



中国科学技术大学

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA

地址: 中国安徽省合肥市 电话: 0551-63602184 传真: 0551-63631760 网址: <http://www.ustc.edu.cn>

5. 操作符重载与临时对象

操作符就是一种函数

operator overloading (操作符重载-1, 成员函数) this
inline complex &
complex::operator += (const complex & y)
{ return -dupl(this, y);
}

所有成员函数都自带指针 this 调用者就是 this

inline complex & c3 += c2 += c1;
-dupl(complex * this, const complex & y)
{ this->re += y.re;
this->im += y.im;
return *this;
}

negate
operator overloading
inline bool operator == (const complex & x,
& double y) {
return ... }
共轭复数

这是标准库里面的代码

return by reference 语法分析

传递者无需接收者是以 reference 形式接收
知道

class body 之外的各种定义

operator overloading (操作符重载-2, 非成员函数) 无 this

为了应付 client 的三种可能用法, 这里对应开发三个函数

temp object 临时对象

下面这些函数决不可 return by reference

因为它们返回的必定是个 local object

能不能 return by reference?

typename()

即 int()

return complex(real(x)+real(y), imag(x)+imag(y));

没有名称, 下一行就死亡

```
#include <iostream>
```

```
ostream&
```

```
operator << (ostream& os, const complex& x)
```

```
{ return os << '(' << real(x) << ',' << img(x) << ')';
```

这种特殊的操作符, 只能写成非成员函数的形式

```
cout << conj(c1);
```

参数是什么? 能否 pass by reference? 要不要加 const?

返回类型? 能否 return by reference?

设计一个 class 要注意

1. 构造函数应该使用的特殊语法

2. 函数是否要加 const? 如果该加但你没加, 可能会有副作用

3. 参数的传递尽量使用 reference, 要不要加 const?

4. return 能否 by reference

5. 数据放在 private, 函数放在 public