



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>



this 指针

C++程序到C程序的翻译

```
class CCar {  
    public:  
        int price;  
        void SetPrice(int p);  
};  
void CCar::SetPrice(int p)  
{ price = p; }  
int main()  
{  
    CCar car;  
    car.SetPrice(20000);  
    return 0;  
}
```

```
struct CCar {  
    int price;  
};  
void SetPrice(struct CCar * this,  
              int p)  
{ this->price = p; }  
int main() {  
    struct CCar car;  
    SetPrice( & car,  
             20000);  
    return 0;  
}
```

this指针作用

- 其作用就是指向成员函数所作用的对象

this指针作用

➤非静态成员函数中可以直接使用this来代表指向该函数作用的对象的指针。

```
class Complex {  
public:  
    double real, imag;  
    void Print() { cout << real << "," << imag ; }  
    Complex(double r,double i):real(r),imag(i)  
    {  
    }  
    Complex AddOne()    {  
        this->real ++; //等价于 real ++;  
        this->Print(); //等价于 Print  
        return * this;  
    }  
};
```

```
int main() {  
    Complex c1(1,1),c2(0,0);  
    c2 = c1.AddOne();  
    return 0;  
} //输出 2,1
```

this指针作用

```
class A
{
    int i;
public:
    void Hello() { cout << "hello" << endl; }
};
int main()
{
    A * p = NULL;
    p->Hello(); //结果会怎样?
}
```

this指针作用

```
class A
{
    int i;
public:
    void Hello() { cout << "hello" << endl; }
}; → void Hello(A * this ) { cout << "hello" << endl; }
```

```
int main()
{
    A * p = NULL;
    p->Hello(); → Hello(p);
} // 输出: hello
```

this指针作用

```
class A
{
    int i;
public:
    void Hello() { cout << i << "hello" << endl; }
}; → void Hello(A * this ) { cout << this->i << "hello"
    << endl; }
//this若为NULL，则出错！！

int main()
{
    A * p = NULL;
    p->Hello(); → Hello(p);
} // 输出: hello
```


this指针和静态成员函数

静态成员函数中不能使用 this 指针！

因为静态成员函数并不具体作用与某个对象！

因此，静态成员函数的真实的参数的个数，就是程序中写出的参数个数！