**14 重载运算符与类型转换**

1，基本概念

重载运算符由关键字operator和其后要定义的运算符号共同组成。

重载运算符函数的参数数量与该运算符作用的运算对象数量一样多。

如果运算符函数是成员函数，则它的第一个运算对象绑定到隐式的this指针上，因此成员运算符函数的（显式）参数数量比运算符的运算对象总数少一个。

我们也能像调用普通函数一样直接调用运算符函数。

如果类含有算术运算符或者位运算符，则最好也提供对应的复合赋值运算符。

具有对称性的运算符，例如算术，相等性，关系和位运算等，它们通常是普通的非成员函数。

2，常见的重载运算符

1，输入和输出运算符

2，算术和关系运算符

3，赋值运算符

4，下标运算符

5，递增和递减运算符

6，成员访问运算符

7，函数调用运算符

3，可调用对象与function

可调用对象：函数，函数指针，lambda，bind创建对象以及重载了函数运算符的类。

可以使用function统一管理上述类型：

function<int(int,int)> f1 = add;

function<int(int,int)> f2 = divede();

function <int(int,int)> f3 = [](int i, int j){};

4，类型转换运算符

类型转换函数：operator type() const;

不允许转换成数组或者函数类型，但允许转换成指针或者引用类型。

类型转换函数必须是类的成员函数，不能声明返回类型，形参列表为空。

类型转换函数通常应该是const。

class SmallInt{

public:

operator int() const{return val;}

};

SmallInt si;int num = si + 3;

在实践中，类很少提供类型转换运算符。但是，对于类来说，定义向bool的类型转换还是比较普遍的现象。