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

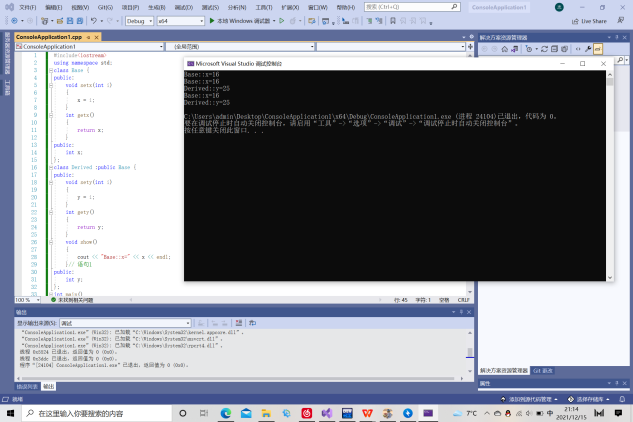
cout << "Base::x=" << bb.getx() << endl; // 语句8

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

return 0;

}

运行结果：



感想

派生可以在既有类的基础上,通过增加少量代码或修改少量代码的方法得到新的类, 从而较好地解决了代码重用的问题。  
派生有两种派生方式：公有派生和私有派生，无论哪种派生方式, 基类中的私有成员既不允许外部函数访问, 也不允许派生类中的成员函数访问,但是可以通过基类提供的公有成员函数访问。  
公有派生与私有派生的不同点在于基类中的公有成员在派生类中的访问属性。