

类成员函数形式的运算符重载

运算符重载的方法是定义一个重载运算符的函数。运算符函数重载一般有两种形式：重载为类的成员函数和重载为类的非成员函数。

成员函数形式的运算符重载函数定义的一般形式为：

<返回类型说明符> operator<运算符符号>(<参数表>)

{

 <函数体>

}

- 当运算符重载为类的成员函数时，函数的参数个数比原来的操作数要少一个（后增、后减单目运算符除外）。因为成员函数用this指针隐式地访问了类的一个对象，它就是运算符函数最左边的操作数。
- 调用成员函数运算符的形式如下：
- <对象名> <运算符> <参数>
- 它等价于：
- <对象名>.operator<运算符>(<参数>)

- 【例2-17】 利用成员运算符重载函数实现两个复数对象的加法计算。

```
// Complex.h
class Complex
{
public:
    Complex();
    Complex(double r,double i);
    Complex operator+(Complex &rc); //重载运算符+ , 成员重载
    void Display();
private:
    double m_real;
    double m_imag;
};
```

```
// Complex.cpp
#include "Complex.h"
#include <iostream>
using namespace std;
Complex::Complex()
{
    m_real=0;
    m_imag=0;
}
Complex::Complex(double r,double i)
{
    m_real=r;
    m_imag=i;
}
```

```
Complex Complex::operator+(Complex &rc)
{
    Complex c;
    c.m_real=m_real+rc.m_real;
    c.m_imag=m_imag+rc.m_imag;
    return c;
}
void Complex::Display()
{
    cout<<"("<<m_real<<","<<m_imag
    <<"i)"<<endl;
}
```

```
// testComplex.cpp
#include <iostream>
#include "Complex.h"
using namespace std;
int main()
{
    Complex c1(1,2),c2(3,4),c3;
    c3=c1+c2; //等价于 : c3=c1.operator+(c2);
    cout<<"c1=";
    c1.Display ();
    cout<<"c2=";
    c2.Display ();
    cout<<"c3=c1+c2=";
    c3.Display();
    return 0;
}
```

程序的运行结果为：

c1=(1,2i)

c2=(3,4i)

c3= c1+c2=(4,6i)