第五章习题心得：c++最后的多态，这丰富了c++的一切可能性。

在C+中,多态性的实现和联编(也叫绑定)这一概念有关。一个源程序经过编译、连接,成为可执行文件的过程是把可执行代码联编(或称装配)在一起的过程。其中在运行之前就完成的联编称为静态联编,又叫前期联编;而在程序运行时才完成的联编叫动态联编,也称后期联编。

静态联编是指系统在编译时就决定如何实现某一动作。静态联编要求在程序编译时就知道调用函数的全部信息,因此,这种联编类型的函数调用速度很快。效率高是静态联编的主要优点。

动态联编是指系统在运行时动态实现某一动作。采用这种联编方式,一直要到程序运行时才能确定调用哪个函数。动态联编的主要优点是:提供了更好的灵活性,问题抽象性和程序易维护性。

静态联编支持的多态性称为编译时多态性,也称静态多态性。在C++中,编译时多态性是通过函数重载(包括运算符重载)和模板实现的。利用函数重载机制,在调用同名的函数时,编译系统可根据实参的具体情况确立所要调用的是哪个函数。

动态联编所支持的多态性称为运行时多态性,也称动态多态性。在C++中,运行时多态性是通过虚函数来实现的。

运算符重载的规则。

C++语言对运算符重载制定了以下一些规则:

(1)C++中绝大部分的运算符允许重载,不能重载的运算符只有少数几个。

(2)C++语言中只能对已有的C+运算符进行重载,不允许用户自己定义新的运算符。(3）运算符重载是针对新类型数据的实际需要,对原有运算符进行适当的改造完成的。一般来讲,重载的功能应当与原有的功能相类似(如用“+”实现加法,用“-”实现减法)。

(4)重载不能改变运算符的操作对象(即操作数)的个数。(5)重载不能改变运算符原有的优先级。

(6)重载不能改变运算符原有的结合特性。

(7）运算符重载函数的参数至少应有一个是类对象(或类对象的引用)。

(8）运算符重载函数可以是普通函数﹐也可以是类的成员函数﹐还可以是类的友元函数。

(9)一般而言,用于类对象的运算符必须重载,但是赋值运算符“=”例外,不必用户进行重载。但在某些情况下,例如数据成员中包含指向动态分配内存的指针成员时,使用系统

提供的对象赋值运算符函数就不能满足程序的要求,在赋值时可能出现错误。在这种情况下,就需要用户自己编写赋值运算符重载函数。

友元运算符重载函数和成员运算符重载函数的不同有以下几点:

(1）对双目运算符而言,成员运算符重载函数参数表中含有一个参数,而友元运算符重载函数参数表中含有两个参数;对单目运算符而言,成员运算符重载函数参数表中没有参数,而友元运算符重载函数参数表中含有一个参数。

(2〉双目运算符一般可以被重载为友元运算符重载函数或成员运算符重载函数,但有一些情况,必须使用友元运算符重载函数，例如一个常数与一个对象相加。有的运算符(如“\_”等)只能使用成员运算符重载函数。

(3）成员运算符函数和友元运算符函数都可以用习惯方式调用,也可以用它们专用的方式调用。

Copyright IMG_2562021-2099 wangzezheng. All rights reserved