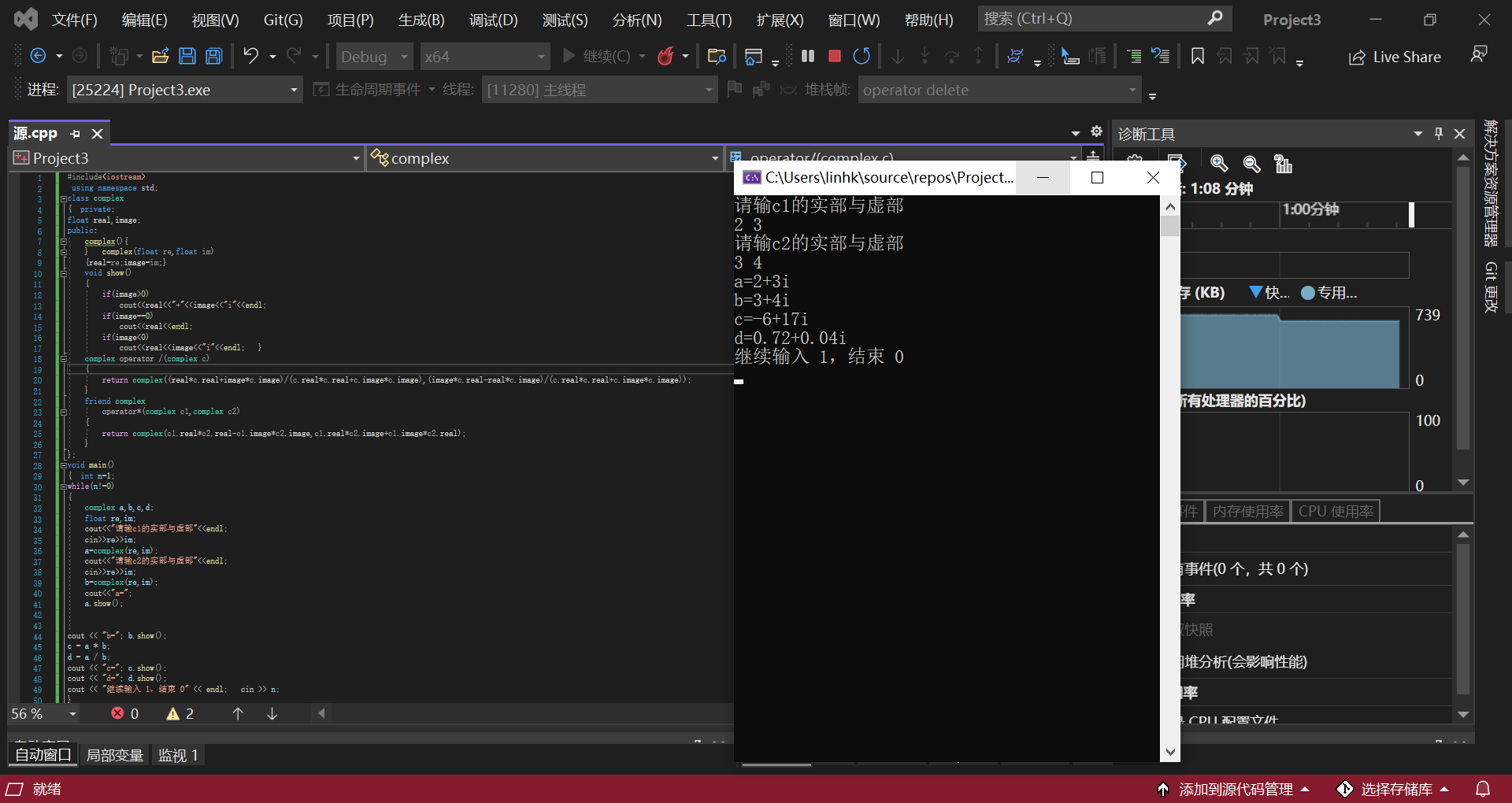
**程序代码（结果）**



#include<iostream>

using namespace std;

class complex

{ private:

float real,image;

public:

complex(){

} complex(float re,float im)

{real=re;image=im;}

void show()

{

if(image>0)

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

if(image==0)

cout<<real<<endl;

if(image<0)

cout<<real<<image<<"i"<<endl; }

complex operator /(complex c)

{

return complex((real\*c.real+image\*c.image)/(c.real\*c.real+c.image\*c.image),(image\*c.real-real\*c.image)/(c.real\*c.real+c.image\*c.image));

}

friend complex

operator\*(complex c1,complex c2)

{

return complex(c1.real\*c2.real-c1.image\*c2.image,c1.real\*c2.image+c1.image\*c2.real);

}

};

void main()

{ int n=1;

while(n!=0)

{

complex a,b,c,d;

float re,im;

cout<<"请输c1的实部与虚部"<<endl;

cin>>re>>im;

a=complex(re,im);

cout<<"请输c2的实部与虚部"<<endl;

cin>>re>>im;

b=complex(re,im);

cout<<"a=";

a.show();

cout << "b="; b.show();

c = a \* b;

d = a / b;

cout << "c="; c.show();

cout << "d="; d.show();

cout << "继续输入 1，结束 0" << endl; cin >> n;

}

}

**心得感悟**

通过友元的方式，某个普通函数或者类的成员函数可以访问某个类中的私有数据，友元提供了一种不同类或对象的成员函数之间、类的成员函数与普通函数之间共享数据的机制。