

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

三种运算符重载函数

实现运算符重载函数

```
Complex Complex::operator + (Complex &c2)
{
    return Complex(real+c2.real, imag+c2.imag);
}
Complex c3=c1+c2;
```

Q: 为什么说 + 是二元运算 ???!!!

A: 当前对象加参数中提供的对象，隐含地有
operator + (Complex *this, Complex &c2)

换成友元函数定义也可以说明

```
Complex operator + (Complex &c1, Complex &c2)
{
    return Complex(c1.real+c2.real, c1.imag+c2.imag);
}
Complex c3=c1+c2;
```

Q: 如何实现“复数+实数”?

A1：用成员函数实现

```
Complex Complex::operator + (double &d)
{
    return Complex(real+d, imag);
}
```

A2：用友元函数实现

```
Complex operator + (Complex &c, double &d)
{
    return Complex(c.real+d, c.imag);
}
```

Q: 如何调用?

A: Complex c1(3,4),c2(5,-10)

Complex c3=c1+3.14; //正确

Complex c4=3.14+c2; //错误

如何实现？用成员函数，还是用友元函数

实现“复数+实数”

☐ 用成员函数实现

```
Complex Complex::operator + (double &d)
{
    return Complex(this->real+d, imag);
}
```

☐ 用友元函数/一般函数实现

```
Complex operator + (Complex &c, double &d)
{
    return Complex(c.real+d, c.imag);
}
```

如何实现实数+复数？

☐ 用友元函数实现

```
Complex operator + (double &d, Complex &c)
{
    return Complex(c.real+d, c.imag);
}
```

☐ 一般函数实现？

☐ 不能用成员函数实现！

运算符重载函数可以是类的成员函数

- ☐ 它可以通过this指针自由地访问本类的数据成员，少写一个函数的参数，但有要求。

运算符重载函数也可以是类的友元函数

- ☐ 如果运算符左侧的操作数属于C++标准类型(如int)或是一个其他类的对象，则运算符重载函数不能选用成员函数
- ☐ 为方便访问类的私有成员，声明为友元函数为佳

运算符重载函数还可以是普通函数

- ☐ 只有在极少数的情况下才使用（因普通函数一般不能直接访问类的私有成员）

例：复数的三种加法：友元函数实现

```
class Complex{
public:
    friend Complex operator+ (Complex &c1,Complex &c2); //1
    friend Complex operator+ (const double &d,Complex &c); //2
    friend Complex operator+ (Complex &c, const double &d); //3
    void display( );
private:
    double real;double imag;
};

Complex operator + (Complex &c1,Complex &c2) //1
{return Complex(c1.real+c2.real, c1.imag+c2.imag);}

Complex operator + (const double &d,Complex &c) //2
{return Complex(d+c.real, c.imag);}

Complex operator + (Complex &c, const double &d){...} //3
```

```
Complex c1(5,-10), c2(3, 4), c3, c4, c5;
c3=c1+c2;
c4=2+c1;
c5=c1+2.34;
```

惯例

- ❏ 由于友元的使用会破坏类的封装，**原则上要尽量将运算符函数作为成员函数**
- ❏ 综合地，一般**单目运算符重载为成员函数，双目运算符重载为友元函数**
- ❏ 对于数学上交换律成立的运算符，应两次重载运算符（以复数加为例）

```
Complex operator+(Complex &c, int &i)
```

```
{return Complex(i+c.real,c.imag);}
```

```
Complex operator+(int &i,Complex &c)
```

```
{return Complex(i+c.real,c.imag);}
```

- ❏ 有的运算符必须定义为类的成员函数
 - ❏ 如：赋值运算符、下标运算符、函数调用运算符
- ❏ 有的运算符则不能定义为类的成员函数
 - ❏ 如：流插入“<<”和流提取运算符“>>”、类型转换运算符

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net

企业服务：<http://ems.csdn.net/>

人才服务：<http://job.csdn.net/>

CTO俱乐部：<http://cto.csdn.net/>

高校俱乐部：<http://student.csdn.net/>

程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>

项目外包：<http://www.csto.com/>

CSDN博客：<http://blog.csdn.net/>

CSDN论坛：<http://bbs.csdn.net/>

CSDN下载：<http://download.csdn.net/>