

21世纪大学计算机基础规划教材

Visual C++ 程序设计基础

付灵丽 柴欣 主编

- 采用循序渐进逐步深入的方法对VC的基础、面向对象的程序设计方法及Windows编程进行讲解，使读者易学易懂。
- 大部分章节结合典型实例讲解基本概念和方法，同时还配有大量习题，便于读者练习。
- 语言严谨、流畅，通俗易懂，重点突出，实例丰富。



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪大学计算机基础规划教材

Visual C++ 程序设计基础

付灵丽 柴 欣 主 编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书采用循序渐进的方法,对 VC++的基础、面向对象的程序设计方法及 Windows 编程进行讲解。全书共 9 章,各部分内容相互衔接、前后呼应,讲解由浅入深、循序渐进。为了提高读者的编程技巧,在大部分章节里结合典型实例讲解基本概念和方法,同时还配有大量习题,便于读者练习。同时与本书配套的《Visual C++程序设计实验教程》提供了上机指导及上机习题参考答案。

本书语言表达严谨、流畅,通俗易懂,重点突出,实例丰富,适合作为大专院校各专业程序设计语言课程的正式教材,又可作为研究生计算机基础教育的教材,也比较适合广大计算机爱好者自学和参考。

图书在版编目(CIP)数据

Visual C++程序设计基础/付灵丽,柴欣主编. —北京:中国铁道出版社,2004.8
(21 世纪大学计算机基础规划教材)

ISBN 7-113-06140-0

I. V… II. ①付… ②柴… III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 091631 号

书 名: Visual C++程序设计基础

作 者: 付灵丽 柴 欣

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 戴 薇

责任编辑: 苏 茜 夏华香 秦绪好

封面设计: 薛 为

印 刷: 北京鑫正大印刷有限公司

开 本: 787×1092 1/16 印张: 20 字数: 478 千

版 本: 2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-06140-0/TP·1297

定 价: 28.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

编 委 会

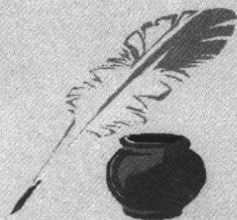
主 编：付灵丽 柴 欣

副主编：史巧硕 张红梅

编 委：（按姓氏汉语拼音顺序）

梁艳红 刘洪普 马 岱

武优西 于 明 朱怀忠



前 言

随着计算机技术的飞速发展, 社会对人才的计算机应用与开发水平的要求也日益提高, 为适应此形势, 高校的计算机基础教学内容也在不断的改革之中。

目前, C 语言已经成为高校理工科学生的必修或选修课, 但随着软件工程技术的不断发展, 面向对象的编程技术已经成为当今软件开发的重要手段之一, 尤其是 Visual C++ 的出现, 大大推进了面向对象与可视化编程技术的应用与发展。因此, 掌握“面向对象与可视化程序设计”的内容和方法已经成为大学生计算机应用与开发能力的要求之一。由于 Visual C++ 功能强大, 使用灵活, 对于初学程序设计的人来说是较困难的, 高校理工科非计算机专业的低年级学生也同样面临这个问题。虽然目前市面上关于 Visual C++ 的书籍很多, 但适合以上读者群体的高教教材并不多见, 鉴于这种情况, 为了使初学程序设计的读者能够掌握 Visual C++ 的程序设计方法, 并初步具备使用 Visual C++ 开发应用程序和解决实际问题的能力, 我们根据近几年的实际教学经验, 确定了本书的结构, 精选了 Visual C++ 的内容。

全书共分为 9 章, 第一部分 1~6 章, 介绍 C++ 的基础部分。在该部分中, 较为系统地讲述了 C++ 语言的基础知识、基本规则及编程方法, 其中第 1 章介绍了程序设计语言的发展, 并通过一个简单实例, 对 Visual C++ 6.0 集成开发环境进行了介绍; 第 2~6 章讲述了 C++ 语言的基本内容, 包括: 常量、变量、运算符和表达式、各种语句、函数和预处理、指针和引用、结构和联合体等, 这些内容也是构成 C++ 程序的基础。第二部分第 7 章重点介绍了 C++ 的面向对象的基本思想及设计方法。在这部分中, 较系统地讲述了 C++ 语言中面向对象的主要特征, 如封装、继承、多态等, 这些都是 C++ 的核心内容, 体现了 C++ 语言面向对象的特点。第 8、9 章重点介绍了可视化编程的基本方法, 在这部分中主要介绍 Windows 编程基础知识、使用 MFC 进行可视化编程的基本方法。

本书的作者长期从事 C++ 语言程序设计课的教学工作, 并曾利用 C++、VC++ 开发了多个软件项目, 因此有着丰富的教学经验和较强的科研能力, 对 Visual C++ 有着较深入的理解。本着加强基础、注重实践、勇于创新、突出应用的原则, 力求使本教材达到可读性、适用性和先进性。为了便于读者自学, 在全书的体系结构和内容上注意了由浅入深、深入浅出、循序渐进的方针。为了提高读者编程技巧, 在大部分章节中都提供了典型例题。不仅适用于教学, 同时也适用于用 Visual C++ 开发应用程序的用户参考。如教师选用本书作为大学生软件技术基础课程的教材时, 可根据实际授课时数取舍本书章节。由于授课时数的限制, 教师可在规定授课时数内重点讲授第一部分的内容, 而在后续的选修课程或研究生课程中介绍第二部分和第三部分的内容, 这样可使学生完整地学习 Visual C++ 的内容。

本书中所给出的程序示例均在 Visual C++ 6.0 下进行过调试和运行。为帮助读者更好地学习 Visual C++, 作者还编写了配套的《Visual C++ 程序设计实验教程》一书, 该书充实了一些编程实例, 并提供实验习题的参考答案。

本书由付灵丽、柴欣主编, 并负责全书的总体策划与统稿、定稿工作, 史巧硕、张红梅任副主编, 各章编写分工如下: 第 1、2 章由付灵丽编写, 第 3、6 章由马岱编写, 第 4 章由史巧硕编写, 第 5 章由柴欣编写, 第 7 章由武优西编写, 第 8、9 章由张红梅编写。参加本书

大纲讨论及部分编写工作的老师还有：于明、刘洪普、朱怀忠、梁艳红等。

在本书的编写过程中一直得到有关专家热心的指导与无私的帮助，编者在此一并表示衷心的感谢。此外在本书写作时，还参考了大量文献资料，在此也向这些文献资料的作者深表感谢。

由于时间仓促和水平所限，书中难免有不当和欠妥之处，敬请各位专家、读者不吝批评指正。

编 者

2004 年 8 月

目 录

第 1 章 绪 论	1
1-1 概述	1
1-1-1 面向对象方法的由来和发展	1
1-1-2 面向对象语言	2
1-1-3 Visual C++的发展	2
1-2 C++程序的基本结构	3
1-2-1 C++程序结构	3
1-2-2 C++程序中的注释	5
1-3 程序的调试与运行	5
1-3-1 Visual C++ 6.0 开发环境	5
1-3-2 开发环境的初步实践	6
第 2 章 数据类型及表达式	11
2-1 词法符号	11
2-1-1 字符集	11
2-1-2 标识符	11
2-1-3 关键字	12
2-2 基本数据类型	12
2-3 常量与变量	14
2-3-1 常量	14
2-3-2 变量	17
2-4 运算符与表达式	19
2-4-1 算术运算	20
2-4-2 关系运算	23
2-4-3 逻辑运算	24
2-4-4 赋值运算	26
2-4-5 逗号运算	27
2-4-6 sizeof 运算符	28
2-5 类型转换	28
2-5-1 自动类型转换	29
2-5-2 强制类型转换	29
习题	30
第 3 章 结构化程序设计	32
3-1 C++基本语句	32
3-2 数据的输入与输出	33

3-2-1	数据的输出 cout	33
3-2-2	数据的输入 cin	36
3-3	结构化程序设计方法概述	37
3-3-1	程序设计的发展历程	37
3-3-2	结构化程序设计概念	38
3-4	结构化程序设计方法	40
3-4-1	逐步细化法	40
3-4-2	模块法	40
3-5	顺序结构程序设计	41
3-6	选择结构程序设计	42
3-6-1	用 if 语句实现选择结构	42
3-6-2	if 语句的嵌套	47
3-6-3	条件表达式	48
3-6-4	switch 语句	50
3-7	循环结构的实现	55
3-7-1	while 语句	55
3-7-2	do-while 语句	57
3-7-3	for 语句	59
3-7-4	循环的嵌套	63
3-7-5	break 语句	65
3-7-6	continue 语句	67
3-7-7	几种循环的比较	68
3-8	程序设计举例	68
习题		71
第 4 章	数组与指针	72
4-1	数组	72
4-1-1	一维数组	72
4-1-2	二维数组	78
4-1-3	字符数组	83
4-2	指针	92
4-2-1	指针的概念	92
4-2-2	指针变量的定义	94
4-2-3	指针变量的初始化	95
4-2-4	指针的运算	98
4-3	指针与数组	101
4-3-1	指向数组的指针	102
4-3-2	通过指针变量使用数组元素	103
4-3-3	指针与字符串	106

4-3-4	多级指针与指针数组	109
4-3-5	数组指针	120
4-4	引用	121
4-4-1	引用及其声明	121
4-4-2	引用的使用	121
4-5	内存管理	123
	习题	126
第 5 章	函数与预处理	127
5-1	函数的定义	127
5-1-1	函数概念的引入	127
5-1-2	函数的定义	128
5-1-3	空函数	130
5-1-4	return 语句	130
5-1-5	函数原型	132
5-2	函数的调用	133
5-2-1	函数的调用	133
5-2-2	参数传递机制	134
5-3	指针与函数	137
5-3-1	指针作为函数参数	137
5-3-2	函数调用中数组的传递	140
5-3-3	函数指针	143
5-3-4	指针函数	146
5-4	函数的嵌套调用	147
5-5	函数的递归调用	150
5-6	内联函数和重载函数	153
5-6-1	内联函数	153
5-6-2	重载函数	155
5-7	默认参数的函数	157
5-7-1	默认参数的函数	157
5-7-2	使用默认参数的函数时的注意问题	157
5-8	作用域与生命期	158
5-8-1	作用域	158
5-8-2	全局变量和局部变量	161
5-8-3	生命期	164
5-9	编译预处理	169
5-9-1	宏定义 (Macro)	169
5-9-2	文件包含 (#include)	173
5-9-3	条件编译	175

习题.....	182
第6章 复杂数据类型	184
6-1 结构体类型.....	184
6-1-1 结构体类型的定义.....	184
6-1-2 结构体类型变量的定义.....	185
6-1-3 结构体变量的初始化.....	187
6-1-4 结构体成员的访问.....	187
6-1-5 结构体数组.....	189
6-1-6 结构体指针.....	192
6-1-7 结构体指针的应用.....	194
6-2 联合体.....	201
6-2-1 联合体的定义.....	201
6-2-2 访问联合体的成员.....	202
6-2-3 联合体类型的特点.....	202
6-3 枚举类型.....	205
6-3-1 枚举类型及枚举变量的定义.....	205
6-3-2 枚举元素的访问.....	206
6-4 用 typedef 定义类型.....	206
习题.....	207
第7章 类与对象.....	208
7-1 类与对象.....	208
7-1-1 类的定义.....	208
7-1-2 对象——类的实例.....	212
7-1-3 构造函数与析构函数.....	213
7-1-4 类与对象的进一步讨论.....	217
7-2 友元与运算符重载.....	224
7-2-1 友元.....	224
7-2-2 运算符重载.....	226
7-3 继承与派生.....	228
7-3-1 派生类.....	229
7-3-2 派生类对基类成员的覆盖.....	232
7-3-3 派生类的构造函数和析构函数.....	233
7-4 虚函数.....	236
7-4-1 多态性.....	236
7-4-2 虚函数.....	238
7-4-3 抽象基类.....	240
习题.....	241
第8章 对话框和基本控件	243

8-1	Windows 编程与 MFC.....	243
8-1-1	Windows 编程.....	243
8-1-2	MFC.....	244
8-1-3	Visual C++	245
8-1-4	使用 MFC AppWizard 创建基于对话框的应用程序.....	245
8-2	对话框与常用控件概述	251
8-2-1	对话框.....	251
8-2-2	常用控件.....	251
8-2-3	创建与用户交互的基于对话框的 MFC 应用程序.....	253
8-3	添加对话框.....	263
8-3-1	显示 About 对话框	263
8-3-2	多对话框应用程序	264
8-3-3	其他控件使用举例	271
	习题.....	281
第 9 章	菜单与文档视图结构	282
9-1	菜单.....	282
9-1-1	菜单概述.....	282
9-1-2	对话框应用程序中的菜单设计实例.....	285
9-2	文档与视图结构及其应用	289
9-2-1	概述	289
9-2-2	文档类及其派生类	290
9-2-3	视图类及其派生类	292
9-2-4	建立单文档界面的绘画程序实例	294
	习题.....	304

第 1 章

绪 论

面向对象设计技术已成为当今一种流行的软件设计技术。随着 C++ 语言的广泛应用,更推进了面向对象技术的发展。

1-1 概述

1-1-1 面向对象方法的由来和发展

计算机科学发展的每一步几乎都是在软件设计和程序设计语言中得到体现的。软件设计是一个发展的概念,随着软件开发规模的不断扩大和开发方式的变化,程序设计开始被人们作为一门科学来对待,经过多年研究,在计算机科学中发展了许多程序设计的方法。

下面我们通过回顾计算机语言的发展过程,来了解一下面向对象的方法是如何产生的。

20 世纪 50 年代的程序设计都是用机器语言或汇编语言编写的。这种程序的设计相当麻烦,严重影响了计算机的普及应用。随着计算机的应用日益广泛,发展了一系列不同风格的、为不同对象服务的程序设计语言。

最早的高级语言是在 20 世纪 50 年代中期研制的 FORTRAN 语言,它在计算机语言发展史上具有划时代的意义。该语言引进了许多现在仍然使用的程序设计概念,如变量、数组、分支、循环等。但在使用中也发现了一些不足,不同部分的相同变量名容易发生错误。50 年代后期,高级语言 Algol 在程序段内部对变量实施隔离。Algol60 提出了块结构的思想,实际上也是一种初级的封装。在 60 年代开发的 Simula 67,是面向对象的鼻祖。它将 Algol60 块结构的概念向前推进了一步,提出了对象的概念。70 年代出现的 Ada 语言是一种基于对象的语言,是支持数据抽象类型的最重要的语言之一,但它不是全面的支持继承。后来出现的 Smalltalk 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。到 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。

由于自 60 年代末到 70 年代初,出现了大型的软件系统,如操作系统、数据库等,因此给程序设计带来了新的问题。大型软件系统的研制需要花费大量的人力和物力,但编写出来的软件可靠性差,错误多,难以维护,已经到了程序员无法控制的地步,这就是“软件危机”。

1969 年, E. W. Dijkstra 首先提出了结构化程序设计的概念,他强调了从程序结构和风格上研究程序设计,为“软件危机”起了很大的缓解作用。到 20 世纪 70 年代末结构化设计方法得到了很大的发展, Niklaus Wirth 又提出了“算法+数据结构 = 程序设计”的程序设计方法,将软件划分成若干个可单独命名和编址的部分,它们被称为模块,模块化使软件能够有效地被管理和维护,能够有效地分解和处理复杂问题。

由于软件开发是对问题的求解过程,从认识论角度看,软件开发过程包括人们对要解决问题及相关事物的认识和基于认识所进行的描述。而结构化设计方法不能直接反映出人类认识问题的过程。随着计算机软件的发展,软件系统越来越复杂庞大,结构化程序设计方法已显力不从心。

20 世纪 80 年代,面向对象的程序设计语言日趋成熟,在软件开发中各种概念和方法积累的基础上,就如何超越程序的复杂性障碍,如何在计算机系统中自然地表示客观世界等问题,人们提出了面向对象的设计方法,它不是以过程为中心,而是以对象代表的问题为中心环节,提出了“对象+对象+……=程序设计”理论,使人们对复杂系统的认识过程与系统的程序设计实现过程,尽可能地一致。面向对象的程序设计方法的出现使“软件危机”得到很好地解决。

1-1-2 面向对象语言

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。归纳起来大致分两类:一类是纯面向对象的语言,如 Smalltalk 和 Eiffel;另一类是混合型的面向对象语言,如 C++和 Objective C。

C++语言是由 AT&T 公司的贝尔实验室的 Bjarne Stoustrup 博士开发的,它的创作灵感来源于计算机语言多方面成果的凝聚,特别是 BCPL(Basic Combined Programming Language, C 语言的基础)和 Simula 67(以面向对象为核心的语言),同时也借鉴了 Algol68 语言。C++的名字是由 Rick Masenirti 提出,到 1983 年确定。

C++是一门高效实用的混合型程序设计语言,它最初的设计目标是:支持面向对象编程技术;支持抽象形态的类;更好的 C 语言。C++语言包括两部分:一是 C++基础部分,它是以 C 语言为核心的;另一部分是 C++面向对象特性部分,是 C++对 C 语言的扩充部分。这样它既支持面向对象程序设计方法,又支持结构化程序设计方法,同时广泛的应用基础和丰富的开发环境的支持,也使面向对象设计得到很快普及。

C++的基础部分与 C 语言相比除了一些细微的差别外,C++可以说是 C 语言的加强版,它保留了 C 语言功能强、效率高、风格简洁、适合于大多数的系统程序设计任务等优点,使得 C++与 C 之间取得了兼容性,因此,在过去的软件开发中积累的大量的 C 的库函数和实用程序都可在 C++中应用。

另外,C++语言通过对 C 的扩充,克服了原有 C 语言的缺点,完全支持面向对象程序设计方法。它支持类的概念。类是一种封装数据和对这些数据进行操作的函数的用户定义的类型,它使抽象得以描述。类还提供了数据隐蔽,确保了程序的稳定性、可靠性和可维护性。它还支持继承、派生和多态性等层次结构,使得其代码具有高度的可重用性。

1-1-3 Visual C++的发展

随着 C++逐渐成为 ANSI 标准,这种新的面向对象程序设计语言迅速成为程序员最广泛使用的工具。众多 C++语言的开发环境也随之不断地推出,竞争十分激烈。1986 年 Borland 公司开发了 Turbo C++ 程序语言,而后又推出了 Borland C++版本。Microsoft 公司于 80 年代中期在 Microsoft C 6.0 的基础上开发了 Microsoft C/C++ 7.0,同时引进了 Microsoft Foundation Class (MFC)库 1.0 版本,完善了源代码。以前这些版本都是依赖于 DOS 环境,或在 Windows

下的 DOS 模式里运行。不久 Microsoft 公司推出的 Microsoft C/C++ 8.0, 即 Visual C++ 1.0 版本, 它是 Microsoft 公司推出的第一个真正的基于 Windows 环境下的可视化的集成开发环境, 将编辑、编译、链接和执行集成为一体。从 Visual C++1.5 版本以后, Microsoft 公司决定不再将更多的努力花的支持 16 位编程上, 虽然 Visual C++2.0 仍提供对 16 位的支持, 但从 2.0 版本以后, Visual C++更多地用来创建 32 位程序。在版本上, Microsoft 公司没有推出 3.0 版本, 版本号直接从 2.0 跳到 4.0, 这样, Visual C++与 MFC 的版本号取得一致。由于 Internet 的流行明显地影响了产品的设计, 在 4.0 版本中, Visual C++引进了为 Internet 编程而设计的新类库。5.0 版本也增加了一些新类, 但注意力更多地集中在改善产品的界面上, 以提供一个更好的在线帮助系统、更高级的宏能力和对开发者组内进行类和其他代码共享的支持。6.0 版本在功能上做了进一步的改进。Visual C++经历了从 1.0 到 6.0 等版本的发展, 软件系统逐渐庞大, 功能日益完善。

1-2 C++程序的基本结构

1-2-1 C++程序结构

了解和掌握程序的结构是编写程序的基础, 就象盖房子的地基一样。一般来说, 一个 C++ 程序的基本框架结构包含了声明区、主函数区和函数定义区三大部分。下面通过一个简单的例子认识 C++程序。

例 1.1 一个简单的 C++程序。

```
#include <iostream.h>
int add(int ,int);
void main()
{
    int a,b;
    a=5;
    b=7;
    int c=add(a,b);           //调用 add 函数
    cout << "a+b="<<c<<endl; //输出 a+b 的值
}
int add(int x ,int y)
{
    int z= x+y;
    return z;
}
```

运行该程序, 在屏幕上出现: a+b=12

1. 声明区

声明区出现在程序文件的所有函数的外部, 它所包含的内容如下, 但并不是每一个程序都需要, 要视问题的不同而变化。一般有以下几种情况:

- (1) 包含头文件: 如例 1.1 中 `#include <iostream.h>`
- (2) 函数声明: 如例 1.1 中 `int add(int ,int);`
- (3) 宏定义: 如 `#define PI 3.1415926`

- (4) 全局变量声明;
- (5) 类定义: 如 `class name{.....};`
- (6) 结构体定义: 如 `struct record{.....};`
- (7) 条件编译: 如 `#ifdef.....` 等

为了使程序的结构清晰,一般将类型及类的定义、函数的说明等放在一个源代码文件中,成为头文件(扩展名为.h或.H)。头文件可以由系统提供,用户可直接使用。如程序中包含的 `iostream.h` 文件是 C++ 系统提供的定义了标准输入、输出流操作的头文件;也可以由用户根据程序需要自己编写头文件。

2. 主函数区

主函数是以 `main()` 开始,是整个程序运行的入口,该函数中可能包含以下几个方面的内容:

- (1) 局部变量的声明: `int a,b;`
- (2) 函数调用: 如 `int c=add(a,b);`
- (3) 一般运算: 如 `a=5;`
- (4) 结构控制: 如 `if(a>b)c=a;`
- (5) 对象与结构的处理;
- (6) 系统函数调用等

3. 函数定义区

程序中除了 `main()` 函数外,还可以包含其他函数,每个函数都有一个不同的函数名称,以供主函数或其他函数调用。每个函数都是由函数说明和函数体组成的。如例 1.1 中的 `add()` 函数。

(1) 函数的说明部分

函数的说明部分包括函数返回值类型、函数名、函数的形式参数。

例如,在例 1.1 中的 `add` 函数的说明部分:

```
int add (int x,int y)
```



函数返回值类型 函数名 参数类型和参数名

函数有返回值时应指明返回值的类型;函数名后必须有一对圆括号 `()`;函数的参数可有也可没有,没有参数的函数为无参函数,有参数的函数称为有参函数,若有多个参数时,参数之间用逗号隔开。

(2) 函数体部分

函数体是用一对花括号 `{}` 括起来的用于完成某种功能的语句的集合。函数体一般包括变量定义和执行语句。在 C++ 语言中,一个变量必须在使用之前进行定义,但变量的定义可出现在第一次使用之前的任意位置。

例如:

```
int add(int x ,int y)
{
    int z;
    z= x+y;          //变量 z 第一次使用前定义
    return z;
}
```


每一个语句（包括变量的定义）的最后都必须有一个分号，分号是 C++ 语言中不可缺少的部分，它表示一条语句的结束。若语句后缺少了分号，程序在编译时会报错。

函数体可以是空的，则该函数为空函数。

例如：

```
int m_function()
{ }
```

空函数不完成任何功能，一般是程序员为以后对系统的开发及功能的完善预留的。

1-2-2 C++程序中的注释

注释是程序员对程序语句所做的说明，是提高程序的可读性的一种手段。C++ 的注释可用 “//” 两个符号注释从该符号开始到该行的结束。也可用一对 “/*” 和 “*/” 符号成对使用，用于注释从 “/*” 开始到 “*/” 结束的任意多行。在生成执行程序时，C++ 忽略注释，并把每一个注释都看作是空格。

1-3 程序的调试与运行

1-3-1 Visual C++ 6.0 开发环境

Visual C++ 6.0 是 Microsoft 公司在 1998 年推出的基于 Windows 9X 和 Windows NT 的优秀集成开发环境。该开发环境为用户提供了良好的可视化编程环境，程序员可以利用该开发环境轻松地使用 C++ 源代码编辑器、资源编辑器和使用内部调试器，并且可以创建项目文件。Visual C++ 6.0 不仅包含编译器，而且它还包含了许多有用组件，通过这些组件的协同工作，可以在 Visual C++ 6.0 集成环境中轻松地完成创建源文件、编辑资源，以及对程序的编译、连接和调试等各项工作。

在 Visual C++ 6.0 中，任何程序的开发过程均遵循图 1-1 所示的开发流程，任何一步有错误或结果不满意均需回到前一步进行修改，再按顺序进行，直到所求结果满意为止。

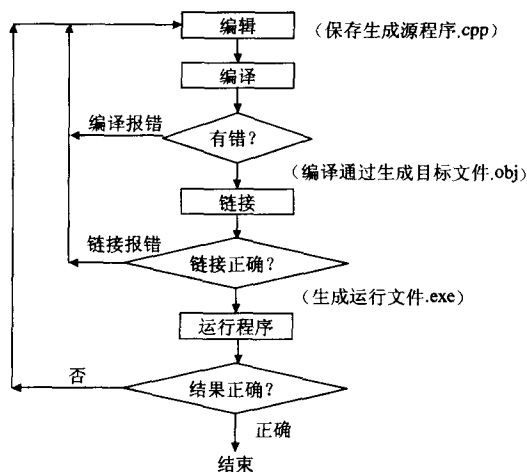


图 1-1 调试与运行程序流程图

1-3-2 开发环境的初步实践

熟悉编程环境的最好方法是编制一个简单的应用程序。下面我们利用 AppWizard（应用程序向导）创建一个基于 DOS 平台项目文件（控制台应用程序）为例，说明应用程序创建的全过程。有关开发环境及其他 Windows 应用程序的创建将在配套的实验指导教材相应的章节中做详细的介绍。

1. 启动 Visual C++6.0

成功地安装了 Visual C++以后，可以在“开始”菜单中的“程序”选项中选择“Microsoft Visual Studio 6.0”级联菜单下的“Microsoft Visual C++ 6.0”命令，启动 Visual C++，进入 Visual C++ 6.0 的集成环境，如图 1-2 所示。

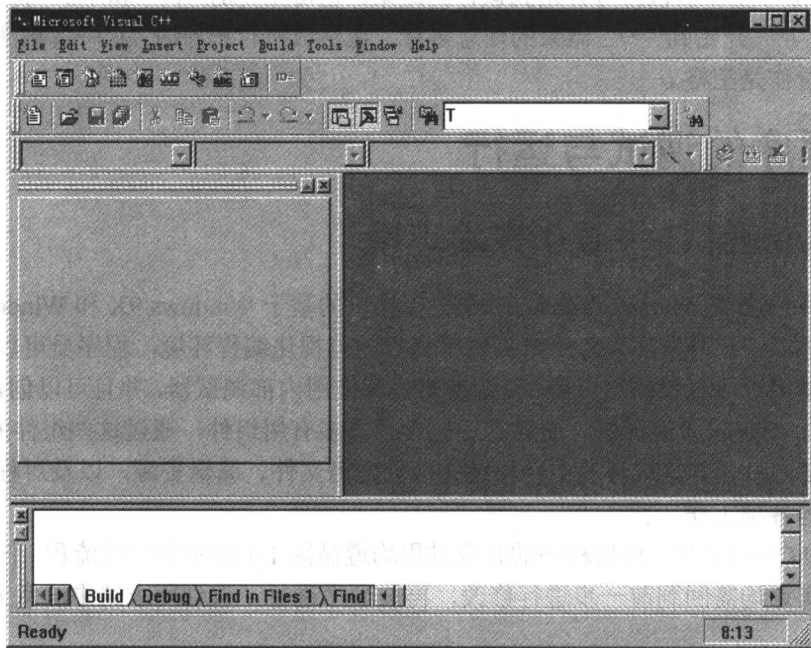


图 1-2 Visual C++界面

2. 创建项目

若开始一个新程序的开发，必须先用 AppWizard（应用程序向导）建立新工程项目。

(1) 建立新工程项目

在 File 菜单下，选择 New 命令，弹出“New”对话框的“Project”标签，如图 1-3 所示。Visual C++可为用户创建用于多种目的的项目，如创建 DOS 平台及 Windows 平台下的项目文件；创建数据库项目、动态链接文件等，如在“Project”标签下选择“Win32 Console Application”项，可创建一个基于 DOS 平台的项目文件；在“Location”编辑栏中选择该工程项目所存放的位置；在“Project name”编辑栏中输入该项目名。单击“OK”按钮，弹出创建 Win32 Console Application 项目步骤一对话框，如图 1-4 所示。

该对话框提供了四种项目的类型，选择不同的选项，意味着系统会自动生成一些程序代码，为项目增加相应的功能。如选择“An empty project”选项，则生成一个空白的项目，单