

# 运算符重载概述

## ► 运算符重载

- 运算符重载 (operator overloading) 是一种形式的 C++ 多态。
- 运算符重载就是对已有的运算符重新进行定义，赋予其另一种功能，以适应不同的数据类型。
- 很多 C++ 运算符已经被重载，例如将 \* 运算符用于地址，将得到存储在这个地址中的值；但将它用于两个数字时，得到的是它们的乘积。
- C++ 允许将运算符重载扩展到用户定义的类型。例如，允许使用 + 将两个对象相加。
- 运算符重载只是一种“语法上的方便”，可以使代码看起来更自然，是另一种函数调用的方式。
- 要重载运算符，需要使用被称为运算符函数的特殊函数形式。格式如下：

```
operator op (argument-list)
```

## ► 运算符重载实现

```
int a = 10;  
int b = 20;  
int c = a + b;
```

```
class Time {  
public:  
    int hours;  
    int minutes;  
    Time(int h, int m) {  
        hours = h;  
        minutes = m;  
    }  
}
```

```
Time t1(2, 30);  
Time t2(1, 40);  
Time t3 = t1 + t2;
```

```
Time addTime(Time& t) {  
    Time time;  
    time.hours = hours + t.hours;  
    time.minutes = minutes + t.minutes;  
    return time;  
}
```

```
Time t3 = t1.addTime(t2);
```

## ► 运算符重载实现

```
Time operator+(Time& t){  
    Time time;  
    time.hours = hours + t.hours;  
    time.minutes = minutes + t.minutes;  
    return time;  
}
```

```
Time t1(2, 30);  
Time t2(1, 40);  
Time t3 = t1 + t2;
```

```
Time t3 = t1.operator+(t2);
```

```
// 全局函数实现运算符重载  
Time operator+(Time& t1, Time& t2)  
{  
    // ...  
}
```

# Thanks

