**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(22);

bb.sety(33);

bb.show();

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

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

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

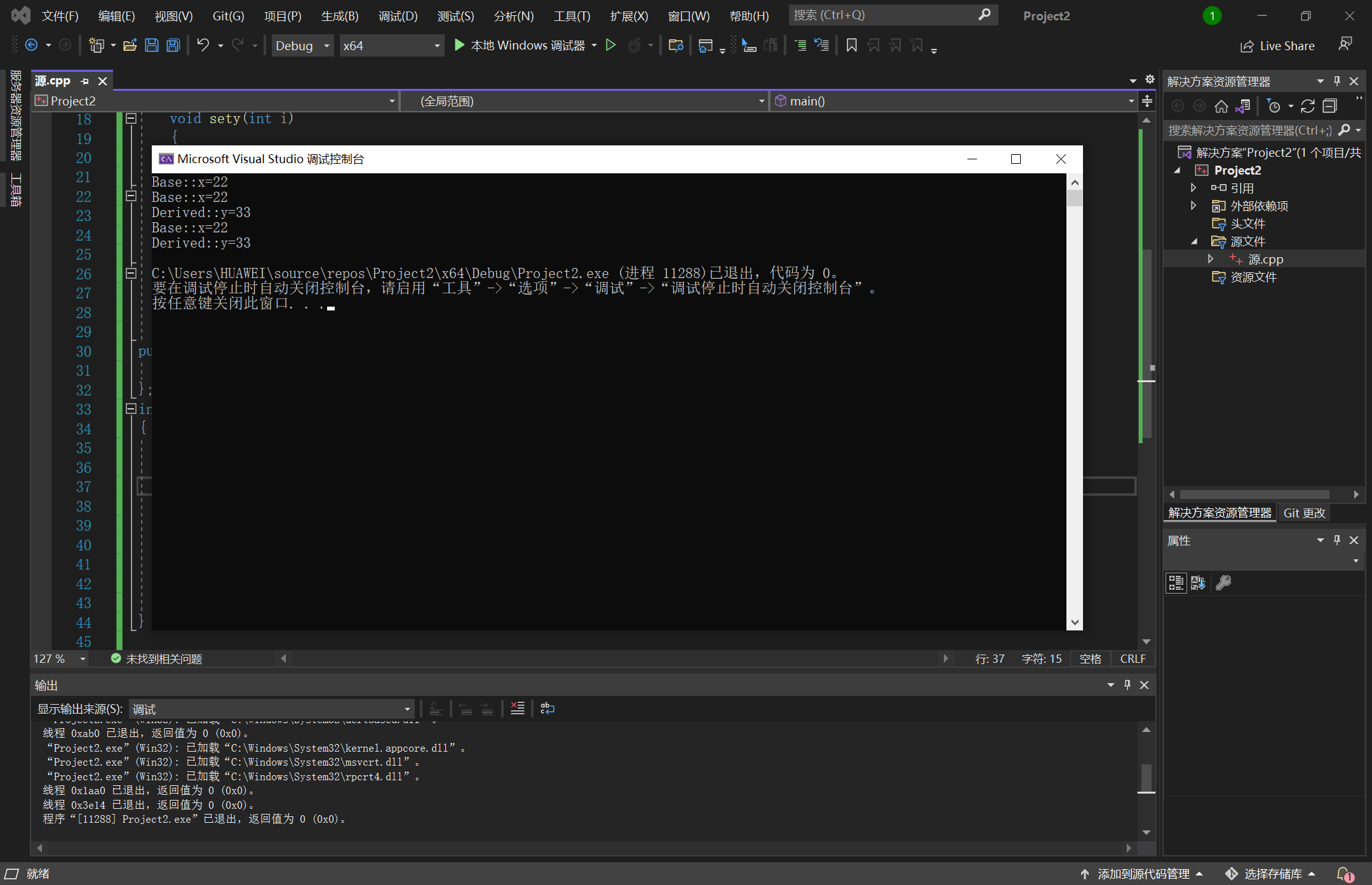
cout << "Derived::y=" << bb.gety() << endl;

return 0;

}

**程序结果：**

更改前：



将x改为私有后：

语句1，6报错。原因为基类的私有成员x不可被派生类Derived和派生类对象bb访问，仅能被基类自己的成员函数getx访问，因此语句8没有错误。

将x改为保护后：

语句6报错。原因为基类的保护成员x不可被派生类对象bb访问，仅能被派生类Derived和基类自己的成员函数getx访问，因此语句1和8没有错误。

将派生类改为私有继承后：

语句3,6,8报错。基类中的所有公有成员均变为私有访问属性，即仅可内部访问（派生类对其访问），而对象bb则没有办法法访问。

将派生类改为保护继承后：

语句3,6,8报错。基类中的所有公有成员均变为保护访问属性，即仅可内部访问（派生类对其访问），但是对象bb则没有办法访问。

**感想心得：**

这次上机后我明白了一个类可以派生自多个类，这意味着，它可以从多个基类继承数据和函数。定义一个派生类，我们使用一个类派生列表来指定基类。我对于继承方式和类中成员的声明属性有了更加深刻清晰的认识，在今后的使用当中，我要多多看看课本，得到自己的理解，在以后的编程中可以写出属于自己想法的程序。