



程序设计实习

郭炜 微博 <http://weibo.com/guoweiofpku>

<http://blog.sina.com.cn/u/3266490431>

刘家瑛 微博 <http://weibo.com/pkuliujiaying>



复制构造函数 (copy constructor)

基本概念(教材P183)

- 只有一个参数, 即对同类对象的引用。
- 形如 `X::X(X&)` 或 `X::X(const X &)`, 二者选一
后者能以常量对象作为参数
- 如果没有定义复制构造函数, 那么编译器生成默认复制构造函数。默认的复制构造函数完成复制功能。

基本概念

```
class Complex {  
    private :  
        double real,imag;  
};
```

`Complex c1;` //调用缺省无参构造函数

`Complex c2(c1);`//调用缺省的复制构造函数,将 `c2` 初始化成和 `c1` 一样

基本概念

►如果定义的自己的复制构造函数，
则默认的复制构造函数不存在。

```
class Complex {  
    public :  
        double real,imag;  
    Complex(){ }  
    Complex( const Complex & c ) {  
        real = c.real;  
        imag = c.imag;  
        cout << "Copy Constructor called";  
    }  
};  
Complex c1;  
Complex c2(c1);//调用自己定义的复制构造函数，输出 Copy Constructor called
```

基本概念

➤ 不允许有形如 `X::X(X)` 的构造函数。

```
class CSample {  
    CSample( CSample c ) {  
        } //错，不允许这样的构造函数  
};
```

复制构造函数起作用的三种情况

1) 当用一个对象去初始化同类的另一个对象时。

```
Complex c2(c1);
```

```
Complex c2 = c1; //初始化语句，非赋值语句
```

复制构造函数起作用的三种情况

2) 如果某函数有一个参数是类 A 的对象，那么该函数被调用时，类 A 的复制构造函数将被调用。

```
class A
{
    public:
    A() { };
    A(A & a) {
        cout << "Copy constructor called" << endl;
    }
};
```


复制构造函数起作用的三种情况

2)如果某函数有一个参数是类 A 的对象，
那么该函数被调用时，类A的复制构造函数将被调用。

```
void Func(A a1){ }  
int main(){  
    A a2;  
    Func(a2);  
    return 0;  
}
```

程序输出结果为: *Copy constructor called*

复制构造函数起作用的三种情况

- 3) 如果函数的返回值是类A的对象时，则函数返回时，
A的复制构造函数被调用：

```
class A
{
    public:
    int v;
    A(int n) { v = n; };
    A( const A & a) {
        v = a.v;
        cout << "Copy constructor called" << endl;
    }
};
```

复制构造函数起作用的三种情况

3) 如果函数的返回值是类A的对象时，则函数返回时，
A的复制构造函数被调用：

```
A Func() {  
    A b(4);  
    return b;  
}  
  
int main() {  
    cout << Func().v << endl;  
    return 0;  
}
```

输出结果：

Copy constructor called

4