## 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;

}// 语句1

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;

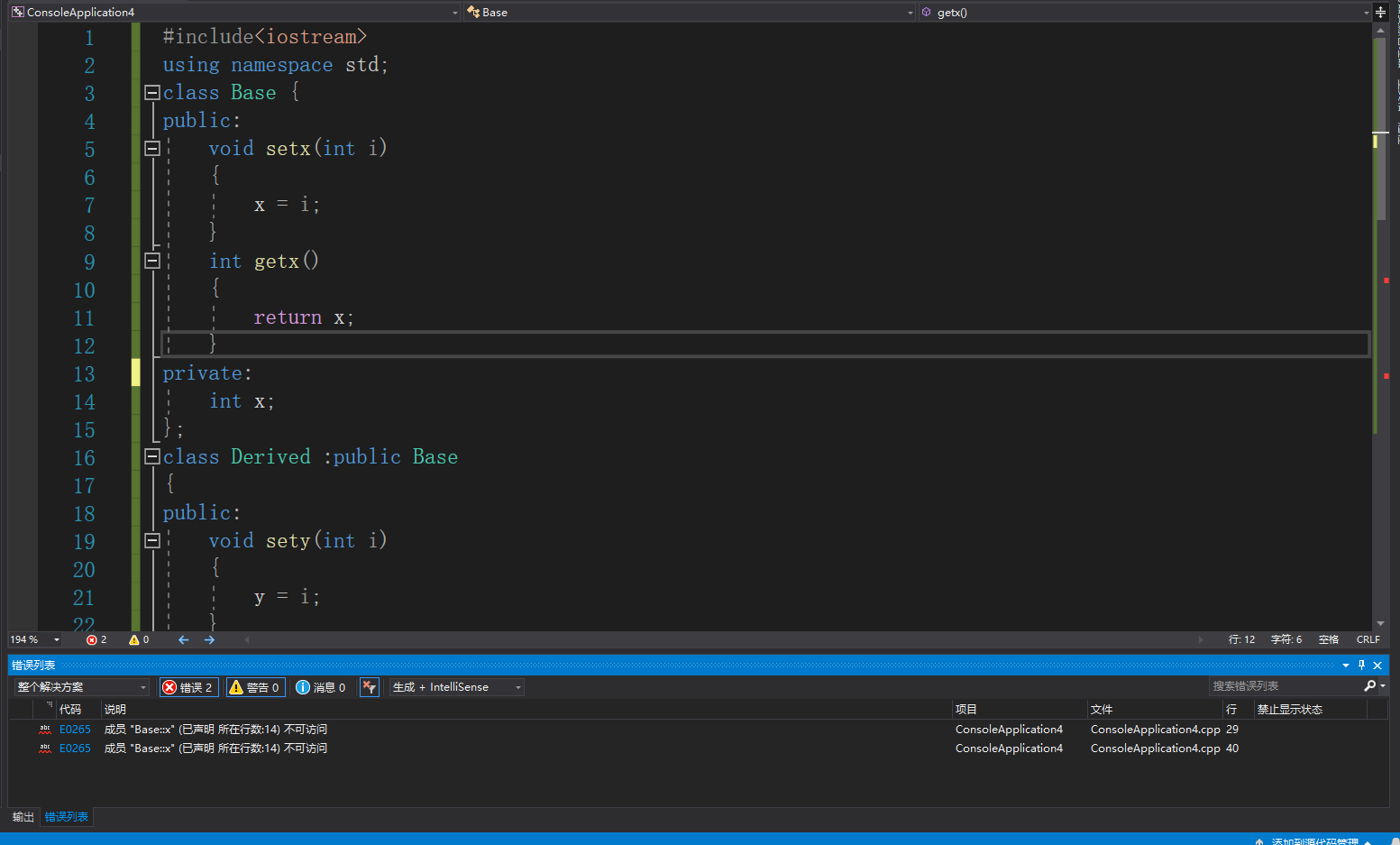
return 0;

}

## 程序结果

1、

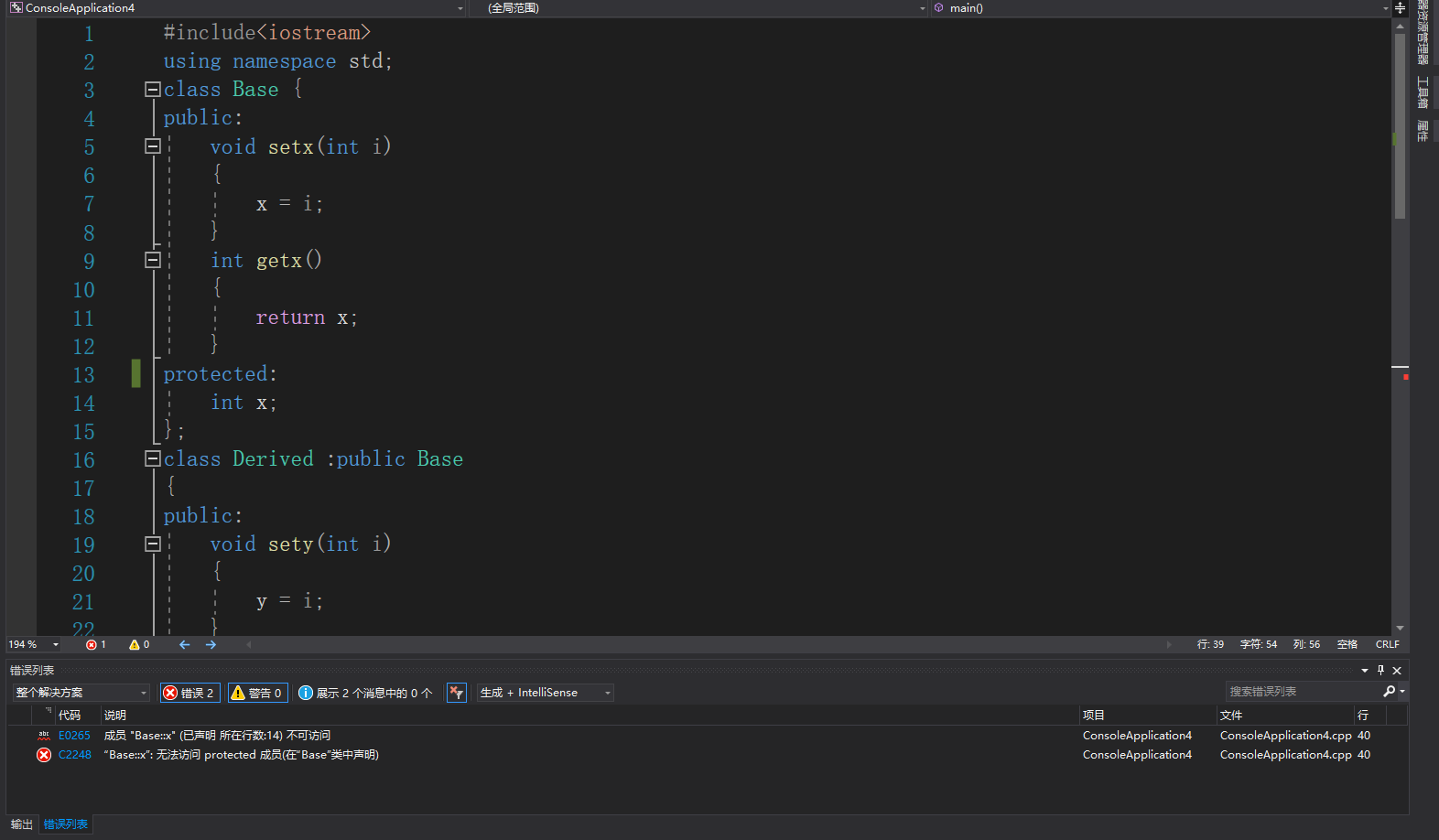
2、



**出现的问题：**x在基类Base外（无论是派生类中新增的成员函数还是派生类的对象中）无法访问。

**原因：**基类的私有对象不允许派生类继承，因此在私有派生类中，无论是派生类成员函数还是通过派生类的对象，都无法直接访问自基类继承而来的私有成员。

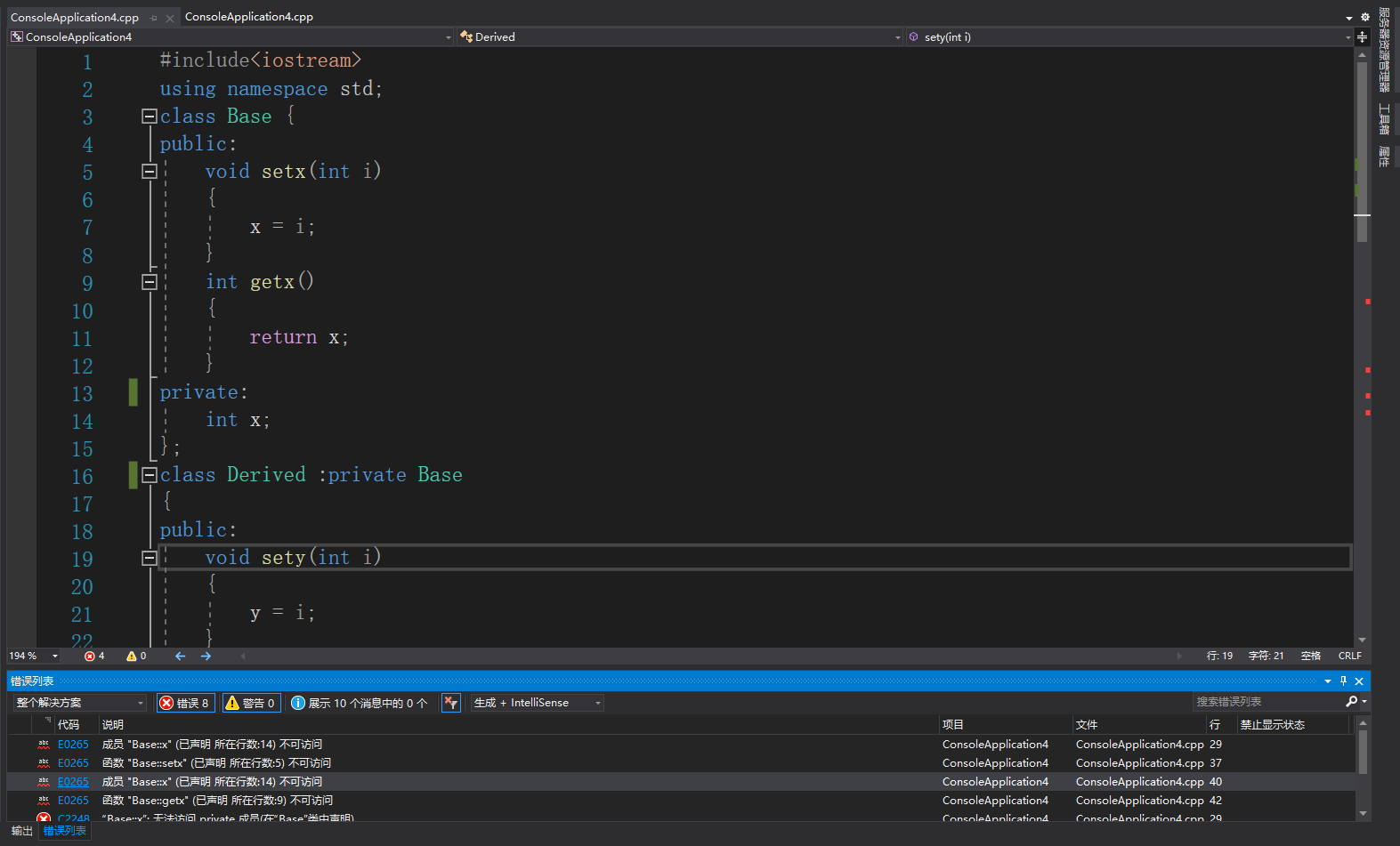
3、



**出现的问题：**声明为Protected的变量x无法被派生类的对象访问。

**原因：**当类的继承方式为公有继承时，基类中的所有保护成员在派生类中仍以保护成员的身份出现。此时派生类可以内部访问保护成员，但派生类对象不可访问保护成员。

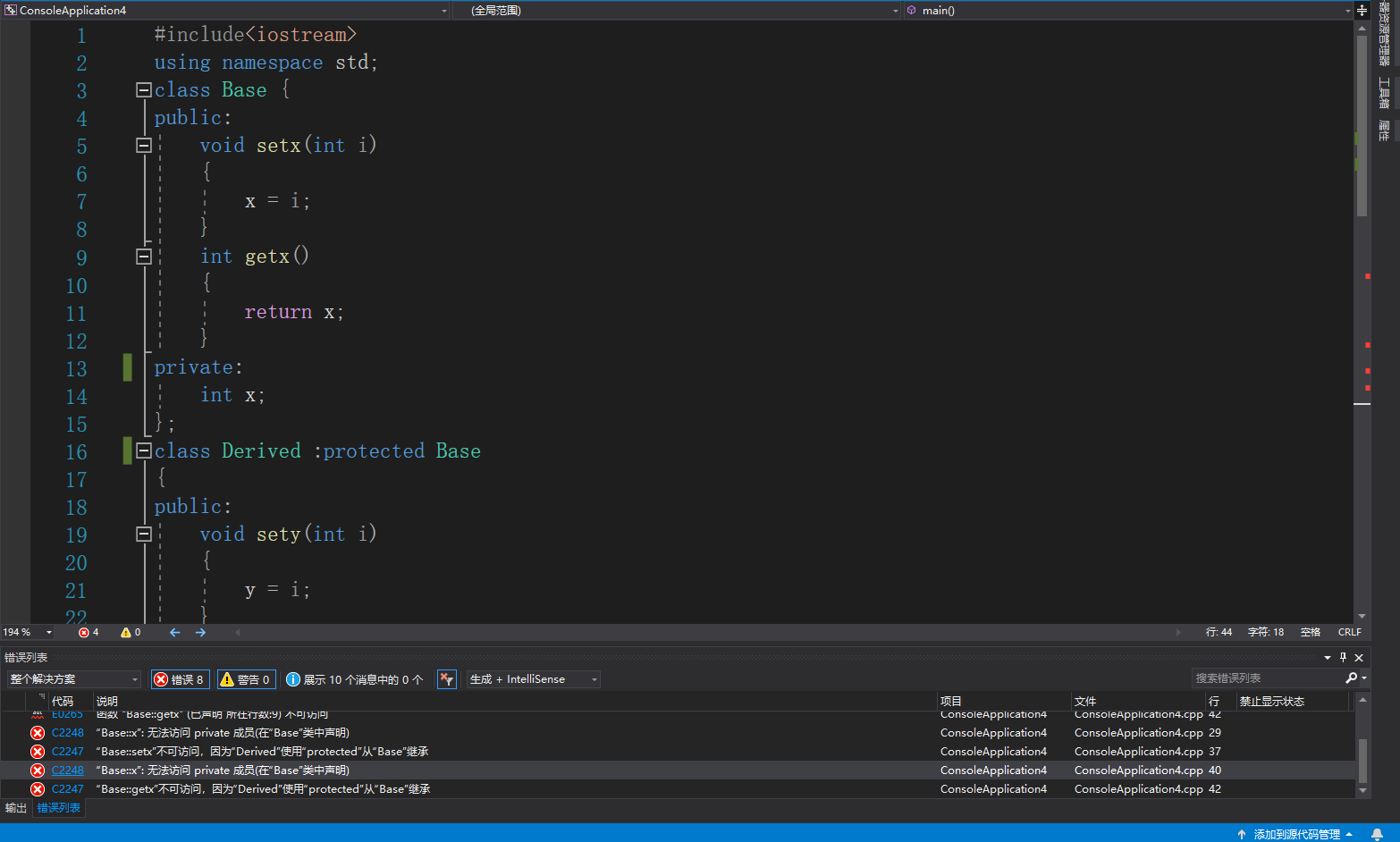
4、



**出现的问题：**Derived类的对象x无法访问基类Base中的函数与成员。

**原因：**当类的继承方式为私有继承时，基类的公有成员和保护成员被继承后作为派生类的私有成员，派生类的成员函数可以直接访问他们，但是在类外部通过派生类的对象是无法访问的。

5、



**出现的问题：**Derived类的对象无法访问继承自Base类的成员。

**原因：**类的继承方式为保护继承时，基类的公有成员和保护成员都被继承到派生类中都作为派生类的保护成员，派生类的其他成员可以直接访问他们，但是在类的外部，不能通过派生类的对象来访问他们。

## 程序心得

1、类的继承方式有很多种，每一种都要熟练掌握，并编写过程中做到灵活使用，才能真正实现代码的复用，提高编程效率。

2、派生类的构造函数需要以合适的初值作为参数，其中一些传递给基类的构造函数，用于初始化相应成员，另一些参数要用于对派生类新增的成员对象进行初始化。

Copyright ©2021-2099 BinLu. All rights reserved