#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;

}

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？

原因：派生类中的成员函数不能直接访问基类中的私有成员，基类中的私有成员不能被外部直接访问

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？

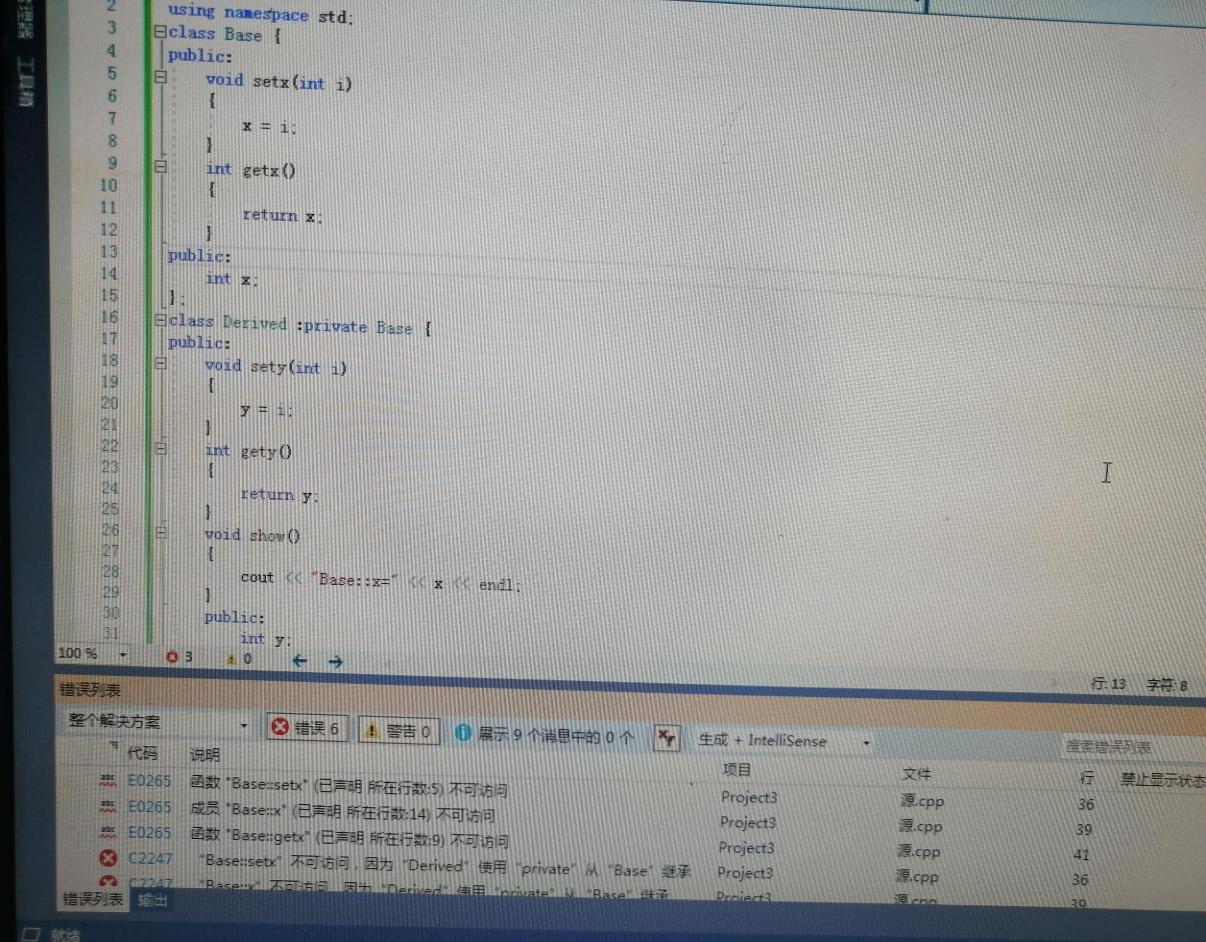
原因：基类中的保护成员不能被外部直接访问

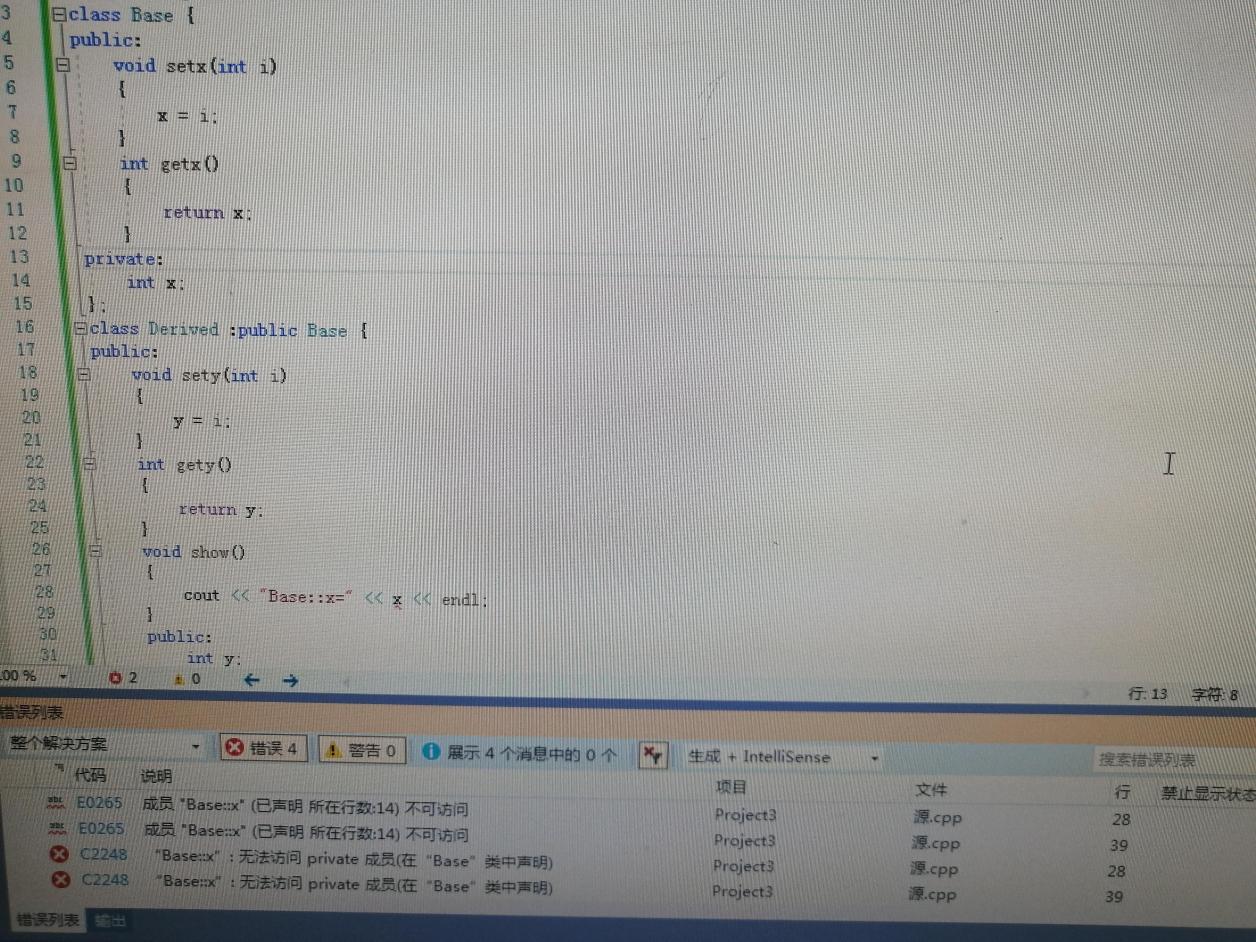
在原程序的基础上，将派生类Derived的继承方式改为private时，会出现哪些错误，为什么？

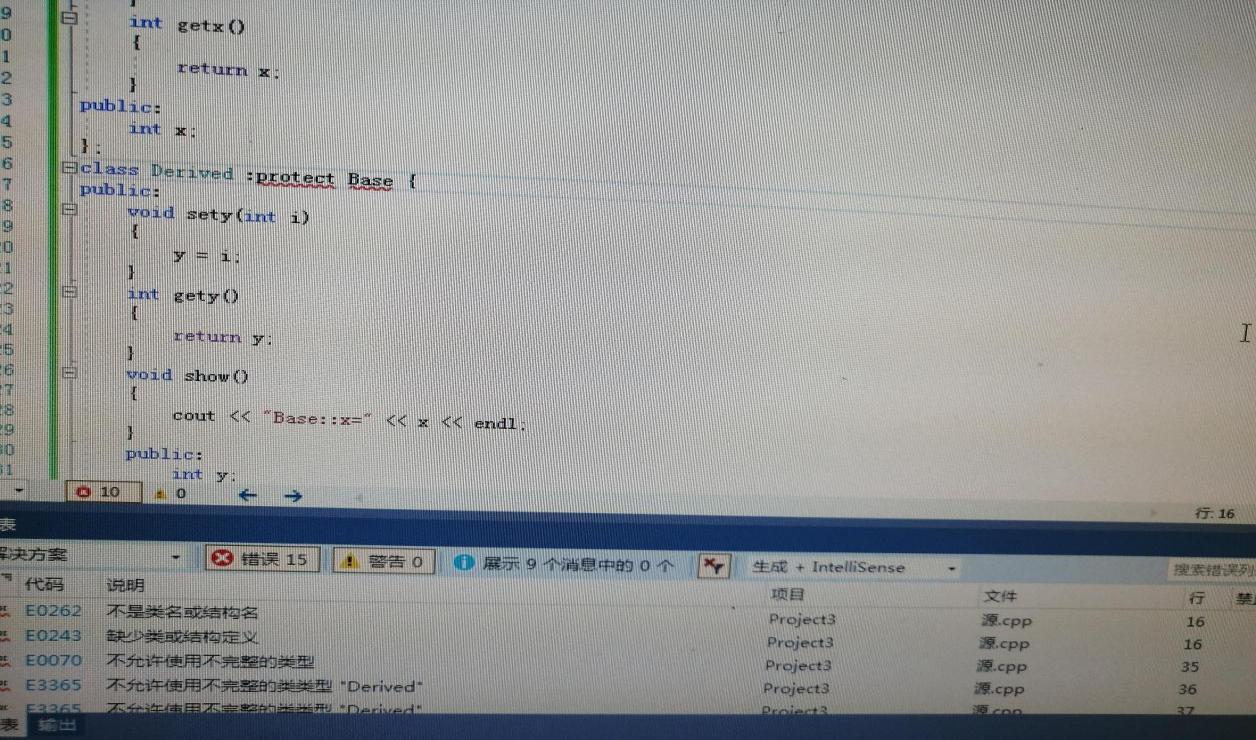
原因：基类成员setx函数在派生类中为私有成员，不能被直接访问；基类中的公有成员x在派生类中为私有成员，不能被直接访问；基类的成员getx函数在派生类中为私有成员，不能被直接访问

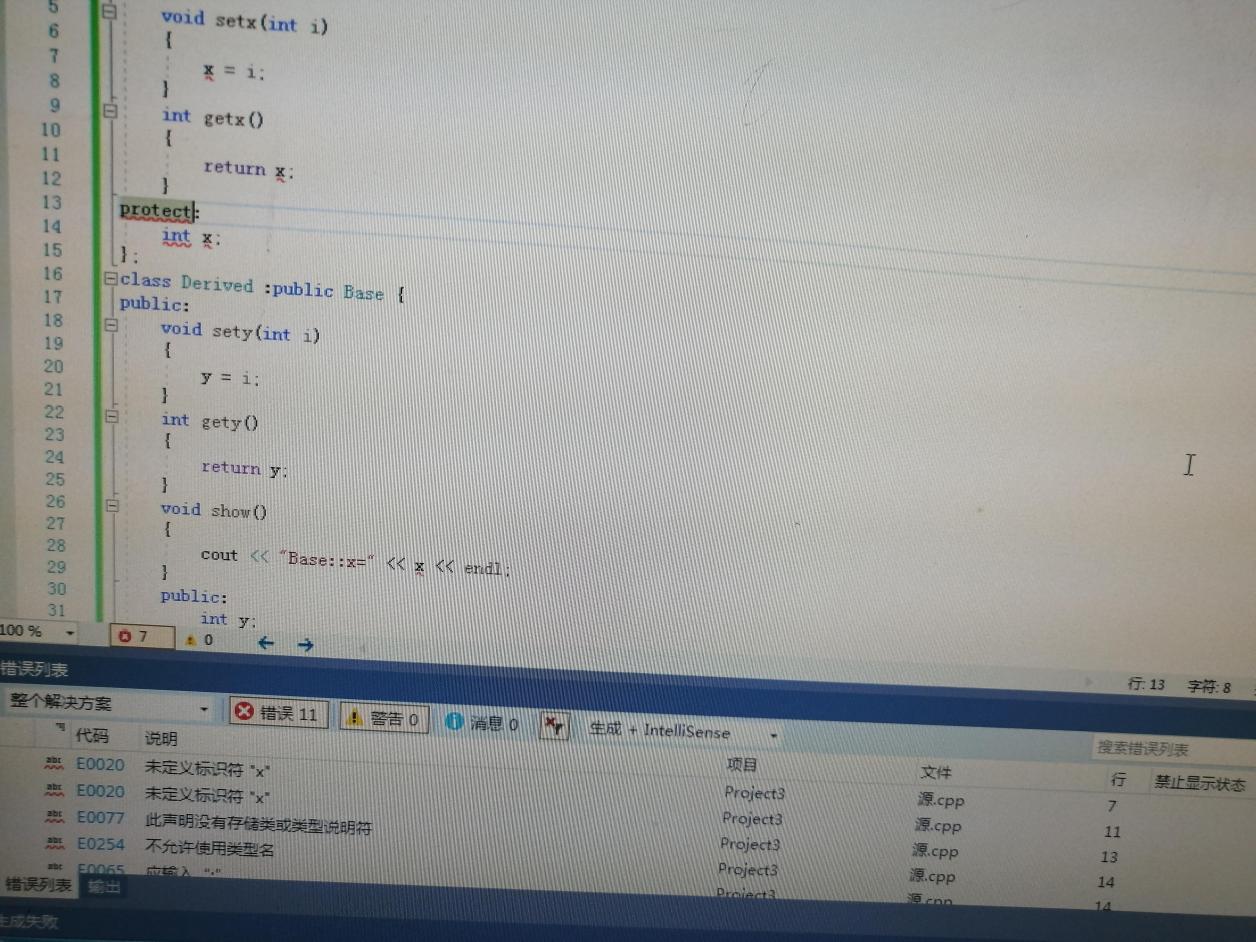
在原程序的基础上，将派生类Derived的继承方式改为protected时，会出现哪些错误，为什么？

原因：基类的setx成员函数在派生类中为保护成员，不能被直接访问；基类中的公有成员x在派生类中为保护成员，不能被直接访问；基类的getx成员函数在派生类中为保护成员，不能被直接访问。









上机心得：编写程序容易将英文写成中文，有点麻烦，编写程序不够熟练，通过这次程序能发现程序的错误所在。

Copyright IMG_2562021-2099 wangzezheng. All rights reserved