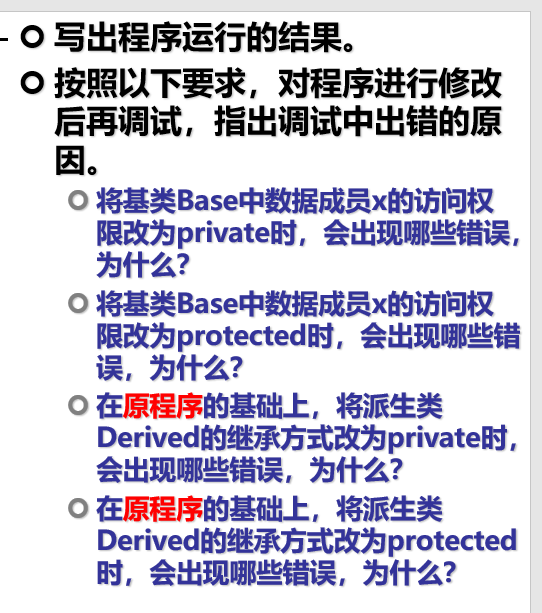
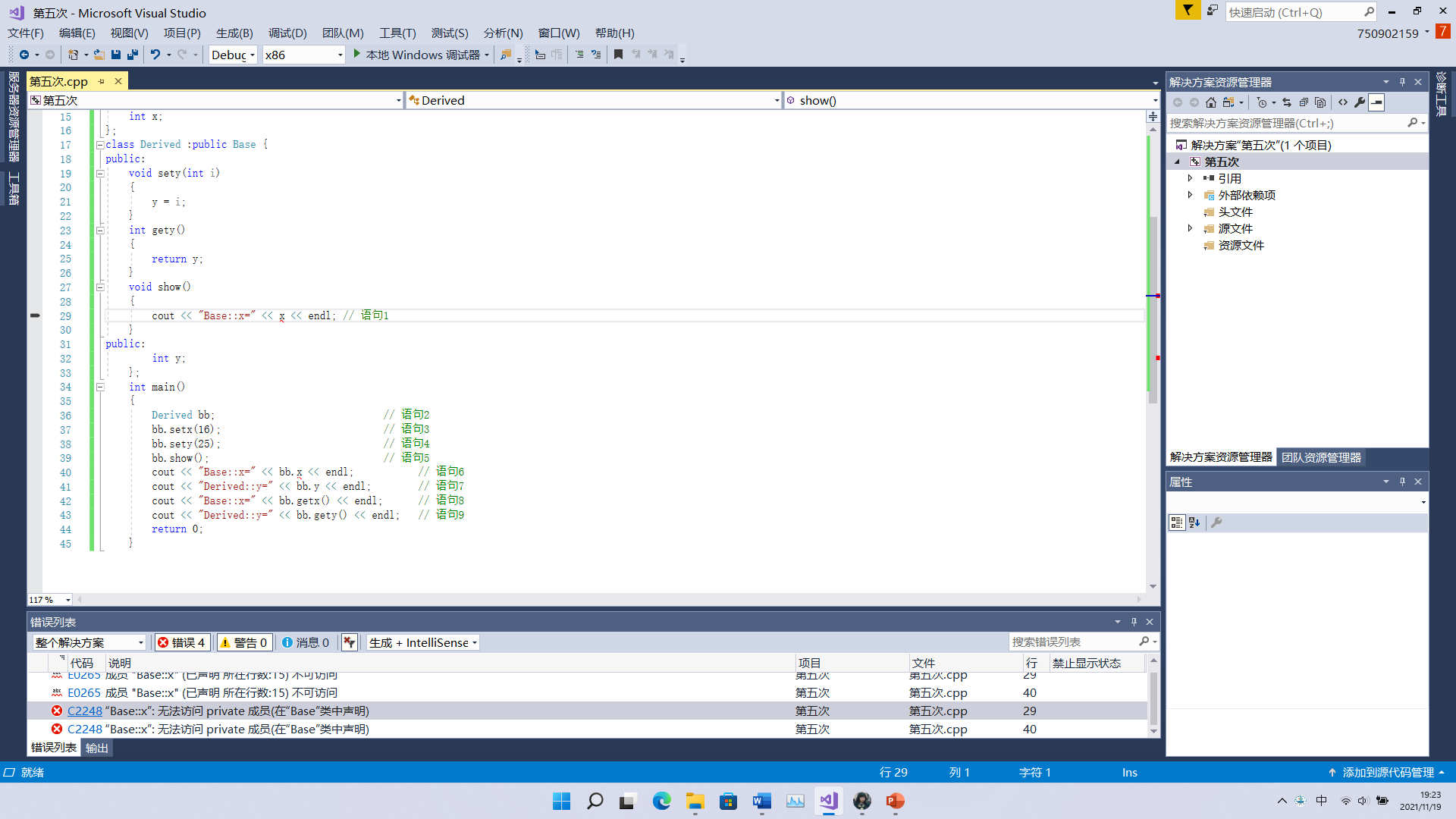
**第五次上机实验报告**

实验要求

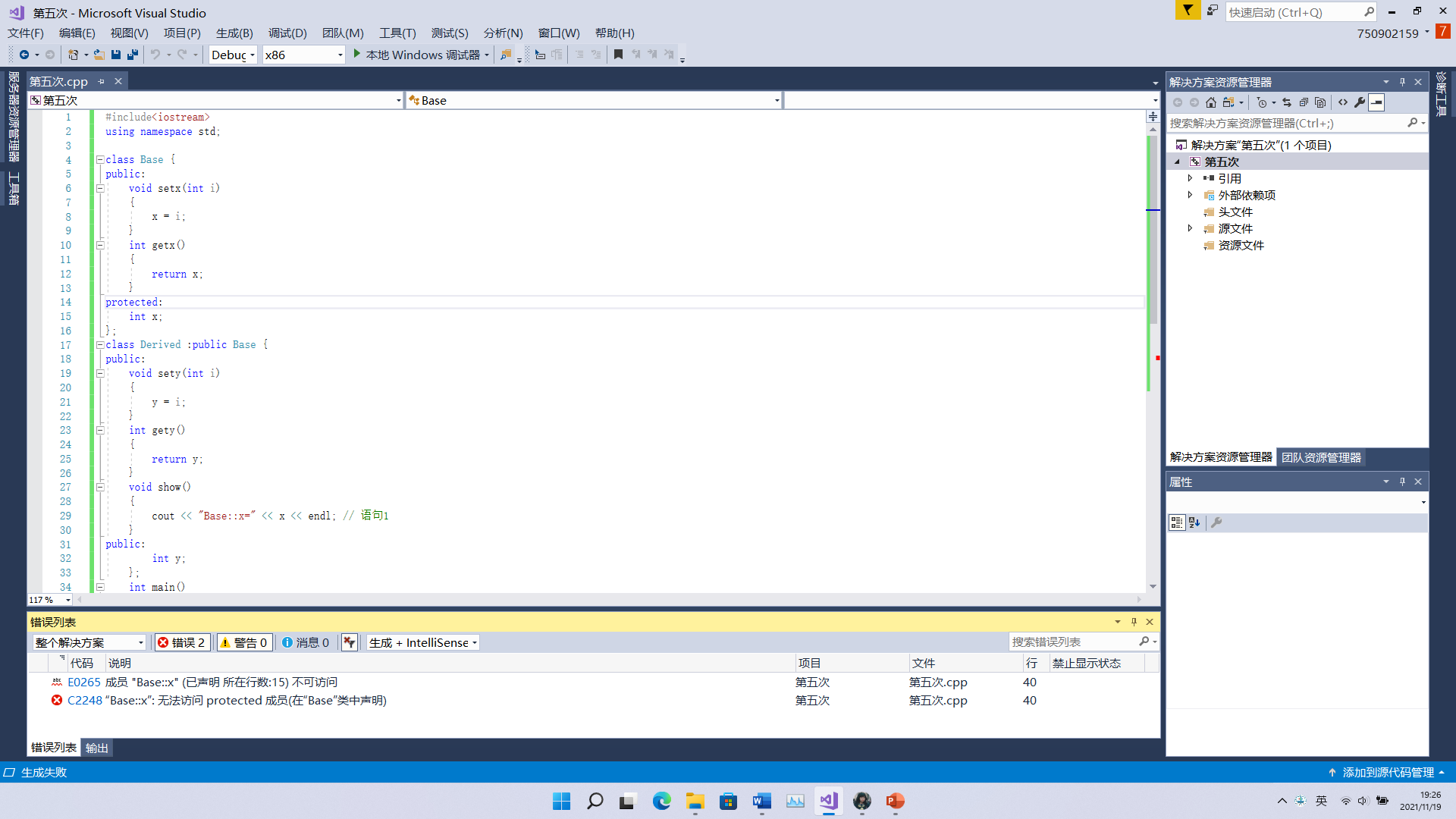


心得体会

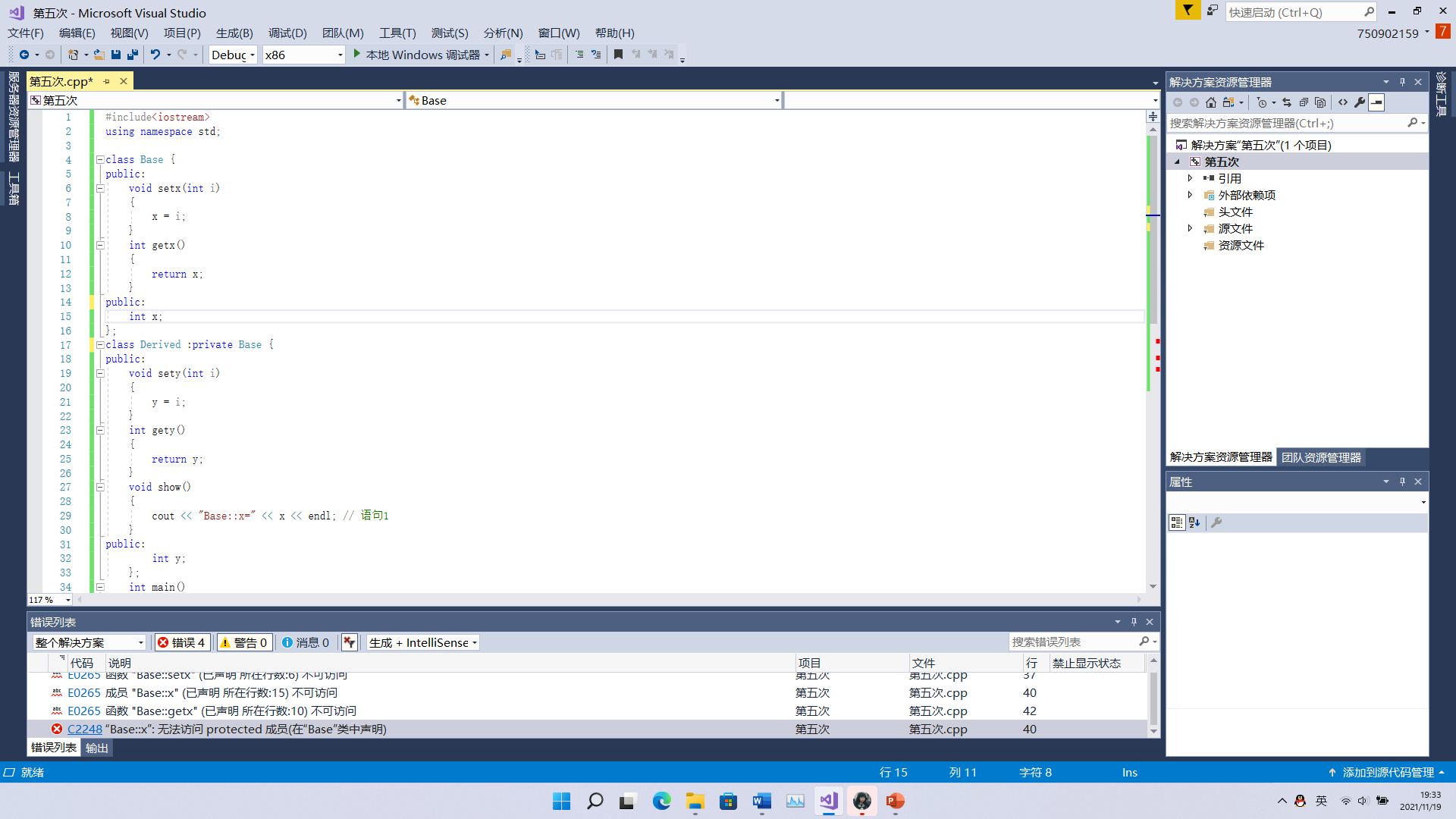
原代码运行结果

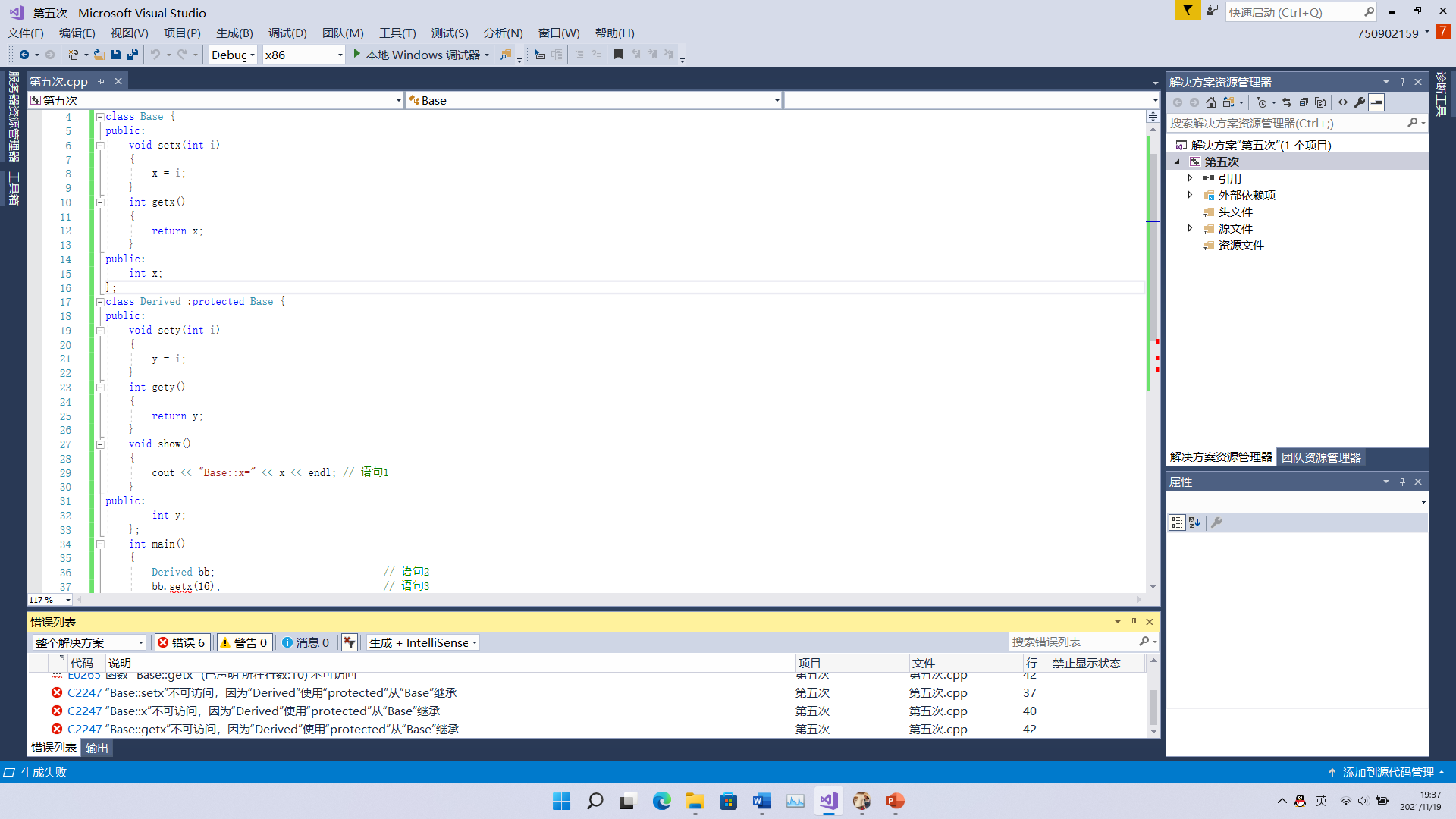


将x变量改为私有后，私有属性共有继承后仍为私有，故基类中的私有成员通过对象不可访问。



将x变量改为保护成员后，保护成员公有继承后仍为保护成员，通过对象不可访问，但派生类内部仍可访问。（内部可访问，外部不可访问）

私有继承基类后，基类中所有成员变量和成员函数变为私有，通过对象（外部访问不可访问）然sety等函数仍为共有



改为保护继承后，继承后的成员变为保护，创建派生类对象时，通过对象不可访问，但派生类内部仍可访问。（内部可访问，外部不可访问）

类的继承可以增加代码的执行的重复性，提高代码的利用率。类的继承应该秉持着“不忘初心的原则”来继承。成员的访问权限只能降低不能升高，public大于protected大于private。

实验代码与程序结果

|  |
| --- |
| //版权声明  //Copyright 2021-2099 ZhantengZhang.  //202030310284 张展腾  #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;  }  /\*  #include <stdio.h>  #include <math.h>  #include <windows.h>  #include <tchar.h>  float f(float x, float y, float z) {  float a = x \* x + 9.0f / 4.0f \* y \* y + z \* z - 1;  return a \* a \* a - x \* x \* z \* z \* z - 9.0f / 80.0f \* y \* y \* z \* z \* z;  }  float h(float x, float z) {  for (float y = 1.0f; y >= 0.0f; y -= 0.001f)  if (f(x, y, z) <= 0.0f)  return y;  return 0.0f;  }  int main() {  SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 0x0C);  HANDLE o = GetStdHandle(STD\_OUTPUT\_HANDLE);  \_TCHAR buffer[25][80] = { \_T(' ') };  \_TCHAR ramp[] = \_T(".:-=+\*#%@");  int count = 0;  int count1 = 0;  //system("color F4");  for (float t = 0.0f;; t += 0.1f) {  int sy = 0;  float s = sinf(t);  float a = s \* s \* s \* s \* 0.2f;  for (float z = 1.3f; z > -1.2f; z -= 0.1f) {  \_TCHAR\* p = &buffer[sy++][0];  float tz = z \* (1.2f - a);  for (float x = -1.5f; x < 1.5f; x += 0.05f) {  float tx = x \* (1.2f + a);  float v = f(tx, 0.0f, tz);  if (v <= 0.0f) {  float y0 = h(tx, tz);  float ny = 0.01f;  float nx = h(tx + ny, tz) - y0;  float nz = h(tx, tz + ny) - y0;  float nd = 1.0f / sqrtf(nx \* nx + ny \* ny + nz \* nz);  float d = (nx + ny - nz) \* nd \* 0.5f + 0.5f;  \*p++ = ramp[(int)(d \* 5.0f)];  }  else  \*p++ = ' ';  }  }  for (sy = 0; sy < 25; sy++) {  COORD coord = { 0, sy };  SetConsoleCursorPosition(o, coord);  WriteConsole(o, buffer[sy], 79, NULL, 0);  }  if (count <= 10) {  printf("I Love You ———.Mua~\n");//表白内容  printf(" By 编程爱我");// 表白者的名字  count++;  }  else {  printf("You Are My Best Lover.\n");  printf(" Stephen Ge");  count++;  if (count >= 20) {  count = 0;  }  }  Sleep(33);  }  }\*/ |

版权声明

Copyright ©2021-2099 ZhantengZhang.

202030310284 张展腾