第五次实验报告

实验内容:

#include<iostream>

using namespace std;

class Base {

public:

void setx(int i)

{

x = i;

}

int getx()

{

return x;

}

public:

int x;

};

class Derived :public Base {

public:

void sety(int i)

{

y = i;

}

int gety()

{

return y;

}

void show()

{

cout << "Base::x=" << x << endl;

}// 语句1

public:

int y;

};

int main()

{

Derived bb; // 语句2

bb.setx(16); // 语句3

bb.sety(25); // 语句4

bb.show(); // 语句5

cout << "Base::x=" << bb.x << endl; // 语句6

cout << "Derived::y=" << bb.y << endl; // 语句7

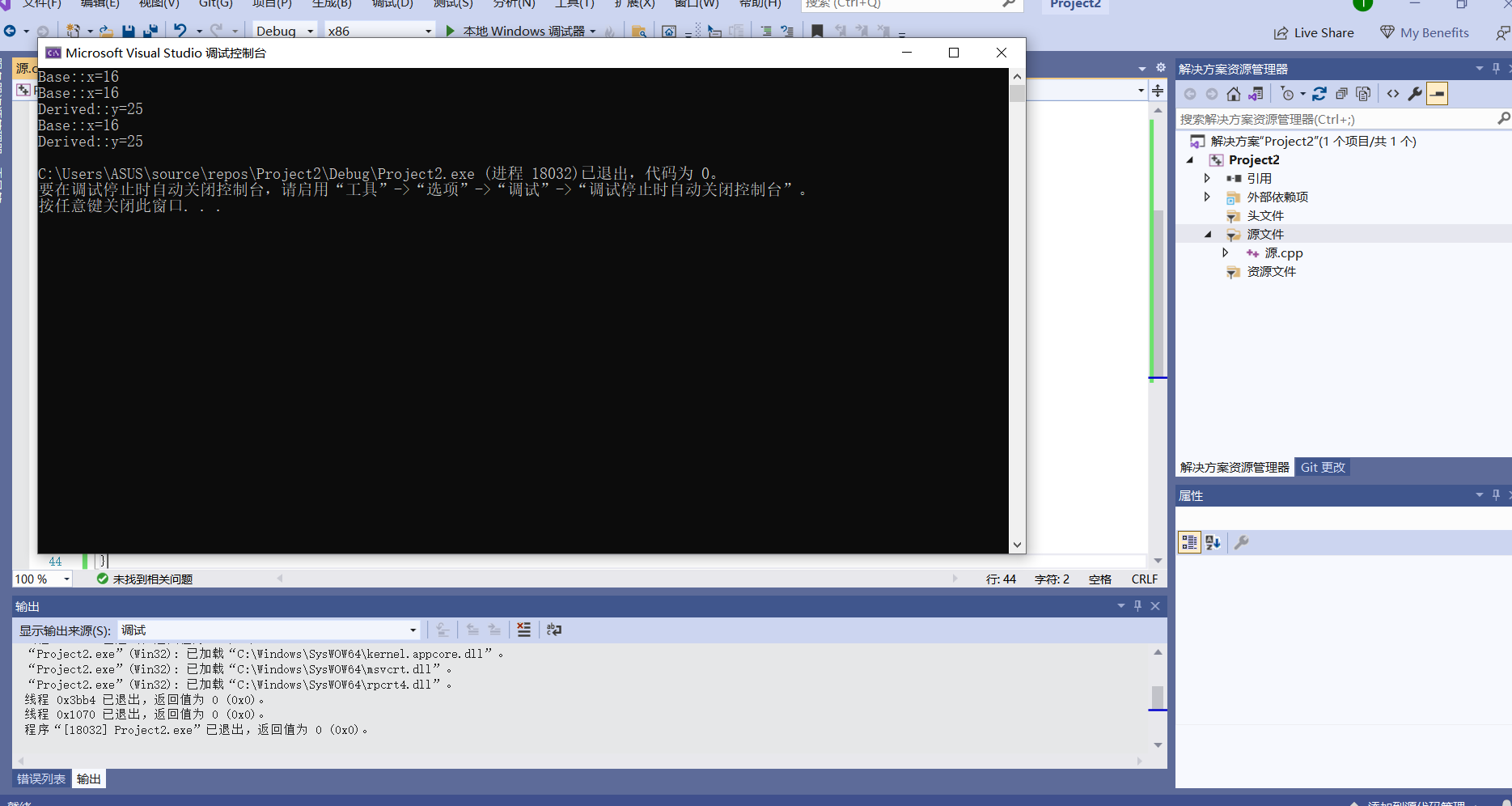
cout << "Base::x=" << bb.getx() << endl; // 语句8

cout << "Derived::y=" << bb.gety() << endl; // 语句9

return 0;

}

运行结果：



1如果把Base中的x改为private类型，在Derived继承Base时,Derived是无法操纵Base中的x参量的，所以show()函数是无法被调用的。

2如果把Base中的x改为protected类型，在Derived继承Base时,x数据被保护了，Derived是无法操纵Base中的x参量的，所以show()函数是无法被调用的。

3如果把继承类型改为private类型，在Derived继承Base时,其中的bb.setx()函数是无法进行类外访问的。

4如果把继承类型改为protected类型，在Derived继承Base时,其中的bb.setx()函数是无法进行类外访问的。

体会：

本次实验使我进一步了解三种继承的使用方法，以及如何调用派生类和基类的不同成员的权限，让我对派生类与继承的理解进一步的加深。在今后的操作中要对知识进行更好的掌握，帮助自己更好的进行操作。

Copyright ©2021-2099 YimingYang. All rights reserved