# 实验五 派生类与继承

程序代码：

#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;

}

public:

int y;

};

int main()

{

Derived bb;

bb.setx(16);

bb.sety(25);

bb.show();

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

cout << "Derived::y=" << bb.y << endl;

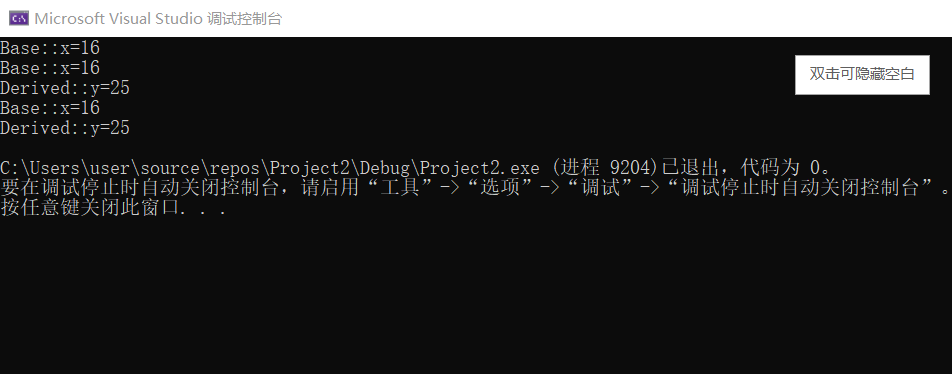
cout << "Base::x=" << bb.getx() << endl;

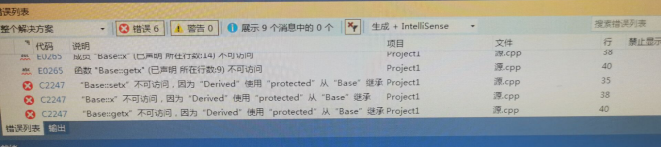
cout << "Derived::y=" << bb.gety() << endl;

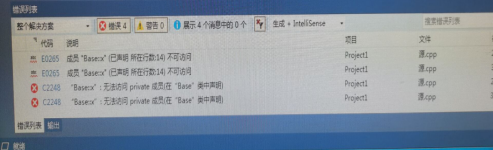
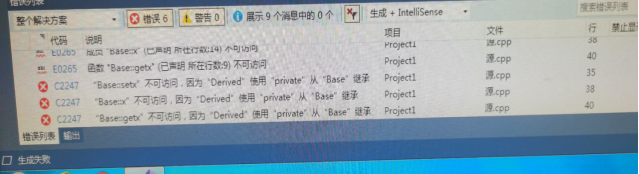
return 0;

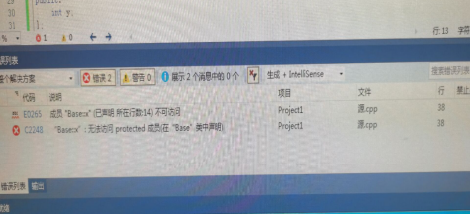
}

运行结果：









心得体会：

派生类对基类的访问方式有两种：内部访问和对象访问。在编写程序时要注意成员函数的类型和继承方法，其对程序正常运行有着很大的影响。

Copyright ©2021-2099 Tianhang Li. All rights reserved