

运算符重载为友元函数

郭 炜 刘家瑛

北京大学



运算符重载为友元

- 通常, 将运算符重载为类的成员函数
- 重载为友元函数的情况:
 - 成员函数不能满足使用要求
 - 普通函数, 又不能访问类的私有成员



运算符重载为友元

```
class Complex{  
    double real, imag;  
    public:  
        Complex(double r, double i):real(r), imag(i){ };  
        Complex operator+(double r);  
};  
Complex Complex::operator+(double r){ //能解释 c+5  
    return Complex(real + r, imag);  
}
```



- 经过上述重载后:

Complex c ;

c = c + 5; //有定义 , 相当于 c = c.operator +(5);

- 但是:

c = 5 + c; //编译出错

- 为了使得上述表达式能成立, 需要将+重载为普通函数

```
Complex operator+ (double r, const Complex & c) {
```

```
//能解释 5+c
```

```
    return Complex( c.real + r, c.imag);
```

```
}
```



普通函数不能访问私有成员

→ 将运算符+重载为友元函数

```
class Complex {  
    double real, imag;  
public:  
    Complex( double r, double i):real(r),imag(i){ };  
    Complex operator+( double r );  
    friend Complex operator + (double r, const Complex & c);  
};
```