

计算机教育丛书



浩强创作室

HAO QIANG STUDIO



电子·教育

谭浩强 主编

非计算机专业教材系列



Visual C++ 6.0 实用教程

● 李凤霞 薛静峰 黄都培 张凤宜 编著

C-43



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>



计算机教育丛书

非计算机专业教材系列

谭浩强 主编

Visual C++ 6.0 实用教程

李凤霞 薛静峰 编著
黄都培 张凤宜

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是为 Visual C++ 6.0 的初级读者所编写的实用教程。全书共 11 章。首先从基本概念入手,介绍了 Visual C++ 6.0 的特点和发展、Visual C++ 6.0 集成开发环境的功能和使用、C++ 语言基础知识和 Windows 环境下进行程序设计的基础知识。然后通过具体的例子说明了用应用程序向导创建 Windows 应用程序的方法,讲述了 Visual C++ 6.0 的主要功能和使用,包括菜单、工具栏、图标、位图和对话框等 Windows 资源的使用、文档与视图的结构、图形处理、各种常用控件的使用等。为了使读者能更深入地了解 Visual C++ 6.0,在第 10 章中简单介绍了 Visual C++ 6.0 的高级应用。最后从应用的角度出发,在第 11 章中给出了一个应用实例,以便读者参考。

全书以面向对象程序设计语言为基础,以 Visual C++ 6.0 开发环境为主线,介绍了面向对象程序设计的基本概念和 Visual C++ 6.0 的具体应用。书中结合应用实例,强调“实用”,力图展示给初学者一个好的面向对象程序设计的入门向导。全书各章配有习题,内容由浅入深,循序渐进,适合作为大专院校的教材和教学参考书,也可供广大计算机工作者和电脑爱好者自学。

图书在版编目(CIP)数据

Visual C++ 6.0 实用教程/李凤霞等编著.-北京:电子工业出版社,2001.9

(计算机教育丛书.非计算机专业教材系列)

ISBN 7-5053-6678-5

I.V… II.李… III.C语言-程序设计-教材 IV.TP312

中国版本图书馆 CIP 数据核字(2001)第 061071 号

丛 书 名:计算机教育丛书 非计算机专业教材系列

书 名:Visual C++ 6.0 实用教程

主 编:谭浩强

编 著:李凤霞 薛静峰 黄都培 张凤宜

责任编辑:陈晓明

特约编辑:雨 寒

排版制作:电子工业出版社计算机排版室

印 刷 者:北京市增富印刷有限责任公司

装 订 者:三河市新伟装订厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:16.75 字数:428.8 千字

版 次:2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

书 号:ISBN 7-5053-6678-5
TP·3723

印 数:6000 册 定价:20.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

《计算机教育丛书》序

20世纪90年代初,我国出现了第二次计算机普及高潮。与80年代初出现的第一次计算机普及高潮相比,这次高潮具有全方位、多层次的特点,各行各业的人都迫切地要求学习计算机知识,掌握计算机的应用。计算机知识已成为当代知识分子知识结构中不可缺少的重要组成部分。计算机既是先进科学技术的结晶,又是大众化的工具。这个特点只有计算机才具备。

过去,计算机只能为少数人所掌握,今天我们要向全中国千百万人民群众普及计算机知识。我们的目标是:把计算机从少数专家手中解放出来,使之成为广大群众手中的工具。我们要破除对计算机的神秘感。实践表明:具有高中以上文化程度的人,是很容易学会计算机的初步操作和应用的。

当然,计算机的应用是分层次的,不同的人在不同的层次上使用着计算机。计算机科学技术内容极为丰富,浩如翰海,它的发展又极为迅速,要在短时期内全部、深入地掌握计算机的知识和应用,几乎是不可能的,我们必须循序渐进、由浅入深、逐步提高。我们说,入门不算难,提高需要下功夫。

对各行各业学习计算机的人员来说,学习计算机的目的是为了应用。应当强调:以应用为目的,以应用为出发点,根据不同工作岗位的特点,需要什么就学什么。实践证明,从学习计算机的应用入手,是学习计算机知识的捷径。

普及计算机教育需要有适用的教材和参考用书。它们应当百花齐放,风格各异,让读者在琳琅满目的书架上能找到自己所需要的书。几年前,我们开始出版《计算机教育丛书》,根据读者的需要,陆续出版了十几本书(主要是供大学生用的教材)。受到社会广大读者的欢迎。许多读者热情地鼓励我们扩展题材,区分层次,不拘一格,推动应用。我们愿意为推动计算机教育与普及贡献自己绵薄之力。

高等学校非计算机专业的学生占全体大学生数90%以上,在这部分学生中进行计算机教育对提高大学生的业务水平和整体素质十分重要。针对当前高校非计算机专业的需要,在本丛书中设置了“大学计算机公共课系列”和“非计算机专业教材系列”。在这两个系列中组织了三个层次的教材(计算机公共基础、计算机技术基础和计算机应用基础),供各校选用,在“非计算机专业教材”系列中还包括了全国计算机等级考试的部分教材,以推动社会上的计算机普及教育。

本丛书的作者多数在各高等学校或研究单位工作,具有丰富教学和研究经验的专家、教授,其中有的同志在我国计算机教育界中享有盛名,颇有建树,并且编写过多种计算机书籍。本丛书遵循的原则是:内容新颖、概念清晰、实用性强、通俗易懂、层次配套。十几年来我们按照这个原则组织了一批教材和其他计算机读物,受到广大读者的欢迎。我们始终认为:作者必须充分了解读者、理解读者、多为读者设想,写出受读者欢迎的作品。我们愿与大家共同努力以实现这个目标。

本系列丛书是由浩强创作室策划和组织编写的。参加工作的有:谭浩强、朱桂兰、薛淑斌、李盘林、徐士良、赵鸿德、边奠英、曲建民、于长云、鲁声清、韩劭、周晓玉、许向荣、孟宪福、高福成、朱淑文等。

由于计算机技术发展十分迅速,加以我们水平有限,本丛书肯定会有不少缺点或不如人意之处,敬请广大读者批评指正。

《计算机教育丛书》主编

谭浩强

1999年7月

前 言

近年来,随着计算机技术日新月异的发展,计算机可视化技术引起了人们越来越多的重视和应用,尤其是 Windows 操作系统的出现,以其新颖的图形用户界面、卓越的多任务操作系统性能、高层次可视化的软件开发平台迅速风靡全球,使得熟悉 DOS 软件开发的用户必须转向设计 Windows 应用程序。为了缩短 Windows 应用程序的开发周期,降低开发难度,一个完全可视化的程序开发环境非常必要。

Visual C++ 6.0 是目前国内最为流行的新一代面向对象的程序开发环境,它是美国微软公司推出的 Visual Studio 6.0 开发工具套件的其中之一。由于 Visual C++ 6.0 具有优越的性能和强大的功能,成为该开发工具套件中最为完善的软件开发系统之一。Visual C++ 6.0 提供了一个高效的 Windows 编程环境,将程序和资源的编辑、编译、调试和运行融为一体,并且提供了大量的程序开发工具,如应用程序向导 AppWizard、类向导 ClassWizard 等。这些工具都高度集成于一个名为 Developer Studio 的开发平台中,利用该开发平台,用户可以自动生成应用程序框架,编写出各种各样的软件。而且,从最简单的基于对话的程序到复杂的多用户程序,都可以直接设计软件的 Windows 用户界面。除此之外,微软基础类库 MFC 使得应用程序框架、类框架及函数框架的自动生成更加快捷方便,大大提高了程序的质量。

Visual C++ 的语言基础是面向对象的 C++。对于程序设计的初级读者来说,Visual C++ 不是一个简单的程序开发工具,而是具有一定的学习难度。本书以汉化的 Visual C++ 6.0 为参考,并标注了原文解释,各章配有习题,在内容的选取和章节的安排上充分考虑到初学者的特点,是 Visual C++ 的入门教程。全书共分为 11 章,其中第 1 章到第 4 章主要介绍了 C++ 语言基础知识、Windows 编程基础知识和 Visual C++ 6.0 的概况、Developer Studio 集成开发环境的功能和初步使用等,旨在使读者了解面向对象程序设计的基本概念和方法,掌握 Windows 环境程序设计的基本特征和规则。第 5 章到第 9 章主要介绍了用 AppWizard 创建 Windows 应用程序、文档和视图、图形处理、对话框和常用控件等 Visual C++ 的具体应用。并通过具体例子说明创建 Windows 应用程序的方法,讲述了 Visual C++ 6.0 的主要功能和使用,包括菜单、工具栏、图标、位图和对话框等 Windows 资源的使用。为了使读者能更深入地了解 Visual C++ 6.0,在第 10 章中简单介绍了 Visual C++ 6.0 的高级应用。最后从应用的角度出发,在第 11 章中给出了一个应用实例,以便读者参考。

本书以面向对象程序设计语言为基础,以 Visual C++ 开发环境为主线,结合应用实例,强调“实用”,力求以通俗易懂的语言,将面向对象程序设计的基础知识和 Visual C++ 6.0 的具体应用展示给读者。书中各章配有思考题和习题,内容由浅入深,循序渐进,适合作为大专院校的教材和教学参考书,也可供广大计算机工作者和软件开发爱好者自学。

参加本书编写的有北京理工大学李凤霞、薛静峰、张凤宜老师及中国政法大学的黄都培老师;北京大学的董理同学和北京理工大学的夏卫国、王尚洋同学参加了本书的习题整理和程序调试工作。本书在编写中自始至终得到了谭浩强教授和薛淑斌老师的认真指导和帮助,也得到了北京理工大学许多同行的关心和支持,在此一并表示衷心的感谢。

鉴于作者水平有限,书中一定存在不少错误和不妥之处,敬请读者批评指正。

作者
于 2001 年 4 月

目 录

第 1 章 Visual C++ 概述	(1)
1.1 由 C 到 C++	(1)
1.1.1 由 C 到 C++	(1)
1.1.2 C++ 对 C 的扩充	(2)
1.2 Visual C++ 的发展概况	(4)
1.2.1 Visual C++ 的诞生	(4)
1.2.2 Visual C++ 的功能和发展	(4)
1.3 Visual C++ 6.0 的新功能特点	(5)
1.3.1 Visual C++ 6.0 的新功能特点	(5)
1.3.2 Visual C++ 6.0 的版本	(6)
1.4 Visual C++ 6.0 的安装与启动	(6)
1.4.1 Visual C++ 6.0 的安装	(6)
1.4.2 Visual C++ 6.0 的启动	(7)
本章小结	(8)
习题	(8)
第 2 章 Visual C++ 6.0 开发环境	(9)
2.1 Developer Studio 集成开发平台	(9)
2.1.1 Developer Studio 主窗口的组成	(9)
2.1.2 菜单栏	(10)
2.1.3 工具栏	(24)
2.1.4 获取联机帮助	(26)
2.2 项目与项目工作区	(28)
2.2.1 什么是项目	(28)
2.2.2 项目工作区	(29)
2.2.3 采用项目开发应用程序	(29)
2.2.4 在项目工作区中添加新项目	(38)
2.3 资源	(39)
2.3.1 资源管理器	(39)
2.3.2 资源符号	(41)
2.3.3 资源符号浏览器	(42)
2.3.4 资源编辑器	(42)
2.4 MFC 库	(47)
本章小结	(48)
习题	(48)
第 3 章 C++ 语言编程基础	(50)

3.1	面向对象编程基础	(50)
3.1.1	什么是面向对象程序设计	(50)
3.1.2	面向对象系统的特征	(51)
3.1.3	面向对象编程的特征	(52)
3.2	一个简单的C++ 程序	(52)
3.3	C++ 语言的基本元素	(53)
3.3.1	基本数据类型	(53)
3.3.2	指针类型	(54)
3.3.3	常量与变量	(55)
3.3.4	标识符与关键字	(55)
3.3.5	基本运算	(56)
3.3.6	控制语句	(57)
3.4	自定义数据类型	(61)
3.4.1	数组	(61)
3.4.2	结构	(61)
3.5	函数	(62)
3.6	C++ 的类与对象	(63)
3.6.1	类	(64)
3.6.2	对象	(66)
3.7	类的函数成员	(66)
3.7.1	构造函数	(66)
3.7.2	析构函数	(67)
3.7.3	友元函数	(68)
3.7.4	函数重载	(70)
3.8	类的继承与派生	(71)
3.8.1	基本概念	(71)
3.8.2	继承机制	(72)
3.8.3	派生类的构造	(72)
3.8.4	派生类的初始化	(72)
3.8.5	关于派生类的进一步说明	(74)
3.8.6	多重继承	(75)
3.9	虚拟函数和多态性	(77)
3.9.1	虚拟函数	(77)
3.9.2	多态性	(78)
	本章小结	(79)
	习题	(79)
第4章	Windows 编程基础	(80)
4.1	Windows 98 简介	(80)
4.2	Windows 应用程序设计的特点	(82)
4.2.1	事件驱动的程序设计	(83)

4.2.2	消息循环与输入	(83)
4.2.3	图形输出	(84)
4.2.4	用户界面对象	(86)
4.2.5	资源共享	(89)
4.2.6	Windows 应用程序组成	(89)
4.3	Windows 98 应用程序接口(API)	(91)
4.3.1	什么是 API	(91)
4.3.2	Win32 API	(91)
4.4	Windows 98 应用程序基础	(92)
4.4.1	WinMain()	(92)
4.4.2	窗口函数	(92)
4.4.3	窗口类	(93)
4.4.4	消息循环	(93)
4.4.5	Windows 数据类型	(93)
4.5	Windows 98 程序框架	(94)
4.5.1	定义窗口类	(96)
4.5.2	创建窗口	(98)
4.5.3	消息循环(Message Loop)	(99)
4.6	Windows 应用程序的开发工具	(100)
	本章小结	(101)
	习题	(101)
第 5 章	创建 Windows 应用程序	(103)
5.1	创建 Windows 应用程序的方法	(103)
5.2	用 AppWizard 创建应用程序框架	(103)
5.2.1	准备工作	(104)
5.2.2	生成应用程序框架	(104)
5.2.3	应用程序的编译、链接与运行	(109)
5.3	浏览新项目	(110)
5.3.1	新项目的目录和文件	(110)
5.3.2	工作区中的 hello 项目	(111)
5.4	应用程序框架结构分析	(112)
5.4.1	应用程序中的类	(112)
5.4.2	应用类 CHelloApp	(113)
5.4.3	主窗口类 CMainFrame	(119)
5.4.4	文档类 CHelloDoc	(119)
5.4.5	视图类 CHelloView	(120)
5.5	新项目中的资源	(121)
5.6	应用程序控制流程	(122)
5.7	在框架程序中添加新内容	(123)
5.7.1	修改源程序	(123)

5.7.2	编译运行程序	(123)
5.8	新项目中的辅助文件	(124)
5.8.1	项目文件	(124)
5.8.2	资源文件	(124)
5.8.3	其他文件	(125)
	本章小结	(125)
	习题	(125)
第6章	文档与视图	(127)
6.1	概述	(127)
6.1.1	Visual C++ 6.0 对文档和视图的支持	(127)
6.1.2	文档与视图的关系	(127)
6.1.3	文档与视图的接口	(128)
6.2	文档与视图	(128)
6.2.1	MFC 各对象之间的关系	(128)
6.2.2	文档类	(129)
6.2.3	视图类	(131)
6.2.4	多文档	(132)
6.2.5	多视图	(133)
6.2.6	分割窗口	(134)
6.2.7	文档和视图的初始化及清理	(135)
6.2.8	文档与视图的滚动和缩放	(136)
6.3	菜单	(137)
6.3.1	建立一个菜单	(138)
6.3.2	创建浮动的弹出式菜单	(139)
6.4	文本处理	(140)
6.4.1	文本输出函数	(140)
6.4.2	设置文本属性	(141)
6.4.3	获取字符属性	(142)
6.4.4	使用字体	(143)
6.4.5	文本处理示例	(145)
6.5	打印和打印预览	(146)
	本章小结	(148)
	习题	(148)
第7章	图形设备	(150)
7.1	在屏幕上绘图	(150)
7.1.1	设备环境与图形设备接口	(150)
7.1.2	坐标系统与映射模式	(151)
7.1.3	绘图模式	(153)
7.1.4	设置背景颜色及模式	(153)
7.1.5	绘图工具	(154)

7.1.6	绘图函数	(158)
7.2	位图	(161)
7.2.1	位图与位图的建立	(161)
7.2.2	从资源中装载位图	(161)
7.2.3	在按钮上放置位图	(162)
7.3	图标	(163)
7.3.1	认识图标	(163)
7.3.2	图标的类型	(163)
7.3.3	创建一个新的图标	(163)
7.3.4	在程序中加载图标	(163)
7.3.5	在按钮上显示图标	(164)
7.4	光标	(165)
7.4.1	认识光标	(165)
7.4.2	使用光标	(165)
7.4.3	将光标改为沙漏	(165)
	本章小结	(166)
	习题	(166)
第 8 章	对话框	(167)
8.1	什么是对话框	(167)
8.2	创建对话框	(167)
8.2.1	添加消息框	(167)
8.2.2	添加对话框	(169)
8.2.3	修改对话框的属性	(170)
8.2.4	设置对话框的背景颜色	(172)
8.2.5	给对话框添加控件	(172)
8.2.6	创建一个对话框类	(173)
8.2.7	添加消息句柄来初始化对话框	(174)
8.2.8	给新对话框添加选择菜单	(175)
8.3	创建基于对话框的项目	(177)
8.3.1	用 AppWizard 创建基于对话框的项目	(177)
8.3.2	浏览 HelloDialog 项目	(177)
8.3.3	使用对话框编辑器	(178)
8.4	通用对话框类	(178)
8.4.1	CFontDialog 类	(179)
8.4.2	CColorDialog 类	(179)
8.4.3	CFileDialog 类	(180)
	本章小结	(180)
	习题	(180)
第 9 章	常用控件	(182)
9.1	静态文本控件	(182)

9.2	按钮控件	(183)
9.2.1	什么是按钮控件	(183)
9.2.2	应用按钮控件的实例	(183)
9.2.3	按钮控件的属性	(184)
9.2.4	按钮成员变量的设置	(186)
9.2.5	用条件语句调控程序	(186)
9.2.6	启用或禁用按钮	(188)
9.2.7	隐藏按钮	(189)
9.2.8	定义或设定 Tab 键切换顺序	(189)
9.3	编辑控件	(190)
9.3.1	什么是编辑控件	(190)
9.3.2	应用编辑控件的实例	(190)
9.3.3	编辑控件的属性	(191)
9.3.4	添加 CEdit 对象	(192)
9.3.5	接收从编辑控件中输入的文本	(193)
9.3.6	给对话框传递参数	(193)
9.4	列表框控件	(195)
9.4.1	什么是列表框	(195)
9.4.2	应用列表框控件的实例	(196)
9.4.3	列表框属性	(196)
9.4.4	使用 CListBox 类	(197)
9.4.5	增加和删除列表项	(197)
9.4.6	接收列表框消息	(198)
9.5	组合框控件	(199)
9.5.1	什么是组合框	(199)
9.5.2	应用组合框控件的实例	(200)
9.5.3	组合框的属性	(201)
9.5.4	对组合框的操作	(201)
9.6	旋钮、滑块和进度控件	(204)
9.6.1	旋钮控件	(204)
9.6.2	滑块控件	(206)
9.6.3	进度控件	(207)
9.6.4	滑块控件和进度控件的协同使用	(208)
	本章小结	(209)
	习题	(209)
第 10 章	高级应用	(211)
10.1	实现多任务	(211)
10.1.1	创建进程	(211)
10.1.2	创建多线程	(212)
10.1.3	同步	(213)

10.1.4	同步对象	(214)
10.2	动态链接库 DLL	(215)
10.2.1	什么是 DLL	(215)
10.2.2	用 AppWizard 创建 DLL	(216)
10.2.3	DLL 的使用与搜索	(217)
10.3	数据库	(218)
10.3.1	ODBC	(218)
10.3.2	DAO	(220)
10.4	多媒体应用程序的开发	(221)
10.4.1	数据格式	(221)
10.4.2	播放函数	(222)
10.4.3	声音服务	(222)
10.4.4	控制接口	(223)
10.4.5	声音压缩管理器	(223)
10.4.6	AVFile 函数	(224)
10.4.7	视频压缩管理器	(224)
10.4.8	视频捕获	(224)
10.4.9	简单的例程	(225)
	本章小结	(225)
第 11 章	综合实例	(226)
11.1	创建应用程序的基本框架	(226)
11.2	视图的处理	(227)
11.2.1	在视图类中添加数据成员	(227)
11.2.2	用 ClassWizard 映射鼠标消息	(227)
11.2.3	添加消息处理函数代码	(228)
11.2.4	定制视图窗口	(229)
11.2.5	建立并运行程序	(230)
11.3	文档的处理	(230)
11.3.1	在文档类中存放直线坐标	(230)
11.3.2	画完直线后存放直线坐标	(232)
11.3.3	修改视图类的 OnDraw 成员函数	(232)
11.3.4	删除文档数据	(232)
11.3.5	处理“撤消”命令	(232)
11.3.6	建立并运行程序	(234)
11.4	串行化处理	(234)
11.4.1	往文档类中添加串行化代码	(234)
11.4.2	编写 CMyLine 对象的串行化代码	(235)
11.4.3	设置修改标记	(235)
11.4.4	建立并运行程序	(236)
11.5	滚动处理	(236)

11.5.1	添加滚动功能	(236)
11.5.2	转换坐标	(236)
11.5.3	限制绘图区域的大小	(236)
11.5.4	建立并运行程序	(236)
11.6	窗口的分割	(237)
11.6.1	修改主边框窗口类	(237)
11.6.2	更新视图	(237)
11.6.3	建立并运行程序	(237)
11.7	创建对话框模板资源	(238)
11.8	创建对话框类并定义成员变量	(240)
11.9	定义消息处理函数	(242)
11.10	修改对话框类代码	(242)
11.11	创建对话类对象	(245)
11.12	添加工具栏按钮	(250)
11.13	编译并运行应用程序	(251)
参考文献		(253)

第 1 章 Visual C++ 概述

作为全书的开篇,本章将对 Visual C++ 的发展概况、主要功能特点以及 Visual C++ 的运行环境和安装过程作简要介绍,以使读者对 Visual C++ 有一个概括的认识。在此同时,本章简要介绍了 C 与 C++ 的主要区别,目的在于使具有 C 程序设计基础的读者找到掌握 Visual C++ 的便捷途径。

1.1 由 C 到 C++

在没有具体讲述 Visual C++ 的功能特点和使用方法之前,先了解一下 Visual C++ 的发展,这将对初学者是有益的。

1.1.1 由 C 到 C++

1. 面向过程的 C 语言

1972 年,一个 C 编程语言在 AT&T 贝尔实验室问世,它是由 Dennis Richie 写成的。C 语言最初只是供 16 位的 DEC PDP-11 小型计算机使用,用它将 UNIX 操作系统模块化并编写了大部分操作系统和系统应用程序。由于 UNIX 操作系统的成功和广泛使用,加之 C 语言本身具有既可作系统编程语言也可作应用程序语言的设计基础,使得 C 语言迅速发展并及时进行标准化,在 80 年代,便成为一种在各种机型和各种操作系统上都可使用的极其流行、应用广泛的高级程序设计语言。广泛流行的 C 编译系统是以 1987 年公布的新标准——87 ANSI C 为基础的。在微型机上使用的 C 语言编译系统有 Microsoft C, Turbo C, Quick C 等。

2. 面向对象语言的诞生

C 语言是面向过程的,它支持长久以来推行的结构化程序设计。随着软件需求的迅速膨胀和软件技术的不断发展,面向过程程序设计在许多方面显得力不从心,为此,提出了面向对象程序设计(Object-Oriented Programming,简称为 OOP)的思想。其实“对象”的概念早在 20 世纪 60 年代就被提出来了,并且推出了第一个面向对象程序设计语言 Simula-67。此后,随着 OOP 技术的不断发展,产生了大量的 OOP 语言,其中最著名的就是 80 年代初期出现的 Smalltalk 语言。Smalltalk 丰富了面向对象的概念,并引入了类、方法、实例等概念和术语,应用了继承机制和动态链接,被认为是一个真正的、完全的面向对象程序设计语言。这是在 C++ 之前最成功的面向对象程序设计语言。到了 20 世纪 80 年代中期以后,面向对象语言已广泛地应用于程序设计中,并且有了许多新的发展,出现了更多的 OOP 语言。

3. 面向对象的 C++ 语言

C++ 是在 C 的基础上发展起来的支持面向对象的程序设计语言。为 C 语言增加面向对

象的类的思想,是 Bjarne Stroustrup 于 1979 年提出的。他首要的研制目标是使 C++ 首先是一个更好的 C,即 C++ 要继承 C 的所有优点,要根除 C 中存在的问题。1980 年 Bjarne Stroustrup 写出了他的第一个语言——C with Classes,称为“带类的 C”。1984 年,这种语言发展为 C++。

C++ 的确达到了它预期的目标。它不仅提供了对 C 语言的向后兼容性,保持了 C 的简洁、高效和接近汇编语言的特点,而且还引入了对象、类、继承等机制,使之能够支持抽象数据类型和继承性,对 C 的类型系统、编译器的纠错能力、开发环境等诸多方面进行了改进和扩充。这使得 C++ 首先成为一个更好的 C,其次实现了使 C++ 成为面向对象程序设计语言的目标。

早期的许多具有 OOP 性能的程序语言,虽然具有面向对象的特征,但不能方便地刻画可视化对象,与用户的交互能力也较差,在程序开发过程中,仍然是依靠程序员编写大量的程序代码。C++ 的推出大大地改变了这种状况,也就是说,用户使用 C++ 时不必自己建立对象,只要在提供的程序框架中加入完成功能的代码,其余的都交给 C++ 去做。C++ 允许在一个具有真正 OOP 扩展的可视化编程环境中使用它的面向对象语言,这种创造性的组合,使得可视化编程与面向对象的开发框架紧密地联系在一起,为 C++ 的程序开发环境创造了广阔的发展空间。

自 1990 年起, C++ 的标准化工作就已经开始了。1995 年,第一个真正的 C++ 国际标准 ANSI/ISO C++ 被认可,这使得 C++ 的应用更为广泛,支持 C++ 的各种开发环境也在相继推出。最初的 C++ 开发环境只是采用一个前端的预处理器实现的。它将 C++ 首先转换成 C,然后用一个原始的 C 编译器进行编译。从 1985 年至 1991 年间, C++ 语言经过了不断的完善和发展,各种 C++ 的编译器都实现了既作为 C 的前端又作为原始编译器的语言这个目标,并且建立了多种编程支持环境,例如 Microsoft C++、Turbo C++、Borland C++ 等。尤其是 Microsoft C/C++ 7.0 的出现,曾在世界软件行业掀起了一场不小的风波。它不但标志着面向对象技术的成熟和完善,同时也开创了以面向对象技术为主导的软件设计的新时代。

为什么 C++ 会如此成功? 首先 C++ 是更好的 C,它是 C 语言的一个超集。因为 C++ 的目标不只是为了将 C 语言转变成 OOP 语言,而且还为了解决当今程序员,特别是那些在 C 语言中已经大量投资的程序员所面临的许多问题。所以从开始研制 C++ 时,就注重了对 C 的兼容要求。但是,值得一提的是, C++ 对 C 的兼容也使得 C++ 受到了一定的限制。因为要保持这种兼容, C++ 就必须支持 C 的面向过程特性,这样,两种不同风格的程序设计技术溶于同一种语言中,使得 C++ 不是一个纯正的 OOP 语言。将 C 转变到 OOP 需要代价,而转变到 C++ 所花的代价较小。因此说 C++ 的成功是经济上的。

在一个纯粹的 OOP 语言中,几乎所有的语言成分都是“对象”,这种语言强调开发快速原型的能力;而混合型的面向对象语言,是在原传统的过程化语言中加入了各种面向对象的语言机制,它所强调的是开发效率和运行效率。实际上 C++ 就是这种混合型的面向对象语言,它的目的是提高效率。效率取决于很多东西,而语言是为了尽可能地帮助使用者,尽可能不用武断的规则或特殊的性能妨碍使用者。所以, C++ 的成功也正是因为它立足于实际:尽可能地程序员提供最大便利。

1.1.2 C++ 对 C 的扩充

C++ 与 C 语言的本质差别就在于 C++ 是对面向对象程序设计的支持。在对 C 语言改进的基础上, C++ 又增加了支持面向对象的新内容,具体可概括为以下七个方面。

(1) C++ 支持数据封装。支持数据封装就是支持数据抽象。在 C++ 中,类是支持数据封装的工具,对象则是数据封装的实现。在面向过程程序设计中,数据只被看成是一种静态的结

构,它只有等待调用函数对它进行处理;在C++中,将数据和对该数据进行操作的函数封装在一起作为一个类的定义。除此之外,封装还提供了一种对数据访问严格控制的机制,数据被隐藏在封装体中,该封装体只通过接口与外界交换信息。

(2)C++类中包含了私有、公有和保护成员。私有、公有和保护成员是C++类中可以定义的三种不同访问权限的成员。对私有(private)成员的访问仅限于在类中说明的函数,类外的函数不能访问私有成员;对公有(public)成员,类外的函数也可以访问,成为该类的接口;对保护(protected)成员,C++规定只有该类的派生类可以访问,而所有在类外的函数不能访问保护成员。

(3)C++中通过发送消息来处理对象。每个对象都是根据所接收到的消息的性质来决定需要采取的行动,以响应这个消息。因此,发送到一个对象的所有可能的消息在对象的类描述中都需要定义,即对每个可能的消息给出一个相应的方法。方法是在类定义中使用函数来定义的,使用类似于函数调用的机制把消息发送到一个对象上。

(4)C++支持继承性。C++支持单继承和多继承,即一个类可以生成一个或多个派生类,派生类继承了基类的所有方法,还可以定义新方法,一个子类的每个对象都包含了从父类继承来的数据成员和自己特有的数据成员。

(5)C++中允许友元破坏封装性。前面已经提到,类中的私有成员一般不允许类外的任何函数访问,但是友元函数可以例外,它可以访问该类的私有成员。

(6)C++中允许函数名和运算符重载。函数名重载和运算符重载都属于多态性。C++支持多态性,它允许相同的标识符或运算符代表多个不同实现的函数。

(7)C++支持动态联编。动态联编是多态性的一个重用特征,C++中允许定义虚函数,通过虚函数来支持动态联编。

以上是C++对面向对象程序设计的一些主要特征的支持。除此之外,C++对C语言主要有以下一些改进。

(1)增加了一些新的运算符。例如,增加了::、new、delete、*、->*等。其中作用域运算符::用来解决“名冲突”,例如在一个函数内说明了一个局部变量sum,另外还有一个全局变量sum,则::sum允许在局部变量sum的作用域内访问全局变量sum。这使得C++应用起来非常方便。运算符::还用在类中。

(2)C++中可以定义强制数据类型,所谓强制数据类型是指在诸如函数自变量和返回值以及变量赋值等内容中,C++规定数据类型转换一般采用强制匹配,提高了系统的安全性。另外,C++规定函数的说明用原型,对缺省类型作了相应的限制,增加了编译系统检查类型的能力。

(3)允许函数重载,允许设置缺省参数。这些措施提高了编程的灵活性,减少冗余性。同时引进了内联函数的概念,提高了程序的效率。

(4)允许函数参数未予指明。例如,C++能用省略号(...)将函数指明为具有未知个数和类型的参数,使用这一特性来抑制参数类型检验,并容许在该函数的接口上有灵活性。

(5)对变量的说明更加灵活。C语言规定在函数体或分程序内,只能先是对变量说明的语句,然后是执行语句,两者不可交叉使用。C++打破了这一限制,可以根据需要随时进行变量说明。

另外,C++与C还有一些虽小但很重要的区别:在C中,字符常量自动转换成整数,而在C++中不作这种转换;在C中,多次说明一个全程变量并不算错,但在C++中这样做是不可以的;在C中标识符最多为31个字符长,C++中不存在这样的限制等等。