



this指针

- 为什么每个对象通过调用类中一个非静态成员函数就能够处理自己的数据呢？

这是由于类的非静态成员函数含有一个指向被调用对象的指针，这个指针就是this指针。this指针是一个隐含于非静态成员函数中的特殊指针，是非静态成员函数中一个类指针类型的形参。当一个对象调用非静态成员函数时，该函数的this指针就指向了这个对象。

- 一般情况下，this指针隐式使用就可以了。但某些情况下需要显式使用。下面是需要显式使用this指针的两种情况：
- （1）非静态成员函数的形参数名与数据成员名相同时，需要显式使用this指针来明确标识哪一个是对象的数据成员。
- （2）非静态成员函数返回的是对象本身或对象地址的时候，需要使用this指针。即直接使用 `"return *this;"` 或 `"return this;"`。

- 【例2-14】 this指针使用示例。下面是Point类的定义，构造函数Point(float x,float y)和普通成员函数SetPoint(float x, float y)的形参与Point类的数据成员同名，此时需要使用this指针。

```
class Point
```

```
{
```

```
public:
```

```
    Point()
```

```
    { x=0, y=0; }
```

```
    Point(float x,float y)
```

```
    { this->x=x; this->y=y; }
```

```
    void SetPoint(float x, float y)
```

```
    { this->x=x; this->y=y; }
```

```
private:
```

```
    float x,y;
```

```
};
```

使用示例：

Point p(2.2f, 3.3f); // 此时执行的构造函数

中this指针指向对象p，即this等于&p

p.SetPoint(3.5f, 4f); // 此时执行的SetPoint

函数中this指针指向对象p，即this等于&p