

计 算 机 系 列 教 材

Visual C++ 程序设计基础教程

主编 李春葆 章启俊



WUHAN UNIVERSITY PRESS

武汉大学出版社

计 算 机 系 列 教 材

Visual C++ 程序设计基础教程

主编 李春葆 章启俊



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

Visual C++程序设计基础教程/李春葆,章启俊主编. —武汉: 武汉大学出版社, 2006. 5

计算机系列教材

ISBN 7-307-04977-5

I. V... I. ①李... ②章... III. C语言—程序设计—教材
IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 024291 号

责任编辑: 黄金文 史新奎

责任校对: 王 建

版式设计: 支笛

出版发行: 武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件: wdp4@whu.edu.cn 网址: www.wdp.com.cn)

印刷: 湖北省京山德新印务有限公司

开本: 787×980 1/16 印张: 27.375 字数: 561 千字

版次: 2006 年 5 月第 1 版 2006 年 5 月第 1 次印刷

ISBN 7-307-04977-5/TP·198 定价: 35.00 元

版权所有, 不得翻印; 凡购我社的图书, 如有缺页、倒页、脱页等质量问题, 请与当地图书销售部门联系调换。

计算机系列教材

编 委 会

主 任：王化文，武汉科技大学中南分校信息工程学院院长，教授
编 委：(以姓氏笔画为序)

万世明，武汉工交职业学院计算系主任，副教授

王代萍，湖北大学知行学院计算机系主任，副教授

龙 翔，湖北生物科技职业学院计算机系主任

张传学，湖北开放职业学院理工系主任

杨宏亮，武汉工程职业技术学院计算中心

李守明，中国地质大学(武汉)江城学院电信学院院长，教授

李晓燕，黄冈科技职业学院电子信息工程系主任，教授

李群芳，武汉工程大学职业技术学院计算机系主任，副教授

明志新，湖北水利水电职业学院计算机系主任

章启俊，武汉商贸学院信息工程学院院长，教授

谭琼香，武汉信息传播职业技术学院网络系

执行编委：

黄金文，武汉大学出版社副编审



内容简介

VC++6.0 是面向对象的可视化程序设计语言,是众多系统软件和应用软件的开发工具。本书从 C 语言出发,介绍 C++ 语言的基本知识和 VC++ 编写 Windows 应用程序的方法。内容丰富,讲述清楚,深入浅出,并提供了大量的学习实例,每章均有练习题和上机实验题。在附录中给出了所有练习题和上机实验题的参考答案。本书可作为计算机专业面向对象程序设计教材,也可用做高职高专计算机教材,还可用于了解和掌握 VC++ 程序设计人员的参考书。



近五年来,我国的教育事业快速发展,特别是民办高校、二级分校和高职高专发展之快、规模之大是前所未有的。在这种形势下,针对这类学校的专业培养目标和特点,探索新的教学方法,编写合适的教材成了当前刻不容缓的任务。

民办高校、二级分校和高职高专的目标是面向企业和社会培养多层次的应用型、实用型和技能型的人才,对于计算机专业来说,就要使培养的学生掌握实用技能,具有很强的动手能力以及从事开发和应用的能力。

为了满足这种需要,我们组织多所高校有丰富教学经验的教师联合编写了面向民办高校、二级分校和高职高专学生的计算机系列教材,分本科和专科两个层次。本系列教材的特点是:

1. 兼顾了系统性和先进性。教材既注重了知识的系统性,以便学生能够较系统地掌握一门课程,同时对于专业课,瞄准当前技术发展的动向,力求介绍当前最新的技术,以提高学生所学知识的可用性,在毕业后能够适应最新的开发环境。

2. 理论与实践结合。在阐明基本理论的基础上,注重了训练和实践,使学生学而能用。大部分教材编写了配套的上机和实训教程,阐述了实训方法、步骤,给出了大量的实例和习题,以保证实训和教学的效果,提高学生综合利用所学知识解决实际问题的能力和开发应用的能力。

3. 大部分教材制作了配套的多媒体课件,为教师教学提供了方便。

4. 教材结构合理,内容翔实,力求通俗易懂,重点突出,便于讲解和学习。

诚恳希望读者对本系列教材缺点和不足提出宝贵的意见。

编委会

2005年8月8日



前言

VC++ 6.0 是面向对象的可视化程序设计语言,它将数据和处理数据的过程当成一个整体,具有封装和数据隐藏、继承和重用以及多态性等优点,目前已经成为开发大型软件的主流开发环境。

VC++ 虽然功能强大,但与其他编程语言(VB、VFP、Delphi 等)相比,却不易学习,让很多初学者望而却步。本书从 C 语言入门,介绍 C++ 语言的基本编程方法,继而讨论 Windows 应用程序的设计方法,最后过渡到 MFC 类库的使用。内容由浅入深地逐步展开,力图使初学者容易理解,而不是死记概念。书中精心设计了大量的例题,具体说明有关概念的应用方法。所有例题都在 Microsoft Visual C++ 6.0 系统中运行通过。

本书共分 13 章,第一章到第七章介绍 C/C++ 语言的基础部分,后 6 章介绍 VC++ 6.0 面向对象的可视化程序设计方法。各章的内容布局如下:第一章是 C++ 基础,阐述了面向对象的有关概念;第二章是类和对象,介绍类和对象的设计方法;第三章是友元,包括友元函数和友元类;第四章是运算符重载,介绍了运算符重载的概念和设计方法;第五章是模板,介绍函数模板和类模板的设计方法;第六章是派生和继承,介绍了设计基类和派生类的相关知识;第七章是多态性和虚函数,介绍了设计虚函数和纯虚函数的方法;第八章是 Visual C++ 6.0 开发环境,介绍了 Visual C++ 6.0 系统的使用过程;第九章是编写传统的 Windows 应用程序,介绍了设计 Windows 应用程序所需的基本概念和相关知识;第十章是利用 MFC 类库编写 Windows 应用程序,介绍了 MFC 类库的结构和 MFC 的消息处理机制;第十一章是使用向导 AppWizard 建立应用程序,介绍了 SDI/MDI 和对话框应用程序的框架;第十二章是用 ClassWizard 设计类,介绍了使用 ClassWizard 向导设计类的相关知识;第十三章是常用控件编程,介绍了按钮、编辑框、列表框、组合框、滚动条等一些常用的控件的使用方法。

本书强调学习过程的练习和实习训练。每章均有练习题和上机实验题。在两个附录中给出了所有练习题和上机实验题的参考答案。

本书可以作为高等院校计算机专业和相关专业学生学习 VC++ 语言和面向对象程序设计课程的教材。

如无特别说明,书中 VC++ 指的是 Microsoft Visual C++ 6.0。



本书的编写得到武汉大学出版社的大力支持,在此作者向他们表示深深的谢意。
由于作者水平所限,书中难免存在不当之处,诚恳希望广大读者批评指正。

作 者

2006 年 3 月



目 录

第一章 C++ 基础	1
1.1 C++ 概述	1
1.1.1 C++ 的起源	1
1.1.2 程序和程序设计语言	1
1.1.3 结构化程序设计	2
1.1.4 面向对象的程序设计	3
1.1.5 C++ 语言的特点	4
1.1.6 C++ 程序开发过程	4
1.1.7 C++ 程序示例	5
1.1.8 VC++ 开发控制台应用程序的步骤	7
1.2 C++ 语言初步	13
1.2.1 C++ 基本数据类型	13
1.2.2 常量	14
1.2.3 变量	16
1.2.4 数据的输入和输出	18
1.2.5 构造数据类型	21
1.2.6 指针	26
1.3 控制语句	31
1.3.1 顺序控制语句	31
1.3.2 选择控制语句	32
1.3.3 循环控制语句	36
1.3.4 跳转语句	38
1.4 函数	40
1.4.1 函数的定义和调用	40
1.4.2 函数的参数传递	42
1.4.3 内联函数	46
1.4.4 函数重载	47
1.4.5 域运算符	49



1.5 断言	49
练习题一	50
上机实验题一	51
第二章 类和对象	52
2.1 类	52
2.1.1 类的定义	52
2.1.2 类的成员函数	54
2.1.3 访问权限	54
2.2 类对象	55
2.2.1 对象的定义格式	56
2.2.2 对象成员的表示方法	56
2.3 构造函数和析构函数	58
2.3.1 构造函数	58
2.3.2 重载构造函数	60
2.3.3 析构函数	63
2.4 类成员指针	65
2.4.1 类数据成员指针	65
2.4.2 类成员函数指针	66
2.5 this 指针	67
练习题二	71
上机实验题二	73
第三章 友元	74
3.1 友元函数	74
3.2 友元类	78
练习题三	79
上机实验题三	80
第四章 运算符重载	81
4.1 运算符重载概述	81
4.2 重载单目运算符	83
4.3 重载双目运算符	85
4.4 重载运算符 new 与 delete	87
4.5 重载类型转换运算符	88
练习题四	89



上机实验题四	89
第五章 模板	90
5.1 模板的概念	90
5.2 函数模板	91
5.2.1 函数模板说明	91
5.2.2 使用函数模板	91
5.2.3 用户定义的参数类型	93
5.3 类模板	95
5.3.1 类模板说明	95
5.3.2 使用类模板	97
练习题五	100
上机实验题五	100
第六章 派生和继承	101
6.1 派生类	101
6.1.1 派生类的定义格式	101
6.1.2 派生类生成过程	103
6.2 访问控制	103
6.2.1 公有继承	104
6.2.2 私有继承	105
6.2.3 保护继承	107
6.3 派生类的构造函数和析构函数	109
6.3.1 构造函数	109
6.3.2 析构函数	114
6.4 虚基类	116
6.4.1 作用域分辨符	116
6.4.2 虚基类说明	118
6.4.3 虚基类的初始化	120
练习题六	124
上机实验题六	126
第七章 多态性和虚函数	127
7.1 静态联编和动态联编	127
7.2 虚函数	130
7.2.1 虚函数说明	130



7.2.2 多继承中的虚函数	133
7.2.3 虚函数的限制	134
7.3 纯虚函数和抽象类	137
7.3.1 纯虚函数	137
7.3.2 抽象类	139
练习题七	141
上机实验题七	143
第八章 Visual C++ 6.0 开发环境	144
8.1 Visual C++ 开发环境概述	144
8.2 Visual C++ 6.0 的菜单栏	145
8.2.1 File 菜单	146
8.2.2 Edit 菜单	146
8.2.3 View 菜单	148
8.2.4 Insert 菜单	148
8.2.5 Project 菜单	149
8.2.6 Build 菜单	149
8.2.7 Tools 菜单	150
8.2.8 Windows 菜单	150
8.2.9 Help 菜单	151
8.3 开发环境的工具栏	151
8.3.1 工具栏的构成	151
8.3.2 工具栏的显示与隐藏	152
8.4 工程和工程工作区	154
8.4.1 工程	154
8.4.2 工程工作区	155
8.5 Visual C++ 的资源	157
8.5.1 资源的分类	158
8.5.2 资源标识符	159
8.5.3 资源的基本操作	159
练习题八	162
上机实验题八	162
第九章 编写传统的 Windows 应用程序	163
9.1 Windows 编程基础	163
9.1.1 什么是 API 和 Windows API	163



9.1.2 窗口	164
9.1.3 句柄和 Windows 消息	164
9.1.4 事件驱动模型和消息循环	168
9.2 编写 Windows 应用程序的步骤	169
9.3 Windows 源程序组成结构	173
9.3.1 C++ 头文件	173
9.3.2 WinMain 函数	173
9.3.3 窗口函数	179
9.3.4 数据类型	180
9.3.5 数据结构	181
9.3.6 常用的函数	182
9.4 Windows 程序实例	184
练习题九	189
上机实验题九	189
第十章 利用 MFC 类库编写 Windows 应用程序	191
10.1 MFC 类库概述	191
10.1.1 CObject 类	191
10.1.2 从 CObject 类派生的主要类	191
10.2 MFC 的消息和消息处理	193
10.2.1 MFC 事件驱动程序设计的引擎——CWinApp 类	193
10.2.2 消息映射和消息处理	194
10.2.3 MFC 预定义的消息和消息处理程序	196
10.2.4 使用 MFC 的 Windows 应用程序示例	197
10.3 Windows 的图形设备接口及 Windows 绘图	199
10.3.1 图形设备接口与图形设备设备描述表	199
10.3.2 坐标系统和映射模式	203
10.3.3 设备环境的颜色	208
10.3.4 GDI 对象	211
10.3.5 与绘图相关的简单数据类	217
10.3.6 常用的绘图函数	219
10.3.7 文本输出过程	223
10.3.8 图形绘制过程	225
10.3.9 文本和图形输出的应用程序示例	226
10.4 Windows 应用程序对鼠标的响应	234
10.4.1 鼠标消息和鼠标消息处理	234



10.4.2 响应鼠标事件的 Windows 应用程序示例	236
10.5 Windows 应用程序对键盘的响应	243
10.5.1 键盘消息和键盘消息处理	243
10.5.2 响应键盘事件的 Windows 应用程序示例	245
练习十	251
上机实验题十	251
第十一章 使用向导 AppWizard 建立应用程序	252
11.1 工程类型和文件类型	252
11.2 AppWizard 的使用	254
11.2.1 步骤 1	254
11.2.2 步骤 2	255
11.2.3 步骤 3	258
11.2.4 步骤 4	259
11.2.5 步骤 5	262
11.2.6 步骤 6	263
11.3 创建一个简单的单文档应用程序	265
11.3.1 使用 AppWizard 生成 SDI 应用程序基本框架	265
11.3.2 SDI 应用程序框架剖析	266
11.3.3 添加代码	282
11.3.4 编译执行程序	283
11.4 创建一个简单的多文档应用程序	284
11.4.1 使用 AppWizard 生成 MDI 应用程序的基本框架	284
11.4.2 MDI 应用程序框架剖析	285
11.4.3 添加代码	299
11.4.4 编译执行程序	300
11.5 创建一个简单的对话框应用程序	301
11.5.1 使用 AppWizard 生成对话框应用程序基本框架	301
11.5.2 对话框应用程序框架剖析	302
11.5.3 添加控件和代码	308
11.5.4 编译执行程序	309
练习十一	309
上机实验题十一	309
第十二章 用 ClassWizard 设计类	310
12.1 启动 ClassWizard 向导	310



12.2 使用 ClassWizard 向导	311
12.2.1 Message Maps 选项卡	311
12.2.2 Member Variables 选项卡	312
12.2.3 Class Info 选项卡	313
12.2.4 如何往视图类中加入内存变量	314
12.2.5 如何为对话框中的控件加入内存变量	314
12.2.6 如何往视图类中加入成员函数	316
12.3 应用 ClassWizard 向导的示例	317
练习题十二	324
上机实验题十二	324
 第十三章 常用控件编程	 325
13.1 控件的创建和使用	325
13.1.1 控件的创建	325
13.1.2 Windows 常用控件介绍	327
13.2 Windows 常用控件	330
13.2.1 静态控件	330
13.2.2 按钮	330
13.2.3 编辑框	335
13.2.4 列表框	338
13.2.5 列表视图控件	343
13.2.6 组合框	362
13.2.7 滚动条	367
13.2.8 滑动条	369
13.2.9 微调按钮	371
练习题十三	379
上机实验题十三	379
 附录 A 练习题参考答案	 381
 附录 B 上机实验题参考答案	 392
 参 考 文 献	 421



第一章 C++ 基础



C++ 语言是一种广泛应用的程序设计语言,它在 C 语言的基础上扩展了面向对象的程序设计特点。最主要的是增加了类功能,使它成为面向对象的程序设计语言,从而提高了开发软件的效率。

1.1 C++ 概述

本节介绍 C++ 的起源、程序设计语言、过程程序设计和面向对象程序设计、C++ 语言的特点、C++ 程序开发过程和使用 VC++ 6.0 开发控制台应用程序的步骤等。

1.1.1 C++ 的起源

C++ 源于 C 语言。1972 年至 1973 年期间,D. M. Ritchie 首创了一种新的程序设计语言,取名为 C 语言。设计 C 语言的最初目的是编写操作系统。由于其具有简单、灵活的特点,C 语言很快就被用于编写各种不同类型的程序,从而成为世界上最流行的语言之一。

但是,C 语言是一个面向过程的语言。随着软件开发技术的进步,程序员们最终发现,把数据和施加在数据之上的操作结合起来,会得到更易于理解的程序,由此产生了面向对象的程序设计思想。于是,20 世纪 80 年代初,美国 AT&T 贝尔实验室的 Bjarne Stroustrup 设计并实现了 C 语言的扩充版本,最初称为“带类的 C”,1993 年正式取名为 C++。C++ 改进了 C 的不足之处,支持面向对象的程序设计,在改进的同时保持了 C 的简洁性和高效性。

目前,C++ 越来越受到重视并已得到了广泛的应用,许多软件公司为 C++ 设计编译系统,提供不同应用级别的类库和越来越方便的开发环境,其中较为著名的有 Microsoft 公司的 Visual C++6.0 和 Borland 公司的 Borland C++5.02 等。利用 C++ 设计并实现应用系统变得日益简单和快捷了。

1.1.2 程序和程序设计语言

程序是一个十分广泛的概念。例如当宣布开会时,便启动了会议程序。当打开计算机电源时,便启动了计算机程序。会议程序可用汉语描述,也可以用其他语言描述,总之是用人类自然语言描述的。计算机程序也可以用不同的语言(例如,机器语



言或更为通用的高级程序设计语言等)描述。

计算机程序是计算机处理对象和计算规则的描述。程序设计语言是用来表示计算机事务处理过程。无论是自然语言还是计算机语言,其基础是一组记号和规则,根据规则由记号构成记号串的总体就是语言。

我们知道,人类自然语言(如汉语)是人们交流和表达思想的工具。那么,人与计算机如何“交流”呢?为此,就产生了计算机语言,其功能是人用计算机语言编写一系列的动作,计算机能够“理解”这些动作,按照指定的动作去执行。正是这种相同点,因而计算机语言和自然语言都叫做“语言”。

自然语言由于其历史性和文化性,除了其语法外,还包含复杂的语义和语境,所以,人们能理解很多不完全符合语法的语句。但计算机语言是人发明的,它主要是用语法来表达程序员的思想,因而在编写程序时要严格遵守语法规则。

如同人类有很多自然语言一样,计算机语言也有很多种。按照计算机历史的发展有如下几类:

- **机器语言** 它是面向机器的,是特定计算机系统所固有的语言。用机器语言进行程序设计,需要对机器结构有较多的了解。用机器语言编写的程序可读性很差,程序难以修改和维护。

- **汇编语言** 为了提高程序设计效率,人们考虑用有助记忆的符号来表示机器指令中的操作码和运算数,例如用 ADD 表示加法, SUB 表示减法等。相对机器语言来看,用汇编语言编写程序的难度有所降低,程序的可读性有所提高,但仍与人类的思维相差甚远。

- **高级语言** 汇编语言和计算机的机器语言十分接近,它的书写格式在很大程度上取决于特定计算机的机器指令,这对于人们抽象思维和交流十分不便。高级语言指的是像 Fortran、C、Pascal 和 Basic 等与具体机器无关的语言,这样程序设计者不需要了解机器的内部结构,只要按照计算机语言的语法编写程序即可。

1.1.3 结构化程序设计

出现高级语言之后,如何用它来编写较大的程序呢?人们把程序看成是处理数据的一系列过程。过程或函数定义为一个接一个顺序执行的一组指令。数据与程序分开存储,程序设计的主要技巧在于追踪哪些函数和调用哪些函数及哪些数据发生了变化。为解决其中可能存在的问题,结构化程序设计便应运而生。

传统的程序设计方法可以归结为“程序 = 算法 + 数据结构”,将程序定义为处理数据的一系列过程。这种设计方法的着眼点是面向过程的,特点是数据与程序分离,即数据与数据处理分离。

结构化程序设计的基本思想是采用自顶向下、逐步细化的设计方法和单入口单出口的控制结构。结构化程序设计为处理复杂问题提供了有力手段,但到 20 世纪 80 年代末,这种设计方法逐渐暴露出以下缺陷: