#include<iostream>

using namespace std;

class Complex {

public:

double real, img;

Complex(double r = 0 ,double i = 0)

{

real = r;

img = i;

}

};

Complex operator\*(Complex a, Complex b)

{

Complex temp;

temp.real = a.real \* b.real-a.img\*b.img;

temp.img = a.real \* b.img + a.img \* b.real;

return temp;

}

int main()

{

double a,b,c,d;

cout << "请输入两个复数的实部和虚部" << endl;

cin >> a >> b >> c >> d;

Complex A(a, b);

Complex B(c, d);

Complex temp;

temp = A \* B;

cout << "乘积的实部是：" << temp.real << endl;

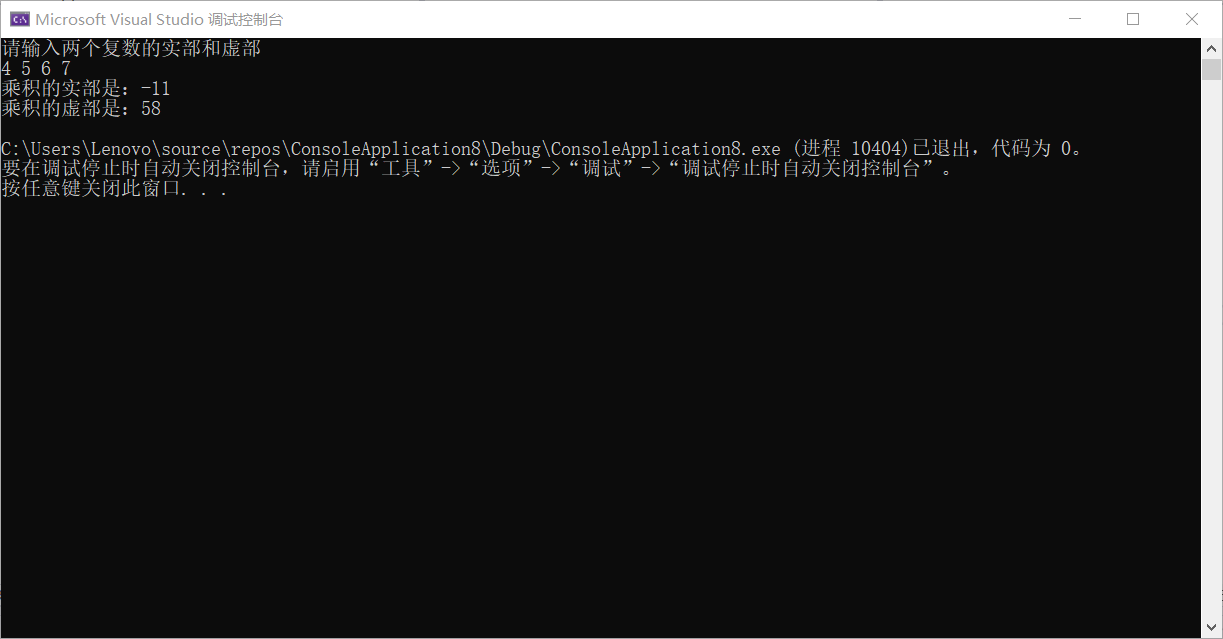
cout << "乘积的虚部是：" << temp.img << endl;

return 0;

}

代码

结果

****

**感悟：c++语言的多态性体现在了重载中，在重载时**\* :: sizeof等等不允许重载，也不许用户虫子自定义运算符。不允许将+ - \* \ 进行无意义的重载，重载也不允许改变运算符的操作对象的个数和运算符原有的结合性

Copyright ©2021-2099 luanxiang wang.All rights reserved