**实验目的：**

* **掌握派生类的声明方法和派生类构造函数的定义方法**
* **掌握不同方式下，基类成员在派生类中的访问属性**

**实验内容：**

* **输入下列程序**

#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;

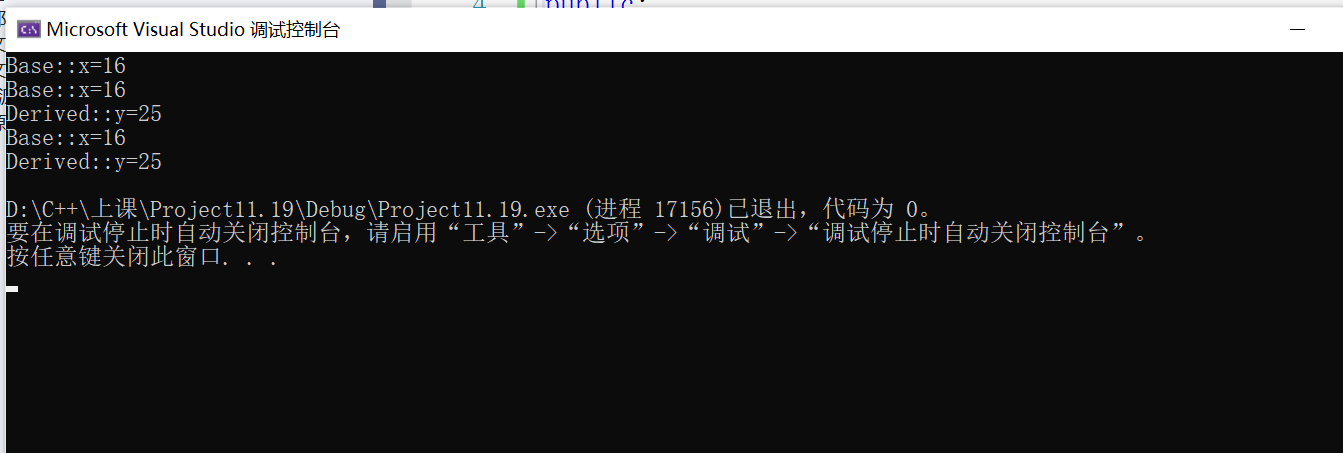
cout << "Base::x=" << bb.getx() << endl;

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

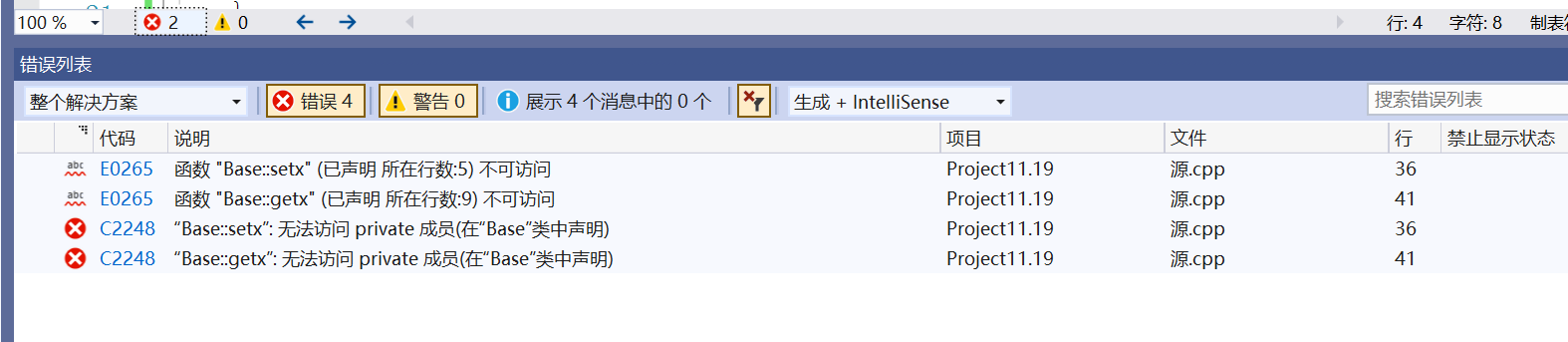
return 0;

}

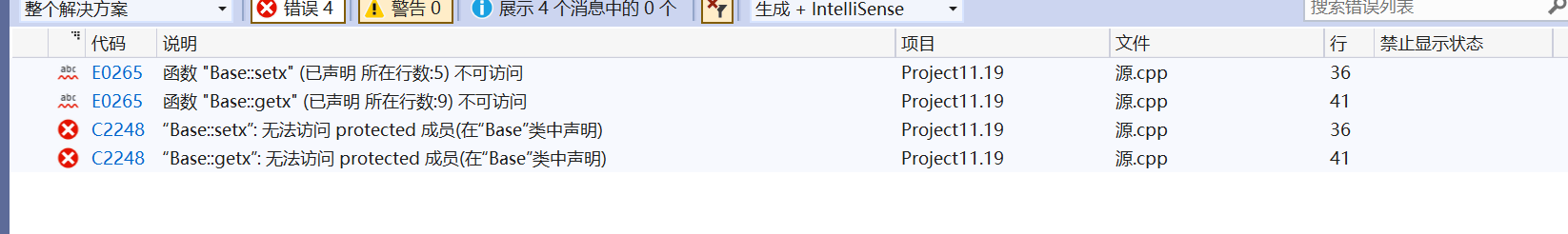
* 写出程序运行的结果。



* 按照以下要求，对程序进行修改后再调试，指出调试中出错的原因。
* 将基类Base中数据成员x的访问权限改为private时，会出现哪些错误，为什么？

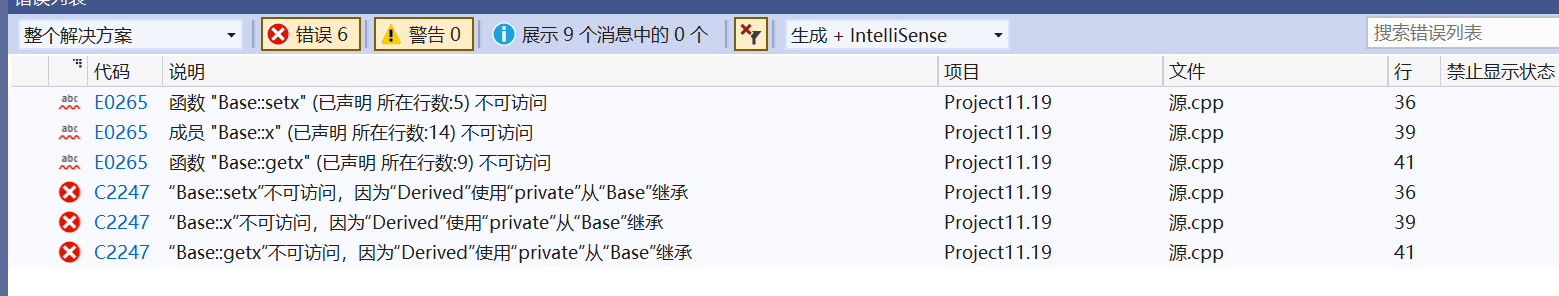
会导致派生类和对象均无法访问基类成员x，进而出现错误。

* 将基类Base中数据成员x的访问权限改为protected时，会出现哪些错误，为什么？



会导致对象无法访问基类成员x。

* 在原程序的基础上，将派生类Derived的继承方式改为private时，会出现哪些错误，为什么？



会导致派生类和对象均无法访问基类的成员函数和成员变量。

* 在原程序的基础上，将派生类Derived的继承方式改为protected时，会出现哪些错误，为什么？



会导致对象无法访问基类的成员函数及变量。

Copyright ©2021-2099 Chen Yilu. All rights reserved