12.3日上机

1.

#include<iostream>

using namespace std;

class Complex {

public:

double real;

double imag;

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

{

real = r; imag = i;

}

};

Complex operator+(Complex co1, Complex co2)

{

Complex temp;

temp.real = co1.real + co2.real;

temp.imag = co1.imag + co2.imag;

return temp;

}

int main()

{

Complex com1(1.1, 2.2), com2(3.3, 4.4), total1, total2;

total1 = operator+(com1, com2);

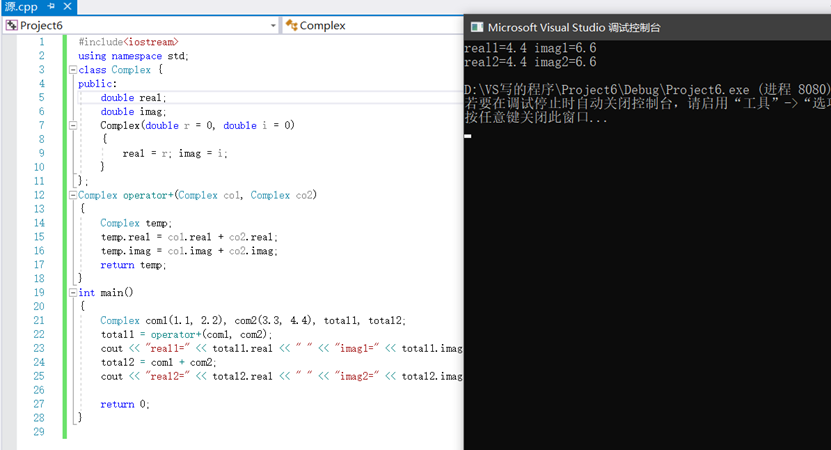
cout << "real1=" << total1.real << " " << "imag1=" << total1.imag << endl;

total2 = com1 + com2;

cout << "real2=" << total2.real << " " << "imag2=" << total2.imag << endl;

return 0;

}



感想心得：

掌握了C++语言多态性的基本概念。掌握了运算符重载函数的声明和定义方法。

运算符重载是通过创建运算符函数实现的，运算符函数定义了重载的运算符将要进行的操作。运算符函数的定义与其他函数的定义类似，惟一的区别是运算符函数的函数名是由关键字operator和其后要重载的运算符符号构成的。运算符函数定义的一般格式如下：

  <返回类型说明符> operator <运算符符号>(<参数表>)

{

     <函数体>

}