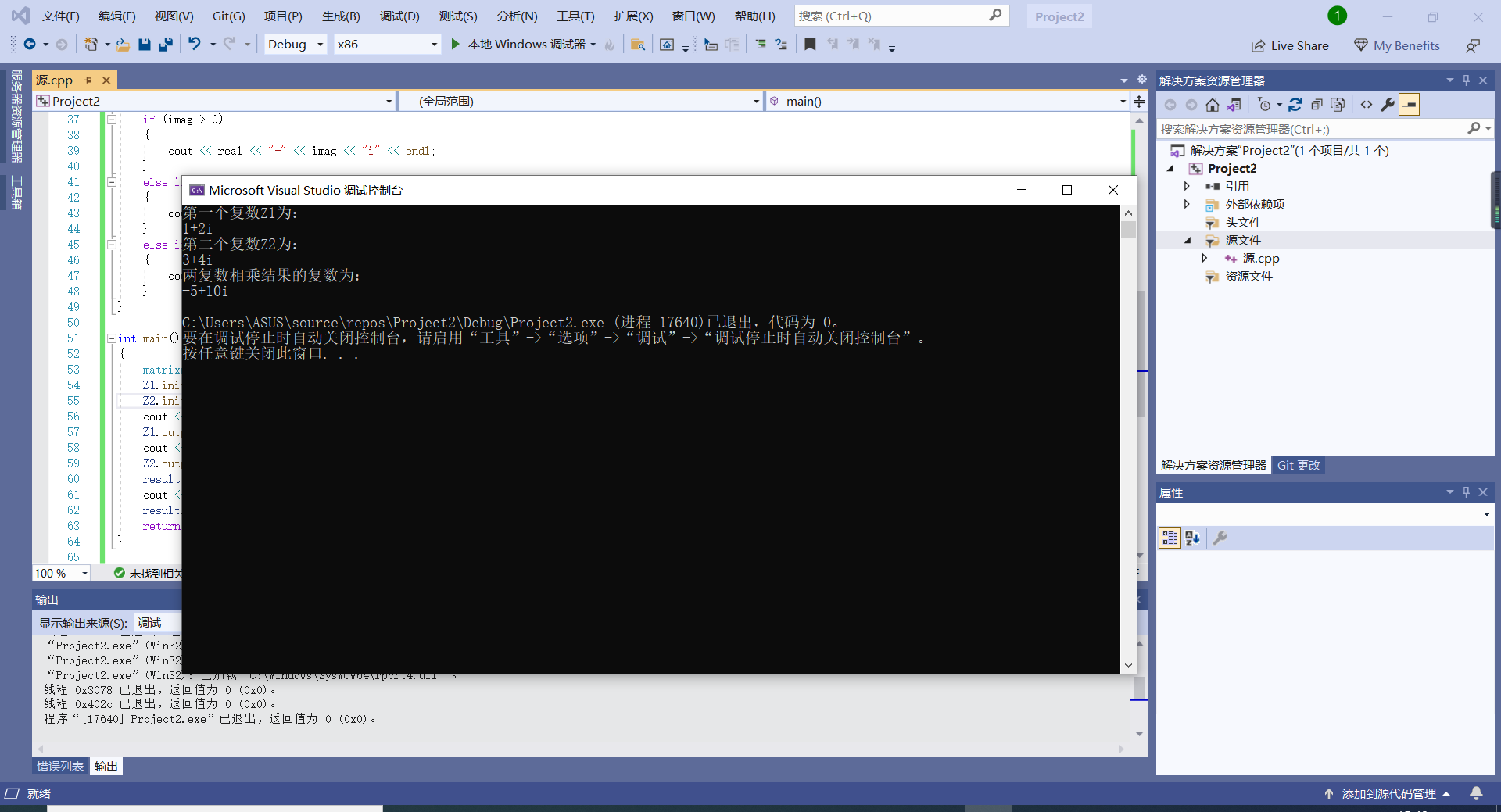
第七次实验报告

实验内容：#include <iostream> using namespace std; class matrixmul{private: double real; double imag;public: void output(); void initSet(double re,double im); matrixmul matrixMul(matrixmul Z1,matrixmul Z2)}; 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;}

运行结果：



体会：

通过这一次的实验操作，了解了运算符重载函数的运用，运算符重载函数本质上是用户为了实现特殊的函数运算而自定义的一种函数，在使用时可将其看作一个运算符号。实验过程中发现自己对于一些细节的把握不够规范，经常容易出台一些细节错误，因此在今后的学习过程中要更加的注意。

Copyright ©2021-2099 YimingYang. All rights reserved