop deconstructed
*/

/*
创建类对象时，应先调用其构造函数。
但是如果这个类有成员对象，则要先执行成员对象自己所属类的构造函数，
当全部成员对象都执行了自身类的构造函数后，再执行当前类的构造函数。

调用顺序，
构造函数
CTop -> CMiddle -> CBottom

析构函数
CBottom -> CMiddle -> CTop
*/
```

参考资料

- 派生类及调用顺序
  https://blog.csdn.net/rhzwan123/article/details/2105205