

运算符重载

2016年11月13日 16:45

1. 单目运算符作为函数成员重载：

- 在类里面声明：

```
Coordinate& operator-();
```

- 定义：

```
Coordinate& Coordinate::operator-()  
{
```

注意引用符号

- 调用：

```
Coordinate coor1(3, 5);
```

```
-coor1;    // coor1.operator-();
```

对比

区别：参数需要引入类的对象

2. 单目运算符作为友元函数重载：

- 在类里声明：

```
friend Coordinate& operator-(Coordinate &coor);
```

- 定义函数：

```
Coordinate& operator-(Coordinate &coor)  
{
```

- 调用：

```
Coordinate coor1(3, 5);
```

```
-coor1;    //operator-(coor1);
```

3. 前置运算符

- 在类里声明：

```
Coordinate& operator++(); //前置++
```

- 定义：

```
Coordinate& Coordinate::operator++()  
{  
    m_iX++;  
    m_iY++;  
  
    return *this;  
}
```

- 调用：

```
++coor1; //coor1.operator++();
```

4. 后置运算符

- 在类里声明：

```
Coordinate operator++(int); //后置++
```

- ☎ 定义：

```
Coordinate operator++(int)  
{  
    Coordinate old(*this);  
    m_iX++;  
    m_iY++;  
  
    return old;  
}
```

- ☎ 调用：

```
coor1++; //coor1.operator++(0);
```

重要