

高等学校计算机科学与技术应用型教材

MIANXIANG DUIXIANG
CHENGXU SHEJI Visual C++
YU JIYU ACIS DE JIHE ZAOXING

面向对象 程序设计

—— Visual C++与基于ACIS的几何造型

主 编◎李少辉 李 焱 刘 弘

第2版



北京邮电大学出版社
www.buptpress.com

高等学校计算机科学与技术应用型教材

面向对象程序设计

——Visual C++与基于 ACIS 的几何造型

(第 2 版)

主编 李少辉 李 焱 刘 弘



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书是面向对象程序设计的基础及提高教程。书中首先全面介绍了 C++ 程序设计的相关知识和面向对象的相关概念,包括 C++ 数据类型、程序结构、类、对象、继承、重载等;然后介绍了 Visual C++ 集成开发环境、MFC 及应用程序框架以及 MFC 程序设计的方法和思路,包括对话框、资源的创建和使用、简单的图形和文本输出、Windows 标准控件等内容,还介绍了有关数据库编程以及 3D 建模引擎 ACIS 和 HOOPS 的内容。本书把 C++ 的概念与 Visual C++ 可视化设计结合起来,使学生更容易接受。为了使读者更好地掌握本书重点,各章节均配备大量的练习和编程习题。本书是作者总结多年教学实践和科研开发经验写成的,用简单的例子和简练的叙述讲解 C++ 编程,注重理论和实践的结合,使读者在掌握基本理论的同时,提高实际动手能力,适合学习面向对象程序设计的高等院校学生使用,也适用于自学 Visual C++ 的学生和其他程序设计人员。

图书在版编目(CIP)数据

面向对象程序设计:Visual C++ 与基于 ACIS 的几何造型/李少辉等主编. —2 版. —北京:北京邮电大学出版社,2012. 11

ISBN 978-7-5635-3241-4

I. ①面… II. ①李… III. ①C 语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 236573 号

书 名:面向对象程序设计——Visual C++ 与基于 ACIS 的几何造型(第 2 版)

主 编:李少辉 李 焱 刘 弘

责任编辑:王丹丹

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号(邮编:100876)

发 行 部:电话:010-62282185 传真:010-62283578

E-mail:publish@bupt.edu.cn

经 销:各地新华书店

印 刷:北京源海印刷有限责任公司

开 本:787 mm×1 092 mm 1/16

印 张:30.5

字 数:794 千字

印 数:1—3000 册

版 次:2005 年 8 月第 1 版 2012 年 11 月第 2 版 2012 年 11 月第 1 次印刷

ISBN 978-7-5635-3241-4

定 价:59.80 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前言

Introduction

本书是面向对象程序设计的入门教材,选用 Visual C++ 作为语言工具。本书充分体现高校计算机专业的培养目标,在编写过程中突出实践性教学环节,特别是 C++ 程序设计方法的基本技能训练,强调培养学生在实践中的应用能力,进而激发学生的创新能力,推进素质教育。编者严格按照教育部的大纲要求进行编写,内容安排充分体现 21 世纪高等教育教材的特色,按照学生学习程序设计方法的心理顺序整合知识,按模块组织教学,使用时可以根据需要灵活选择,弹性较大。

在教材的编写中加强了改革意识,汲取国内外教材编写的先进思想,编写体系与内容选取注重培养学生适应信息化社会要求和实际程序设计的需要。通过合理组织课程内容,以当前比较流行的应用软件 Visual C++ 作为程序设计的平台,使学生掌握 C++ 的程序结构、面向对象、指针与类等的基本知识和基本技能,能够解决日常生活及工作中的常见问题,在此基础上提高分析问题和解决问题的能力,以及获取计算机新知识、新技术的能力,在毕业后具备较强的实践能力、创新能力和创业能力。

本书从 C++ 语言最基本的概念入手,由浅入深,综合大量的编程实例,引导初学者从入门到掌握 C++ 语言的程序设计方法,每一章都有大量的练习和编程习题,帮助读者掌握相关知识点。主要内容包括 C++ 语言基本数据类型、控制结构、函数、数组、指针、类、面向对象特性、流文件及实践练习,Visual C++ 集成开发环境,MFC 的应用以及可视化程序设计。本书的特点是通俗易懂、面向应用、重视实践,以任务驱动式介绍 C++ 语言的编程方法。教材中的例子都配有相应解释或注释语句,方便读者阅读理解。

本书的第 1 章介绍面向对象程序设计基本概念及学习 C++ 语言所必需的一些基础知识,第 2~6 章介绍了 C++ 程序设计的基本方法,第 7~9 章介绍面向对象的程序设计方法,内容涉及类、继承、重载、多态、虚函数等方面的知识,第 10~16 章介绍 Visual C++ 的 MFC 和应用程序框架以及 MFC 程序设计的方法和思路,包括对话框、资源的创建和使用、简单的图形和文本输出、Windows 标准控件、使用 MFC 进行数据库编程的方法等内容,第 17 章为 3D 建模引擎 ACIS 的简介与环境配置,第 18 章为 Hoops 简介。

对于本科学生,建议理论课与上机课时为 1:1,并适当安排一些习题课。

本书在博士生导师刘弘教授的指导下由李少辉、李焱主编,刘弘教授制定了本书的编写要求及编写大纲。李少辉编辑修改了 1~10 章,第 11~18 章由李焱编写,书中所有程序在 Visual C++ 6.0 下运行通过。

编 者

目录

Contents

第 1 章 概论	1
1.1 面向对象程序设计基本概念	1
1.2 C++的词法及语法规则	7
1.3 C++程序结构的组成	9
1.4 C++程序的书写格式	11
1.5 Visual C++6.0 集成开发环境	12
第 2 章 数据类型和表达式	21
2.1 基本数据类型	21
2.2 常量和变量	22
2.3 运算符	26
2.4 表达式	33
2.5 类型定义	36
第 3 章 顺序、条件和循环结构	40
3.1 顺序结构	40
3.2 条件语句	42
3.3 选择语句	46
3.4 循环语句	48
3.5 break 语句和 continue 语句	57
3.6 转向语句	58
第 4 章 数组	68
4.1 数组类型	68
4.2 字符型数组	73
第 5 章 函数与编译预处理	83
5.1 函数的定义和分类	83
5.2 函数的调用	87
5.3 函数的参数和函数的值	88
5.4 内联函数	95
5.5 函数的重载	97
5.6 函数的嵌套调用和递归调用	99
5.7 作用域	104

5.8 编译预处理	124
第6章 指针和引用	141
6.1 指针	141
6.2 引用	150
6.3 指针与引用的区别	152
6.4 指向数组的指针	153
6.5 字符串指针	156
6.6 结构体	162
6.7 联合体	180
6.8 枚举类型	181
第7章 类和对象	189
7.1 类的定义	189
7.2 对象的定义	193
7.3 对象的初始化	198
7.4 成员函数的特性	204
7.5 静态成员	205
7.6 友元	210
7.7 类的作用域	213
7.8 局部类和嵌套类	214
7.9 对象的生存期	216
7.10 对象指针和对象引用	216
7.11 对象和数组	221
7.12 常类型	224
7.13 子对象和堆对象	228
7.14 类型转换	233
第8章 继承和派生	255
8.1 继承	255
8.2 基类和派生类	257
8.3 单继承	262
8.4 多继承	267
8.5 虚基类	273
8.6 组合	277
第9章 多态性与虚函数	290
9.1 运算符重载	290
9.2 静态联编与动态联编	299
9.3 虚函数	300
9.4 纯虚函数与抽象类	303

9.5 虚析构造函数	306
第 10 章 MFC 应用程序概述	314
10.1 MFC	314
10.2 用 MFC AppWizard 建立应用程序	315
10.3 程序分析	322
第 11 章 GUI 设计及菜单	326
11.1 标准菜单的使用	326
11.2 弹出式子菜单	330
11.3 环境菜单	333
第 12 章 创建和使用对话框	336
12.1 创建和设计对话框	336
12.2 创建对话框类	346
12.3 使用控件按钮	348
12.4 显示模态对话框	355
12.5 使用对话框数据交换和数据确认函数	360
12.6 使用非模态对话框	360
第 13 章 应用程序的组成元素	364
13.1 建立图像、位图和图标	364
13.2 在对话框中使用图形资源	369
第 14 章 简单的图形和文本输出	374
14.1 设备环境	374
14.2 使用画笔	380
14.3 使用刷子	387
14.4 使用字体	395
第 15 章 Windows 标准控件	403
15.1 列表控件	403
15.2 在列表控件中添加项目	407
15.3 进度条控件	415
15.4 滚动条控件	417
15.5 使用滑块控件	422
第 16 章 Visual C++ 数据库编程	426
16.1 Visual C++ 开发数据库的特点	426
16.2 MFC ODBC 数据库访问技术	427
16.3 使用 DAO 技术访问数据库	438
16.4 OLE DB 和 ADO 技术概述	448
第 17 章 ACIS 的简介与环境配置	451
17.1 概述	451



17.2 ACIS 的概念	453
17.3 ACIS 的环境配置	454
第 18 章 Hoops 简介	459
18.1 Hoops 的简介	459
18.2 用 Scheme 语言生成 ACIS 程序	462
18.3 用 Windows 控制台环境编译 ACIS 程序	466
18.4 用 ACIS AppWizard 生成应用程序框架	471
参考文献	477

第 1 章 概 论

本章内容提要

面向对象程序设计简介及相关概念;C++语言的起源、特点、与 C 语言的比较;C++的词法及语法规则;C++程序的结构组成及书写格式;Visual C++6.0 集成开发环境

1.1 面向对象程序设计基本概念

1.1.1 面向对象程序设计的起源及有关概念

1. 面向对象的由来和发展

20 世纪 60 年代开发的 Simula 67 提出了对象的概念,它是面向对象语言的鼻祖。对象是代表着待处理问题中的一个实体,在处理问题过程中,一个对象可以某种形式与其他对象通信。从概念上讲,一个对象是既包含数据又包含有处理这些数据操作的一个程序单元。Simula 语言中也使用了类的概念,类是用来描述特性相同或相近的一组对象的结构和行为,该语言还支持类的继承。继承可将多个类组成为层次结构,进而允许共享结构和行为。

20 世纪 70 年代出现的 Ada 语言是支持数据抽象类型的最重要的语言之一。数据抽象是一种数据结构及作用在数据结构上的操作组成的一个实体。把数据结构隐藏在操作接口的后面,通过操作接口实现外部的交流。对外部来讲,只需知道做什么,而无须知道如何做。再将类型扩展到数据抽象上,即将某种类型的操作汇集起来作为一个整体看待,并与该类型一起看作一个独立的单元,构成了抽象数据类型。因此,可以说,数据抽象类型是数据抽象封装后的类型。它包含了该类型下的操作集和由操作集间接定义的数据类型的值集。Ada 语言中面向对象的抽象结构是包,它支持数据抽象类型、函数和运算符重载以及多态性等面向对象的机制。但是,Ada 语言不是全面的支持继承,因此人们常称它为一种基于对象的语言。

后来出现的 Smalltalk 语言是最有影响的面向对象的语言之一。它丰富了面向对象的概念。该语言并入了 Simula 语言的许多面向对象的特征,包括类和继承等。在该语言中,信息的隐藏更加严格,每种实体都是对象。在 Smalltalk 环境下,程序设计就是向对象发送信息,这个信息将表示为一种操作,如两个数相乘、创建一个新类的对象等。Smalltalk 语言是一种弱类型化的语言,一个程序中的同一个对象可以在不同时间内表现为不同的类型。

20 世纪 80 年代中期以后,面向对象的程序设计语言广泛地应用于程序设计,并且有许多

新的发展,出现了更多的面向对象的语言。归纳起来,大致可分为如下两类。

(1)开发全新的面向对象的语言

其中具有代表性的全新的面向对象的语言有 Object-C,它是在 C 语言上扩展而成的,它是 Smalltalk 语言的变种;Eiffel 语言除了有封装和继承外,还继承了几种强有力的面向对象的特征,它是一种很好的面向对象的语言;Smalltalk 80 语言经历了多次修改和更新,新版本有很大的改进,这类全新的面向对象的语言学习起来要从头开始。

(2)对传统语言进行面向对象的扩展

这类语言又称为混合型语言,它的代表有 C++ 语言,它是在 C 语言的基础上增加了面向对象程序设计的支持。这类语言的特点是既支持传统的面向过程的程序设计,又支持新型的面向对象的程序设计。对于一些已经较好地掌握了 C 语言的人来讲,学习 C++ 语言相对容易一些。另外,C++ 语言具有 C 语言的丰富的应用基础和开发环境的支持,普及起来也相对快些。这些就是 C++ 语言当前得以广泛应用的主要原因。

2. 抽象在面向对象中的作用

(1)抽象的概念

从前面的介绍计算机语言发展的历史来看,语言所提供的抽象支持程序在不断地提高。面向对象的程序设计比面向过程的程序设计要更加强调抽象的重要性。

什么是抽象?一般地讲,抽象是通过从特定的实例中抽取共同的性质以形成一般化的概念的过程。抽象是对某个系统的简化的描述,即强调了该系统中的某些特征,而忽略了一部分细节。对系统进行抽象的描述称为对它的规范说明,对抽象的解释成为它的实现。抽象是具有层次的,可分高层次抽象和低层次抽象两大类。高层次抽象将其低层次抽象作为它的一种实现。

抽象是人们在理解复杂现象和求解复杂问题中处理复杂性的主要工具。

(2)面向对象抽象的实现

面向对象的原理有 4 个,它们分别是:数据抽象、行为共享、进化和确定性。这 4 个原理概括了面向对象计算的本质。

①数据抽象。它为程序员提供了一种对数据和为操作这些数据所需要的算法的抽象。数据抽象包含了两个概念:模块化和信息隐藏。这两个概念既是相互独立的,又是密切相关的。

模块化是将一个复杂的系统分解为若干个模块,每个模块与系统中某个特定模块有关的信息保持在该模块内。一个模块是对整个系统结构的某一部分的一个自包含的和完整的描述。模块化的优点是便于修改或维护,系统发现问题后,可以确定问题出在哪个模块上。这种模块化的设计方法构成了面向对象计算的本质。

信息隐藏是指将一个模块的细节部分对用户隐藏起来,用户只能通过一个受保护的接口来访问某个模块,而不能直接地访问一个模块内部的细节。这个接口一般由一些操作组成,这些操作定义了一个模块(或称实体)的行为。这是复杂问题处理中的一种主要工具。另外,在支持信息隐藏的系统中,错误的影响也通常被限制在一个模块内,增强了系统的可靠性。

数据抽象包含了模块化和信息隐藏这两种抽象,这是面向对象方法的核心。

②行为共享。支持行为共享是面向对象程序设计的第二个原理。行为是数据抽象引进的概念,行为是由实体的外部接口进行定义的。行为共享是指许多实体具有相同的接口,这将增加系统的灵活性。例如,同样的一个操作(如显示,即行为),被系统中的几个实体共享,各个实体对该操作的实现可能不同。这就是行为共享的含义。这种行为共享实际上是增强了在一个

系统中的抽象。

分类和层次分类是支持行为共享的最为明显的方式。行为共享是面向对象计算的另一个重要概念,实现面向对象方法的一个重要的任务是对进行分类的研究。

一个分类是由一组实体共同的行为而构成,因此一个特定分类中的所有实体将共享共同的行为。层次分类更是一种普遍的行为共享形式。层次分类允许一个分类包含另一个分类,层次分类是分类的求精。例如,A 分类被包含在 B 分类中,A 分类和 B 分类各自有特定的行为,而 B 分类将共享这些行为。

③进化。它是面向对象计算的第三个原理。进化是考虑到实际中的需求会很快地发生变化。面向对象的方法要支持进化过程就是要适应可能发生的不断变化,这是需求进化。进化的另一个方面是进化式的问题求解。这种观点是从开始到最终结果是以一种增量的方法逐步地对问题进行求解。特别是对于最终目标不能很好定义的问题,这种方法更具有吸引力。进化的问题将涉及一个系统从开始直到后续的维护这整个生命期。

④确定性。这里确定性是指用于描述一个系统确定的行为。一个确定的系统应该确保其中每个行为项都有一个确切的解释,系统不会因不能响应某一行而失败。这对一个大型系统或者复杂系统显得更为重要。确定性与类型的正确性有关,这实际上就是要求在一个系统中不会出现类型方面的错误。在面向对象的系统中,特别是行为共享和进化等机制增加了确保确定性的困难。在确定某个行为项是否有一个解释是可能的,但要确定这个解释是不可能的,因为解释可能会随时间而变化。

3. 面向对象计算的基本特征

面向对象的系统包含了三个要素:对象、类和继承。这三个要素反映了面向对象的传统观念。

面向对象的语言应该支持这三种要素。首先,应该包括对象的概念。对象是状态和操作的封装体,状态是记忆操作结果的。满足这一点的语言被认为是基于对象的语言。其次,应该支持类的概念和特征,类是以接口和实现来定义对象行为的样板,对象是由类来创建的。支持对象和类的语言被认为是基于类的语言。最后,应该支持继承,已存在的类具有建立子类的能力,进而建立类的层次。支持上述三个方面的语言成为面向对象的语言。按这一标准来衡量,Ada 语言是基于对象的语言,Clu 是基于类的语言,而 Simula 和 Smalltalk 是面向对象的语言。C++也是面向对象的语言。

4. 面向对象方法中的重要概念

(1) 对象

在不同领域中对于对象有不同理解。一般认为,对象就是一种事物,一个实体。在面向对象的领域中,最好从以下两个角度来理解它:一是从概念上讲什么是对象,二是在实际系统中如何实现一个对象。

从概念上讲,对象是代表着正在创建的系统中的—个实体。例如,一个商品销售系统,像顾客、商品、柜台、厂家等都是对象,这些对象对于实现系统的完整功能都是必要的。

从实现形式上讲,对象是一个状态和操作(方法)的封装体。状态是由对象的数据结构的内容和值定义的,方法是一系列的实现步骤,它是由若干操作构成的。

对象实现了信息隐藏,对象与外部是通过操作接口联系的,方法的具体实现外部是不可见的。封装的目的就是阻止非法的访问,操作接口提供了这个对象的功能。

对象是通过消息与另一个对象传递信息的,每当一个操作被调用,就有一条消息被发送到这个对象上,消息带来了将被执行的这个操作的详细内容。一般地讲,消息传递的语法随系统

不同而不同,其他组成部分包括目标对象、所请求的方法和参数。

(2)类

类是创建对象的样板,它包含着所创建对象的状态描述和方法的定义。类的完整描述包含了外部接口和内部算法以及数据结构的形式。

由一个特定的类所创建的对象被称为这个类的实例,因此类是对象的抽象及描述,它是具有共同行为的若干对象的统一描述体。类中要包含生成对象的具体方法。

类是抽象数据类型的实现。一个类的所有对象都有相同的数据结构,并且共享相同的实现操作的代码,而各个对象有着各自不同的状态,即私有的存储。因此,类是所有对象的共同的行为和不同状态的集合体。

(3)继承

类提供了说明一组对象结构的机制,再借助于继承这一重要机制扩充了类的定义,实现了面向对象计算的优越性。

继承提供了创建新类的一种方法,这种方法就是说,一个新类可以通过对已有的类进行修改或扩充来满足新类的需求。新类共享已有类的行为,而自己还具有修改的或额外添加的行为。因此,可以说继承的本质特征是行为共享。

从一个类继承定义的新类,将继承了已有类的所有方法和属性,并且还可以添加所需要的新的方法和属性。新类被称为已有类的子类,而已有类称为父类,又叫基类。新类又叫派生类。

1.1.2 C++语言概述

1. C++语言的起源

正如名字上可以猜测到的一样,C++语言是从C语言继承来的,但这种继承主要表现在语句形式、模块化程序设计等方面。如果从更重要的方面——概念和思想方面来看,C++来源于早期的Simula语言,因为C++语言的最大特征是支持“面向对象的程序设计”。Simula语言被广泛的用于系统仿真,设计它的主要目的是模仿现实世界的真实个体,而使用的主要手段是构造计算机领域的对象来表述现实的个体。由于Simula语言的应用领域并不十分广阔,更重要的一点是它缺乏强有力的开发工具支持,它并没有得到很大的重视。随后推出的另外一种面向对象语言Smalltalk也没有取得太大的成功,很多人认为它没有提供给自己足够的灵活性和如同C语言那样丰富的功能,最关键还在于它和人们早已得心应手的语言并不兼容。比如说,一个C程序员可能会对它的新特性退避三舍,因为C的特性对他是十分的熟悉和亲切的,同时C的确是功能强大的,大多数人不愿放弃这些。

C++的产生正是为了解开这样的“情结”。面对越来越大、越来越复杂的系统,使用C语言已经感到力不从心了,但C语言作为应用域最为广泛的程序设计语言之一,又不能轻易放弃。必须有一种面向对象的程序设计语言,它对C语言有很高的兼容性,使得C程序员只需在原有的知识上进行一定的扩充,就能够方便地进行面向对象的程序设计。

1980年起,Bell实验室的Bjaren Stroustrup博士及其同事开始为这个目标对C语言进行改进和补充。由于这种被扩充和改进的C语言的大量特性与类(class)相关,它最初被开发者称为“带类的C”。但很快人们就认识到这个称呼太片面了,这个“扩展了的C”不仅以标准ANSI C作为子集保留了C语言的全部精华,同时又吸收了Simula 67、Algol 68和BCPL语言

的许多特性,它已远远超过了C语言。随着这个语言的广泛应用和在各个领域取得成果的增多,它给程序设计带来的全新概念和表现出来的远大前景更加卓著,它的开发者因此将C++这一名字赋予它。

2. C++语言的特点

(1) C++是面向对象的程序设计语言

与过去的面向过程的程序设计语言比较,C++的最大特征在于它是面向对象的程序设计语言。面向对象的程序设计是程序设计的一种新思想,该思想认为程序是相互联系的离散的对象集合。面向对象的程序设计语言即是支持这种思想的程序设计语言。

(2) 封装性

C++的封装性,是通过引入“类”而产生的,类将一定数据和关于这些数据的操作封装在一起。这个特点可以显著减少程序各模块之间的不良影响,这在多人协作的程序开发中,好处尤为明显。

(3) 继承性

C++的继承性,是指C++原有程序的代码可以方便地移植到C++的新程序中,而新程序在继承旧程序代码的同时可以增添自己的新内容。继承性使得程序代码的重用率得以很大的提高,使得系统开发过程具有更好的连续性,易于应付用户对于软件不断发展的要求。

(4) 多态性

C++的多态性,是指相似而实质不同的操作可以有相同的名称。例如,“和”的操作,可以是“整数和”,也可以是“矢量和”,在C++中,这两种和的操作都可以简单地称为“和”。C++的多态性使得C++与人的思维习惯更趋一致,用C++编制的程序也更方便人的阅读。

3. C++语言与C语言的关系

(1) C++语言与C语言的联系

C语言也诞生在AT&T的Bell实验室,1972年由Dennis Richie为UNIX设计了这个高级语言,今天C语言的使用已遍及计算机的各个领域。

C语言有以下几个显著的特点:

第一,它是一种结构化的语言,要求一个程序由众多的函数组成,程序的逻辑结构由顺序、选择和循环三种基本结构组成,适宜大型程序的模块化设计。

第二,具有很高的可移植性,这使得C语言程序在保证支持不同硬件环境的前提下,具有较高的代码效率。

第三,它提供了丰富的数据类型和运算,具有较强的数据表达能力,因而在许多不同的场合广泛的应用。

总之,C语言反映了设计者追求高效、灵活,支持模块化设计,从而支持大规模软件开发的愿望。

C++语言保留了C语言设计者的良好愿望,并使得C语言语句成为C++语言的一个子集。一般地,用C语言编写的程序可直接在C++编译器下编译。

(2) C++语言与C语言的主要区别

首先,C++提出了类(class)的概念。类是数据和函数的集合,数据用来描述此类对象的行为。例如,大学生代表在大学读书的一类人,即大学生是一个类,每个具体的大学生都是这个类中的对象。大学生这个类中的数据可以是学生的姓名、性别、年龄、学校、专业、入学时间等,描述此类对象的行为可以是入学、学习、毕业等。



C语言中的结构只是数据的集合,这种结构也可在C++语言中使用。不同的是C++语言将C语言中的“结构”概念扩充成近似于上述“类”的概念,即C++语言中的结构既可以有数据,也可以有函数。

C++语言沿用了C语言中的结构,概念上没有变化。

其次,下列关键字是C++语言新增的: class、private、protected、public、this、new、delete、friend、operate、inline、virtual。

(3) C++语言与C语言的细小区别

①C++语言在保留C语言原有注释方式的同时,增加了行注释。以“//”起始的,以换行符结束的部分是行注释。

②const 关键字,用这个关键字修饰的标识符为恒值常量。它的引入可以替代C语言中的宏定义。比较下面两个语句:

```
#define Number 1
const int Number = 1;
```

它们的功能相同,但后一语句在编译时,编译器对于用到Number的地方,将进行严格的类型检查。

③说明结构、联合、枚举变量时,不必在结构名、联合名、枚举名前加关键字 struct、union 和 enum。例如:

```
/* C语言的说明 */
struct Astruct aS;
union Aunio aU;
enum Bool aBool;
/* C++语言的说明 */
Astruct aS;
Aunio aU;
Bool aBool;
```

(4)变量的说明可以放在程序的任一位置上,例如:

```
for(int i = 0; i < 100; i++)
```

(5)提供了作用域运算符“::”,当有某一全局变量被一个局部变量遮挡时,运用作用域运算符仍可以操作该全局变量。例如:

```
.....
int i;
main()
{
    inti;
    i = 5;           //局部变量 i 赋值
    ::i = 10;        //全局变量 i 赋值
}
```

(6)标准输入/输出一般不再使用C语言的 printf 和 scanf,而使用三个标准 I/O 流。它们是:cout(与标准输出设备相联)、cin(与标准输入设备相联)和 cerr(与标准错误输出设备相联)。在微机上,各设备一般分别为显示器,键盘和显示器。“<<”和“>>”分别被重定义为流的插入和提取操作,例如:

```
cout<<" welcome!";           //向显示器输出" welcome!"
```

```
cin>>a;           //从键盘取得数据到 a
cerr<<" there is an error"; //在显示器上立即显示错误信息
```

4. C++展望

从 1983 年 AT&T 的 Bell 实验室推出了它的 C++ 标准后, C++ 的应用已经广泛的深入到计算机的各个领域, 并取得了很大的成功。对象的概念越来越多的被人们应用。面对越来越复杂的系统, 越来越大型的软件, 面向对象的程序设计从一个新的角度出发, 力图通过使问题空间和解题空间保持更为有效的一致性, 使得计算机软件更加有效和易于理解。作为一种成功的面向对象程序设计语言, C++ 是人们对于客观世界进行清晰和准确描述的有效手段, C++ 使得软件开发者能方便地仿真客观世界。

面向对象方法学的提出是程序设计思想的革命, 很多专家学者都为使得这种思想的更完美实现而努力, 不仅是软件在进行巨大变革, 计算机硬件也受到这个思想的影响而发生变化。所以, C++ 也像当年的 C 语言一样, 给软件设计带来一次新的巨大进步, 它的应用会迅速地普及到计算机技术的各个领域, 成为更有效的工具。

1.2 C++ 的词法及语法规则

这里主要介绍 C++ 的字符集和单词。

1.2.1 C++ 的字符集

字符是可以区分的最小符号。C++ 的字符集由下列字符组成:

1. 大小写英文字母

a~z 和 A~Z

2. 数字字符

0~9

3. 运算符、特殊字符、不可打印出字符

空格 换行符 制表符 ! # % ^ & * _ (下画线) - + = ~ <
> / \ | . , : ; ? ' " () [] {}

1.2.2 单词及语法规则

单词又称词法记号, 它是由若干个字符组成的具有一定意义的最小词法单元。

C++ 共有六种单词, 如下所述。

1. 标识符

标识符是由程序员定义的单词, 是对实体进行定义的一种定义符。常见的有函数的名字、类名、变量名、常量名、对象名、标号名、类型名等。C++ 规定, 标识符是由大小写字母、数字字符(0~9)和下画线组成的, 并且以字母和下画线开始, 其后跟零个或多个字母、数字字符或下画线组成。

定义标识符应该注意以下几点:



(1)标识符的长度(组成标识符的字符的个数)是任意的。但特定的编译系统能够识别的标识符的长度是有限的。有的仅能识别前 32 个。

(2)标识符中大小写字母是有区别的。例如, XyZ、XYZ、xyz、Xyz 等都是不同的标识符。

(3)在实际应用中,尽量使用有意义的单词作标识符。

(4)用户定义标识符时,不要采用系统的保留字,保留字是指系统已预定义的标识符,它包含关键字和设备字等。

2. 关键字

关键字是系统已经预定义的单词,有专用的定义。这些关键词都是保留字,用户不可再重新定义。

auto	break	case	char	class	const
continue	default	delete	do	double	else
enum	explicit	extern	float	for	friend
goto	if	inline	int	long	mutable
new	operator	private	protected	public	register
return	short	signed	sizeof	static	static-cast
struct	switch	this	typedef	union	unsigned
virtual	void	while			

3. 运算符

实际上是系统预定义的函数名字,这些函数作用于被操作的对象,获得一个结果值。运算符通常由一个或多个字符组成。

根据运算符操作的对象个数不同,可分为单目运算符、双目运算符、三目运算符。单目运算符又称为一元运算符,它只对一个操作数进行操作。例如,求负运算符(一)。双目运算符又称为二元运算符,可以对两个操作数进行操作,如加法运算符(+)。三目运算符又称为三元运算符,它可以对三个操作数进行操作。C++中仅有一个三目运算符,即条件运算符(?:)。

4. 分隔符

分隔符又称为标点符号。分隔符是用来分隔单词或程序正文的,它用来表示某个程序实体的结束和另一个程序实体的开始。以下是 C++ 中常用的分隔符。

(1)空格符:常作为单词与单词之间的分隔符。

(2)逗号:作为说明多个变量或对象类型时变量之间的分隔符;或用来作为函数的多个参数之间的分隔符。逗号还可以用作运算符。

(3)分号:仅用来做 for 循环语句中 for 关键字后面括号中三个表达式的分隔符。

(4)冒号:作语句标号与语句之间的分隔符和 switch 语句中关键字 case<整常型表达式>与语句序列之间的分隔符。

还有{}等,是用来构造程序的。

5. 常量

常量是在程序中直接使用符号表示的数据,C++中,常量有数字常量,例如,实型常量(浮点常量)和整型常量(十进制常量、八进制常量、十六进制常量)、字符常量、字符串常量等。

6. 注释符

注释在程序中起到对程序的注释说明的作用,注释的目的是为了便于阅读程序,在程序编译的词法分析阶段,注释将在程序中被删除。

C++中,采用了两种注释方法。

一是使用“/*”和“*/”括起来进行注释,这两个符号之间的所有的字符被作为注释处理,该方法适合于多行注释信息的情况。例如:

```
/* this is my first program to send the messages to the next one. and if you want to change the contents of the program, please run the third one */
```

二是使用“//”,从“//”开始,直到它所在行的行尾,所有字符被作为注释处理。这样注释了一行信息。例如:

```
//this is a comment.
```

7. 空白符

空白符是空格符、换行符、水平制表符的统称。空字符是指 ASCII 码值为 0 的那个字符。空字符不同于空白符,它作为字符串的结束符,有特殊的作用。

1.3 C++ 程序结构的组成

1.3.1 一个 C++ 的示范程序

我们要了解 C++ 程序结构的组成,首先应该了解 C++ 程序结构的特点,下面我们先看一个小的示范程序。

这个程序是用来求两个浮点数之和,这两个浮点数是从键盘上输入的,求得和由屏幕输出显示。下面给出该程序源代码。

【例 1.1】 C++ 的一个示例程序。

```
//This is a C++ program
#include <iostream.h>
void main()
{
    double x,y;
    cout<<" Enter two float numbers:";
    cin>>x>>y;
    double z = x + y;
    cout<<" x + y = "<<z<<endl;
}
```

执行该程序,显示如下信息:

```
Enter two float number:7.2 9.3
```

输入两个浮点数,用空格符作分隔,按回车键后,输出如下结果:

```
x + y = 16.5
```

该程序中,只有一个函数 main(),该函数体内有 5 条语句。

1.3.2 C++ 程序的组成部分

结合上面的示范程序,我们可以看出 C++ 程序有如下的基本组成部分。