//继承静态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

//继承中同名静态成员处理方式

#include<iostream>

using namespace std;

class base {

public:

static int ma;

static void func(){

cout << "base-static void func()" << endl;

}

static void func(int a) {

cout << "son-static void func()" << endl;

}

};

int base::ma = 100;//静态成员类内定义，类外初始化

class son :public base {

public :

static int ma;

static void func() {

cout << "son-static void func()" << endl;

}

};

int son::ma = 200;

//同名静态成员属性

void test01() {

//1.通过对象访问

son s;

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

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

//2.通过类名访问

cout << "son ma" << son::ma << endl;

//第一个双冒号代表通过类名方式访问，第二个表示通过父类作用域

cout << "base ma" << son:: base::ma << endl;

}

//同名静态成员函数

void test02() {

//1.通过对象

son s;

s.func();

s.base::func();

//2.通过类名

son::func();

son::base::func();

//子类和父类出现同名静态成员函数，也要隐藏父类中所有同名成员函数

son::base::func(100);

}

int main() {

//text01();

test02();

system("pause");

return 0;

}