//继承同名成员的处理方式

//访问子类同名成员，直接访问

//访问父类同名成员，想要加作用域

#include<iostream>

using namespace std;

class base {

public:

base() {

ma = 100;

}

void func() {

cout << "base-func()调用" << endl;

}

void func(int a) {

cout << "base-func(int a)调用" << endl;

}

int ma;

};

class son :public base {

public:

son() {

ma = 200;

}

void func()

{

cout << "son-func的调用" << endl;

}int ma;

};

//同名成员属性

void test01() {

son s;

cout << "son ma=" << s.ma << endl;

cout << "base ma=" << s.base::ma << endl;

//作用域 base::

}

//同名成员函数

void test02(){

son s;

s.func();//直接调用为子类

s.base::func();//作用域

//想要访问func（int a）

//s.func(100);错，因为如果子类，父类出现同名函数，子类的同名函数会隐藏掉父类中所有同名函数，包括重载

//要加作用域

s.base::func(100);

}

int main() {

//test01();

test02();

system("pause");

return 0;

}