

自定义类的运算符重载概述

- C++中预定义的运算符的操作对象只能是基本数据类型。但实际上，对于许多用户自定义类型（例如类），也需要类似的运算操作。这时就必须在C++中重新定义这些运算符，赋予已有运算符新的功能，使它能够用于特定数据类型，这就是运算符重载。运算符重载表现了C++的可扩展性。
- 运算符函数的定义与其他函数的定义类似，惟一的区别是运算符函数的函数名是由关键字operator和其后要重载的运算符符号构成的。

- 运算符重载时要遵循的规则：
- （1）除了类属关系运算符 “.”、成员指针运算符 “.*”、作用域运算符 “::”、sizeof运算符和三目运算符 “?:” 等5种运算符以外，C++中的所有运算符都可以重载。
- （2）重载运算符限制在C++语言中已有的运算符范围内的允许重载的运算符之中，不能创建新的运算符。
- （3）运算符重载实质上是函数重载，因此编译程序对运算符重载的选择，遵循函数重载的选择原则。

- (4) 重载之后的运算符不能改变运算符的优先级和结合性，也不能改变运算符操作数的个数及语法结构。
- (5) 运算符重载不能改变该运算符用于基本数据类型变量的原有含义。它只能和用户自定义类型的对象一起使用，或者用于用户自定义类型的对象和基本数据类型的变量混合使用时。
- (6) 运算符重载是针对新类型数据的实际需要对原有运算符进行的适当的改造，重载的功能应当与原有功能相类似。
- 对于自定义类的运算符重载函数，可以将其定义为类的成员函数，也可以将其定义为类的非成员函数，为了方便，非成员函数一般采用友元函数形式。