第五次上机实验报告

**学号：**202030310186  **姓名：申一帆**  **班级：**自动2002

1. **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

1. **实验程序结果**

#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

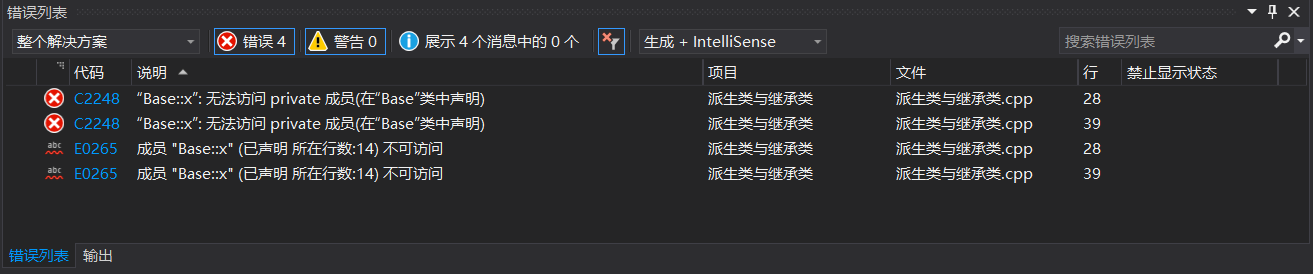
return 0;

}



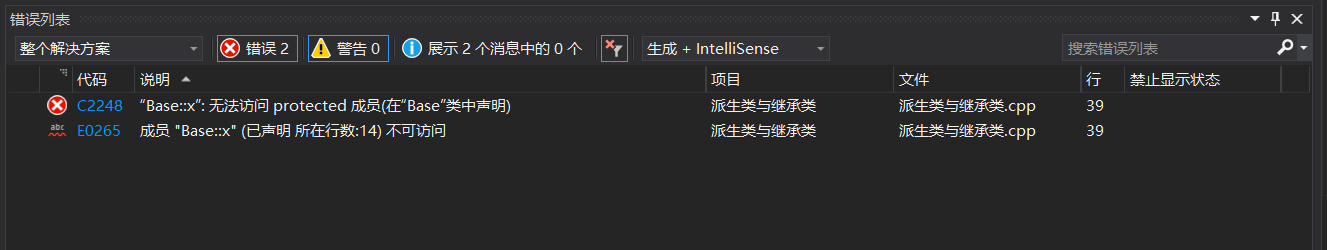
1. **问题分析**

1、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？



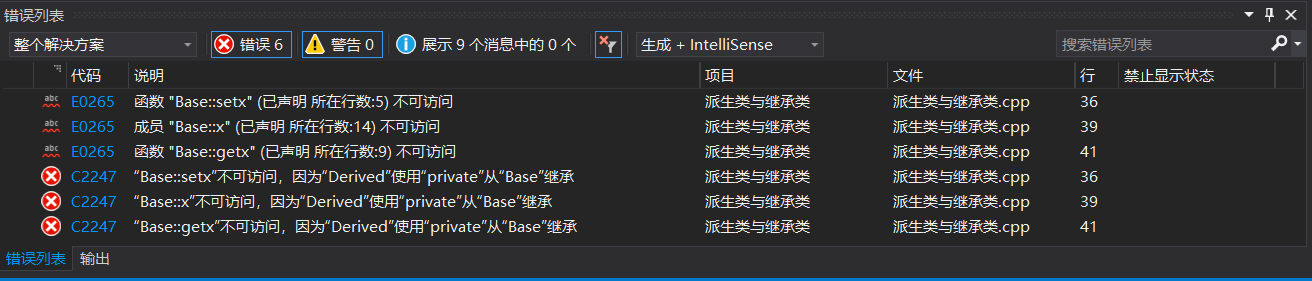
会导致派生类和对象均无法访问基类成员x，进而出现错误。

2、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。

3、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。

1. 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。

1. **实验心得**

对于私有成员变量而言，其初心就是为了保护封装起来不让别人访问，所以不论是内部还是对象访问都不能访问私有型变量，体现了封装性。

Copyright ©2021-2099 YiFanShen. All rights reserved