#include <iostream>

using namespace std;

class matrixmul

{

private:

double real;

double imag;

public:

void output();//打印输出

void initSet(double re,double im);//初始化

matrixmul matrixMul(matrixmul Z1,matrixmul Z2);//函数返回值是multirxmul，所以此函数类型为matrixmal

};

matrixmul matrixmul::matrixMul(matrixmul Z1,matrixmul Z2)

{

double temp1,temp2,temp3;

matrixmul result;

temp1=Z1.real\*Z2.imag;

temp2=Z1.imag\*Z2.real;

temp3=(Z1.imag+Z1.real)\*(Z2.real-Z2.imag);

result.real=temp1+temp3-temp2;

result.imag=temp1+temp2;

return result;

}

void matrixmul::initSet(double re,double im)

{

real=re;

imag=im;

}

void matrixmul::output()

{

if(imag>0)

{

cout<<real<<"+"<<imag<<"i"<<endl;

}

else if(imag==0)

{

cout<<real<<endl;

}

else if(imag<0)

{

cout<<real<<imag<<"i"<<endl;

}

}

int main()

{

matrixmul Z1,Z2,Z3,result;

Z1.initSet(1,-2);

Z2.initSet(3,-4);

cout<<"第一个复数Z1为："<<endl;

Z1.output();

cout<<"第二个复数Z2为："<<endl;

Z2.output();

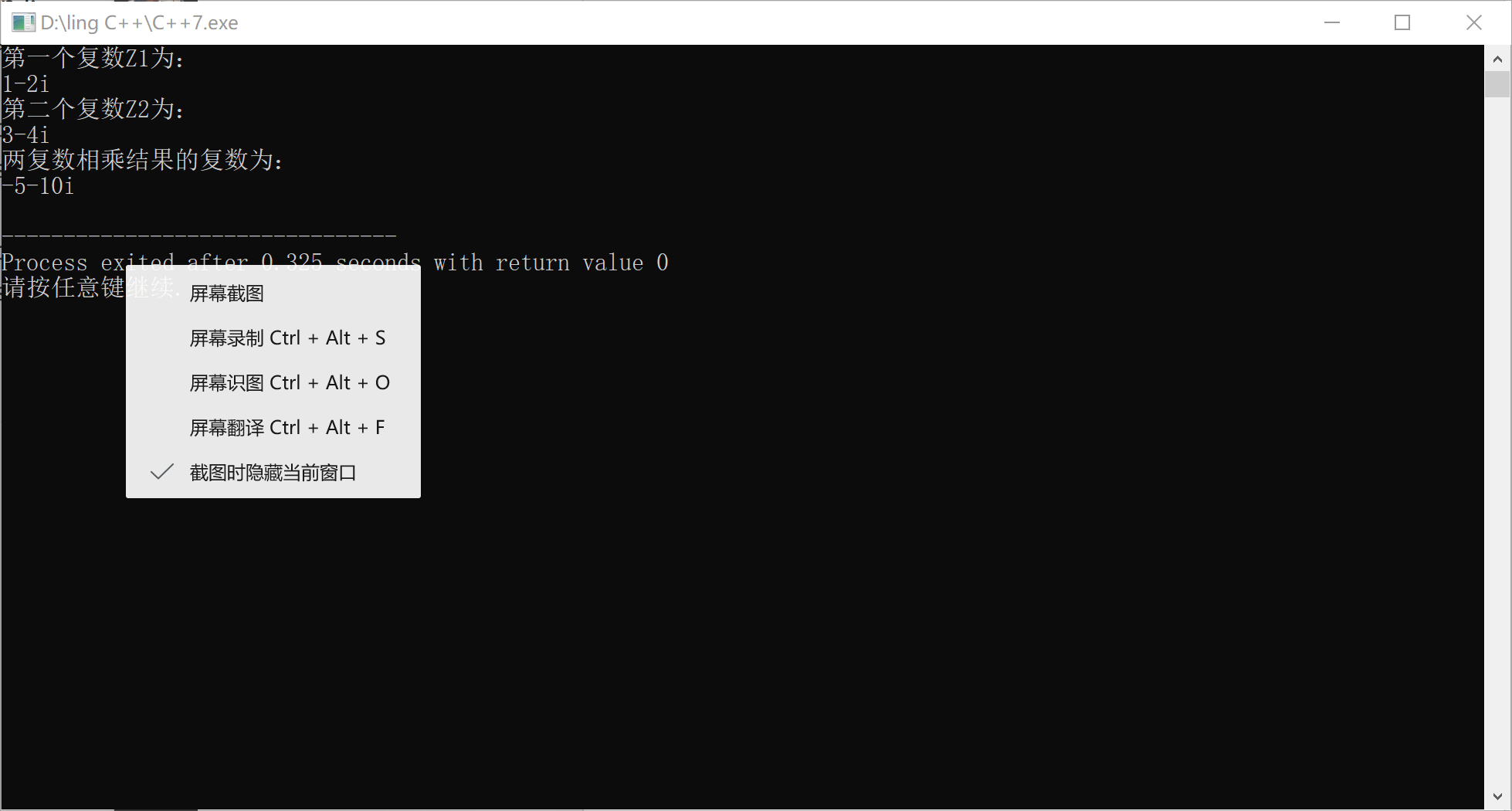
result=Z3.matrixMul(Z1,Z2);

cout<<"两复数相乘结果的复数为："<<endl;

result.output();

return 0;

}



感想：复数类主要考察的是我们对运算符重载的熟练程度，主要有四则运算的+、-、×、/，以及+=、-=、>、<、==、!=前置++，后置++的运算符重载。声明一个复数类Complex，复数的实部 real 和虚部imag分别为该类的两个数据成员，是本章的重点基础。