



高等学校“十二五”规划教材·计算机软件工程系列

Object Oriented Programming (VC++)

面向对象程序设计

(VC++)

主编 李志聪
副主编 富宇 王春英



精讲基础理论 / 注重课程实践
内容通俗易懂 / 案例贯穿始终



哈尔滨工业大学出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

”规划教材·计算机软件工程系列

面向对象程序设计(VC++)

主 编 李志聪

副主编 富 宇 王春英

哈尔滨工业大学出版社

内容简介

本书是一本以 Visual C++6.0 为蓝本,讲述有关面向对象设计思想的教材。本书的最大特色是淡化了很多理论的讲解,通过实际案例的应用,来解释面向对象程序设计的思想和方法,使读者逐步掌握面向对象的相关技术。每一章,首先提出这一章要掌握的主要内容,便于读者清楚每一章的重点内容。C++语法格式的讲解与实例相结合,并在每一章的后面配有相应的练习题与上机练习题,便于读者更好地理解和掌握面向对象的基础知识。在有关章节讲解之后,配有相应的小项目实训,使读者更容易理解面向对象程序设计的思想在解决实际问题中的运用。

本书可以作为软件工程专业及相关专业的教材,对于使用面向对象技术的软件开发人员,也可以作为参考书。

图书在版编目(CIP)数据

面向对象程序设计:VC++/李志聪主编. --哈尔滨:
哈尔滨工业大学出版社,2012.1

ISBN 978 - 7 - 5603 - 3450 - 9

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(2011)第 258537 号

策划编辑 王桂芝 赵文斌
责任编辑 王桂芝 段余男
出版发行 哈尔滨工业大学出版社
社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006
传 真 0451 - 86414749
网 址 <http://hitpress.hit.edu.cn>
印 刷 哈尔滨市石桥印务有限公司
开 本 787mm×1092mm 1/16 印张 16.75 字数 400 千字
版 次 2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷
书 号 ISBN 978 - 7 - 5603 - 3450 - 9
定 价 32.00 元

(如因印装质量问题影响阅读,我社负责调换)

高等学校“十二五”规划教材

计 算 机 软 件 工 程 系 列

编 审 委 员 会

名誉主任 丁哲学

主 任 王义和

副 主 任 王建华

编 委 (按姓氏笔画排序)

王霓虹	印桂生	许少华	任向民
衣治安	刘胜辉	苏中滨	张 伟
苏建民	李金宝	苏晓东	张淑丽
沈维政	金 英	胡 文	姜守旭
贾宗福	黄虎杰	董宇欣	

◎ 序

Foreword

随着计算机工程的发展和社会对计算机工程人才需求的增长,软件工程专业培养目标更加明确,特色更加突出。目前,国内多数高校软件工程专业培养目标是以需求为导向,注重培养学生掌握软件工程基本理论、专业知识和基本技能,具备运用先进的工程化方法、技术和工具从事软件系统分析、设计、开发、维护和管理等工作能力,以及具备参与工程项目的实践能力、团队协作能力、技术创新能力和市场开拓能力,具有发展成软件行业高层次工程技术和企业管理人才的潜力,使学生成为适应社会主义市场经济和信息产业发展需要的“工程实用型”人才。

本系列教材针对软件工程专业“突出学生的软件开发能力和软件工程素质,培养从事软件项目开发和管理的先进工程技术人才”的培养目标,集9家软件学院(软件工程专业)的优秀作者和强势课程,本着“立足基础,注重实践应用;科学统筹,突出创新特色”的原则,精心策划编写。具体特色如下:

1. 紧密结合企业需求,多校优秀作者联合编写

本系列教材编写在充分进行企业需求、学生需要、教师授课方便等多方市场调研的基础上,采取了校企适度联合编写的做法,根据目前企业的普遍需要,结合在校学生的实际学习情况,校企作者共同研讨、确定课程的安排和相关教材内容,力求使学生在校学习过程中就能熟悉和掌握科学研究及工程实践中需要的理论知识和实践技能,以便适应就业及创业的需要,满足国家软件工程专业人才培养需要。

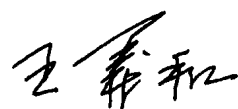
2. 多门课程系统规划,注重培养学生工程素质

本系列教材精心策划,从计算机基础课程→软件工程基础与主干课程→设计与实践课程,系统规划,统一编写。既考虑到每门课程的相对独立性、基础知识的完整性,又兼顾到相关课程之间的横向联系,避免知识点的简单重复,力求形成科学、完整的知识体系。

本系列教材中的《离散数学》、《数据库系统原理》、《算法设计与分析》等基础教材在引入概念和理论时,尽量使其贴近社会现实及软件工程等学科的技术和应用,力图将基本知识与软件工程学科的实际问题结合起来,在具备直观性的同时强调启发性,让学生理解所学的知识。

《软件工程导论》、《软件体系结构》、《软件质量保证与测试技术》、《软件项目管理》等软件工程主干课程以《软件工程导论》为线索,各课程间相辅相成,互相照应,系统地介绍了软件工程的整个学习过程。《数据结构应用设计》、《编译原理设计与实践》、《操作系统设计与实践》、《数据库系统设计与实践》等实践类教材以实验为主题,坚持理论内容以必需和够用为度,实验内容以新颖、实用为原则编写。通过一系列实验,培养学生的探究、分析问题的能力,激发学生的学习兴趣,充分调动学生的非智力因素,提高学生的实践能力。

相信本系列教材的出版,对于培养软件工程人才、推动我国计算机软件工程事业的发展将会起到积极作用。



2011 年 7 月

◎ 前言

Preface

作为软件开发人员来讲,如何将实际的应用问题利用计算机很好地解决,一直是软件开发人员追求的目标。这取决于软件开发人员对实际问题的认知程度,取决于软件开发人员如何将问题解映射成为计算机软件的能力。随着计算机技术的发展,所涉及的应用领域越来越广泛,软件也越来越复杂。如何能更好地解决实际的问题,软件开发人员的程序设计思想起着决定作用。面向对象程序设计思想是 20 世纪 90 年代软件开发方法的主流程序设计思想,很容易将实际要解决的问题映射成为计算机软件,因此,面向程序设计思想是软件开发人员必须掌握的。在众多的面向对象设计语言中,C++受到了广泛的关注。

C++语言本身的目标是将优秀的 C 语言和面向对象理论整合在一起,提供给开发软件的程序员使用,C++是一个更好用的 C。C 语言开发的运行库函数可以不加改动地在 C++语言环境中使用,达到了软件省时省力的重用。另外,C++支持面向对象的程序设计,使用 C++编程,编程效率高;由于 C++引入了面向对象技术和方法解决问题,因此 C++对于问题更容易描述,程序更容易理解与维护;C++的模板对库代码的重用提供了支持;C++更有利于大型程序设计。C++支持面向对象的程序设计方法,特别适合于中型和大型的软件开发项目,从开发时间、费用到软件的重用性、可扩充性、可维护性和可靠性等方面,C++均具有很大的优越性。

本书以 Microsoft 公司开发的 Visual C++ 6.0 为蓝本,讲述基本的面向对象的程序设计思想。本书是软件工程专业系列教材,定位于有一定 C 语言基础的初学者,对于有经验的教师、科技工作者本书也具有参考价值。本书适合作为大学本科(或大专)计算机专业和其他理工科专业 C++程序设计教材。通过本书的学习可以达到如下的目标:

1. 重点掌握 C++语言的语法,理解和掌握面向对象程序设计的思想与方法。
2. 能够灵活地运用 C++语言提供的语法手段解决简单的实际问题。
3. 全方位掌握面向对象编程技术。

本书融入作者多年教学经验的积累,并在实践中不断完善与创新,形成了特有的体系。内容由浅入深,重点、难点分明,从 VC++实际开发中工程文件的组织方法来讲解书中的各个实例,并配有相关的上机实习题,初学者很容易上手。书中各章均有小结、练习题与上机实习题。书中所有示例均在 Microsoft Visual C++ 6.0 环境中运行通过,为使读者阅读方便,特在重点、难点处加以注释。

本书由哈尔滨师范大学李志聪、东北石油大学富宇、哈尔滨理工大学王春英共同编写。其中李志聪任主编,富宇、王春英任副主编。具体编写分工如下:

第 1~3 章及第 12 章由哈尔滨师范大学李志聪编写,第 4~7 章由哈尔滨理工大学王春英编写,第 8~11 章由东北石油大学富宇编写,最后由李志聪统稿。作者水平有限,本书不足之处,敬请读者指正。

编 者

2011 年 8 月

◎ 目 录

Contents

第1章 Visual C++概述	1
1.1 面向对象程序设计思想	1
1.2 C++语言的产生和特点	3
1.3 Visual C++ 6.0 开发环境概述	4
小 结	9
练习题	9
上机实习题	9
第2章 从C到C++	14
2.1 C++的关键字与标识符	14
2.2 C++的数据类型	14
2.3 C++的变量与常量	16
2.4 C++的运算符与优先级	17
2.5 数据的输入和输出	18
小 结	20
练习题	20
上机实习题	21
第3章 函 数	23
3.1 函数的定义和使用	23
3.2 内联函数	24
3.3 带缺省形参值的函数	24
3.4 引 用	26
3.5 函数的重载	30
3.6 函数的模板	31
3.7 实 例	33
小 结	37
练习题	37
上机实习题	38
第4章 类与对象	40
4.1 类和对象	40
4.2 构造函数和析构函数	49
4.3 类的组合	56

4.4	this 指针	60
4.5	类模板	61
4.6	程序实例——人员信息管理程序	64
	小 结	67
	练习题	67
	上机实习题	70
第 5 章	数组与指针	73
5.1	数 组	73
5.2	指 针	76
5.3	动态内存分配	78
5.4	string 类	81
5.5	程序实例——人员信息管理程序	84
	小 结	86
	练习题	86
	上机实习题	88
第 6 章	C++ 程序的结构	89
6.1	标识符的作用域与可见性	89
6.2	对象的生存期	92
6.3	类的静态成员	94
6.4	类的友元	98
6.5	共享数据的保护	101
6.6	多文件结构	105
6.7	程序实例——人员信息管理程序	106
	小 结	108
	练习题	108
	上机实习题	109
第 7 章	继承与派生	112
7.1	类的继承与派生	112
7.2	访问控制	113
7.3	派生类的构造和析构函数	118
7.4	派生类的成员标识与访问	121
7.5	类型兼容规则	126
7.6	程序实例——人员信息管理程序	127
	小 结	133
	练习题	134
	上机实习题	136
第 8 章	多态性	138
8.1	多态性概述	138

8.2 静态联编与动态联编	139
8.3 纯虚函数与抽象基类	142
8.4 运算符重载	149
小 结	158
练习题	158
上机实习题	163
第9章 流类库与输入输出	164
9.1 I/O 流的概念	164
9.2 标准输入输出流	166
9.3 文件流	171
小 结	178
练习题	179
上机实习题	179
第10章 异常处理	180
10.1 异常处理的概述	180
10.2 异常处理的实现	181
10.3 异常处理中的构造与析构	183
10.4 异常处理的嵌套	185
10.5 实 例	187
小 结	189
练习题	190
上机实习题	192
第11章 MFC 库与 Windows 程序开发概述	193
11.1 Windows 编程基本概念	193
11.2 利用 MFC 编写 Windows 程序	194
11.3 Windows 应用程序开发实例	213
小 结	225
上机实习题	225
第12章 学生成绩管理系统实例	226
12.1 项目需求分析	226
12.2 系统的设计	227
12.3 系统的实现	230
小 结	255
参考文献	256

第 1 章

Visual C++ 概述

学习目标:了解面向对象程序设计思想的产生;了解面向对象的基本概念;了解 C++ 语言的特点;掌握利用 Visual C++6.0 创建工程文件的操作过程,编辑和运行工程的操作方法。

1.1 面向对象程序设计思想

1.1.1 面向对象程序设计思想产生的背景

C 程序设计采用的是结构化设计思想的语言,然而随着计算机技术的不断发展和实际的软件开发要解决的问题,结构化思想出现了明显的问题,主要体现在软件重用、软件可维护性和用户动态需求上。

1. 软件重用性

软件重用性是指同一事物不经修改或稍加修改就可多次重复使用的性质,这是软件工程追求的目标之一。

2. 软件可维护性

软件工程强调软件的可维护性,强调文档资料的重要性,规定最终的软件产品应该由完整、一致的配置成分组成。在软件开发过程中,始终强调软件的可读性、可修改性和可测试性,这是软件重要的质量指标。

3. 用户动态需求

用户需求是软件开发的重要前提。尽管在软件开发的初始阶段,都要为用户进行详细的需求分析,然而实际开发过程中,用户的需求是动态变化的。用户的需求产生变化的原因很多,如:用户不清楚软件的具体需求;用户的描述可能不精确,可能有二义性等。

用结构化方法开发的软件,其可重用性和可维护性都比较差,这是因为结构化方法的本质是功能分解,从代表目标系统整体功能的单个处理着手,自顶向下不断把复杂的处理分解为子处理,这样一层一层地分解下去,直到只剩下若干个容易实现的子处理功能为止,然后用相应的工具来描述各个最低层的处理。因此,结构化方法是围绕实现处理功能的“过程”来构造系统的。在实际的软件开发过程中,用户的需求是经常改变的,而用户需求的变化大部分是针对功能的,因此这种变化对于基于过程的设计来说是灾难性的。用这种方法设计出来的系统结构常常是不稳定的,用户需求的变化往往造成系统结构的较大变化,从而需要花费很大代价才能实现这种变化。

因此,设计人员便寻求通过一种新程序设计思想来解决上述问题,面对对象的程序设计

(Object Oriented Programming)思想便诞生了。与传统的面向结构化程序设计思想不同,面向对象的程序设计和问题的求解更符合人们日常自然的思维习惯,因而面向对象程序设计方法不断被软件开发人员接受。

面向对象的程序设计最早起源于20世纪60年代末挪威的K. Nyguard等人推出的编程语言Simula 67,在该语言中引入了数据抽象和类的概念。但真正为面向对象程序设计奠定基础的是由Alan Keyz主持推出的Smalltalk语言,1981年由Xerox Learning Research Group所研制的Smalltalk-80系统,全面地体现了面向对象程序设计语言的特征。

1.1.2 面向对象的基本概念

1. 对象(Object)

面向对象程序设计是一种围绕真实世界的概念来组织模型的程序设计方法,它采用对象来描述问题空间的实体。对象是人們要进行研究的任何事物,从最简单的整数到复杂的飞机等均可看作对象,它不仅能表示具体的事物,还能表示抽象的规则、计划或事件。对象具有状态,一个对象用数据值来描述它的状态。对象还有操作,用于改变对象的状态,操作就是对象的行为。对象实现了数据和操作的结合,使数据和操作封装于对象的统一体中。我们利用计算机解决的实际问题,可以看做是一个或很多个对象,按一定的规则组合在一起,利用它们可以使用的行为及特定的属性,共同完成系统的各项功能。

例如,学生成绩管理系统中,我们可以认为,学生甲、学生乙、教师甲、管理员甲,数据结构课程、C++课程等都是对象。学生甲有姓名为“张三”、学号为“2010000001”,数据结构的成绩为89等的属性,当然也可以有学生的身高、体重等,但身高和体重属性是与学生成绩管理系统无关的属性,可以不考虑。学生甲有查询学生成绩的操作,教师有录入、编辑学生成绩的操作。

2. 类(Class)

具有相同或相似性质的对象的抽象就是类。因此,对象的抽象是类,类的具体化就是对象,也可以说类的实例是对象。

类具有属性,它是对象的状态的抽象,用数据结构来描述类的属性。

类具有操作,它是对象的行为的抽象,用操作名和实现该操作的方法来描述。

例如,学生成绩管理系统中,所管理的主要对象是学生,学生有具体的姓名、学号、各科成绩,因此认为学生是系统的可用对象,可抽象成为学生类,每个对象姓名、学号、各科的名称可抽象成为学生类的数据成员,对学生成绩的查询也抽象成对应类的成员函数。在系统实现时,首先定义学生类,对应的每一个学生对象对应一个学生类的实例。这一生成实例的过程类似C语言的结构体的使用过程,先定义结构体的类型,再生成结构体类型的变量。

3. 封装(Encapsulation)

封装就是将抽象得到的数据和行为(或功能)相结合,形成一个有机的整体,也就是将数据与操作数据的源代码进行有机的结合,形成“类”,其中数据和函数都是类的成员。

封装的目的是增强安全性和简化编程,使用者不必了解具体的实现细节,而只是要通过外部接口特定的访问权限来使用类的成员。

4. 消息 (Message)

消息是一个对象向另一个对象传递的信息。消息的使用类似于函数的调用,消息中指定了接收消息的对象、操作名和参数表(如果需要)。接收消息的对象执行消息中指定的操作,软件系统功能的实现就是一组对象通过执行相互之间的消息来完成的。一个对象的消息对应类的定义时声明的成员函数。

5. 继承 (Inheritance) 与派生 (Derived)

继承是指这样一种能力:它可以使用现有类的所有功能,并在无需重新编写原来的类的情况下对这些功能进行扩展。通过继承创建的新类称为“子类”。从现有类来讲,生成子类的过程称为派生,因此也叫“派生类”。被继承的类称为“基类”、“父类”或“超类”。继承的过程,就是从一般到特殊的过程,这一过程大大提高了软件开发的效率,使得程序员只考虑主要的功能处理就可以了。比如利用 Visual C++ 6.0(简称 VC++6.0)开发 Windows 应用程序时,可以直接继承系统提供的对话框类或窗口类之后,主要关注程序功能的开发,而不用过多在定义对话框或窗口上花费更多的精力,大大地提高了软件开发的效率。

6. 多态性 (Polymorphism)

多态性是允许将父对象设置成为和一个或更多的它的子对象相关的技术,赋值之后,父对象就可以根据当前赋值给它的子对象的特性以不同的方式运作。简单地说,就是允许将子类类型的指针赋值给父类类型的指针。

实现多态的方式是覆盖。覆盖,是指子类重新定义父类的虚函数的做法。当子类重新定义了父类的虚函数后,父类指针根据赋给它的不同的子类指针,动态地调用属于子类的该函数,这样的函数调用在编译期间是无法确定的。因此,这样的函数地址是在运行期绑定的,从而实现运行时的多态性。

7. 重载 (Overloading)

重载,是指允许存在多个同名函数,而这些函数的参数表不同(或许参数个数不同,或许参数类型不同,或许两者都不同)。

其实,重载的概念并不属于“面向对象编程”,重载的实现是编译器根据函数不同的参数表,对同名函数的名称做修饰,然后这些同名函数就成了不同的函数。对于重载函数的调用,在编译器编译时就已经确定了,是静态的。

1.2 C++语言的产生和特点

1.2.1 C++语言的产生

C 语言之所以要起名为“C”,是因为它主要参考一门叫 B 的语言,它的设计者认为 C 语言是 B 语言的进步,所以就起名为 C 语言;但是 B 语言并不是因为之前还有 A 语言,而是 B 语言的作者为了纪念他的妻子,他的妻子名字的第一个字母是 B;当 C 语言发展到顶峰的时刻,出现了一个版本叫 C with Class,那就是 C++最早的版本,在 C 语言中增加 class 关键字和类,当时有很多版本的 C 都希望在 C 语言中增加类的概念;后来 C 标准委员会决定为这个版本的 C 起个新的名字,征集了很多种名字,最后采纳了其中一个人的意见,以 C 语言中的++运算符来

体现它是 C 语言的进步,故而叫 C++,并成立了 C++标准委员会。

一开始 C++是作为 C 语言的增强版出现的,从给 C 语言增加类开始,不断地增加新特性。虚函数(virtual function)、运算符重载(operator overloading)、多重继承(multiple inheritance)、模板(template)、异常(exception)、命名空间(name space)逐渐被加入标准。1998 年,国际标准组织(ISO)颁布了 C++程序设计语言的国际标准 ISO/IEC 1488—1998。C++语言是具有国际标准的编程语言,通常称作 ANSI/ISO C++。1998 年是 C++标准委员会成立的第一年,以后每 5 年视实际需要更新一次标准,目前我们一般称该标准 C++0x,但是由于对于新特性的争端激烈,除了在 Technical Report 1(tr1)中的新增修改被基本确定外,完整的标准还遥遥无期。况且遗憾的是,由于 C++语言过于复杂,以及经历了长年的演变,直到现在都没有一个编译器完全符合这个标准。因此也衍生了很多 C++语言的不同版本。

如同它的名字表达的那样,C++语言是 C 语言的一个超集,它是一门混合型的语言,既支持传统的结构化程序设计,又支持面向对象的程序设计,这是 C++语言成功流行的一个重要原因。目前,最符合和接近 C++标准的编译器有 GNU GCC 4.5.0 和 Visual Studio 2010 等。

1.2.2 C++语言的特点

C++语言的主要特点有两个方面:全面支持 C 语言与面向对象。

C++语言从 C 语言发展而来,保持了 C 语言的简洁、高效和在某些操作上沿用了汇编语言指令的特点。同时,对 C 语言的类型进行了系统的改革和扩充,堵塞了 C 语言中的许多漏洞,C++语言编译提供了更好的类型检查和编译时的分析,能检查出更多的类型错误。C++语言改善了 C 语言的安全性,比 C 语言更安全。

由于 C++语言保持与 C 语言兼容,这就使许多代码不经修改就可在 C++语言编译器下通过,用 C 语言编写的众多库函数和实用软件可方便地移植到 C++语言中。因此,使用 C 语言的程序员能很快学会 C++语言,使用 C++语言进行编程。另外,用 C++语言编写的程序可读性好,代码结构更为合理。

C++语言的最重要特点是支持面向对象的程序设计,使用 C++语言编程,编程效率高;由于 C++语言引入了面向对象的技术和方法,因此 C++对于问题更容易描述,程序更容易理解与维护;C++语言的模板对库代码的重用提供了支持;C++语言更有利于大型程序设计。C++语言支持面向对象的程序设计方法,特别适合于中型和大型的软件开发项目,从开发时间、费用到软件的重用性、可扩充性、可维护性和可靠性等方面,C++语言均具有很大的优越性。

C++语言是一种支持多种程序设计方法的语言,提供对过程化和基于对象的程序设计方法的支持,适合于使用不同开发方法的编程人员。

1.3 Visual C++ 6.0 开发环境概述

C++语言版本较多,VC++6.0 是微软(Microsoft)公司开发的软件,在软件开发时应用较多,并与 Windows 操作系统完全兼容。本书就以 VC++6.0 为蓝本,介绍面向对象的程序设计知识。

1.3.1 Visual C++ 6.0 简介

VC++6.0 是 Microsoft 公司推出的一个基于 Windows 系统平台、可视化的集成开发环境,它的源程序按 C++语言的要求编写,并加入了微软提供的功能强大的 MFC(Microsoft Foundation Class)类库。MFC 中封装了大部分 Windows API(Application Programming Interface,应用程序编程接口)函数和 Windows 控件,它包含的功能涉及到整个 Windows 操作系统。MFC 不仅给用户提供了 Windows 图形环境下应用程序的框架,而且还提供了创建应用程序的组件,这样,开发人员不必从头设计、创建和管理一个标准 Windows 应用程序,而是从一个比较高的起点编程,故节省了大量的时间。另外,它提供了大量的代码,引导用户,从而简化了开发的流程。因此,使用 VC++6.0 提供的高度可视化的应用程序开发工具和 MFC 类库,可使应用程序开发变得简单。

1.3.2 第一个 Visual C++程序

计算机系统中只有安装 VC++6.0,才可以正常使用 VC++6.0 进行编程操作。VC++6.0 将开发项目所需要的文件整合到一起管理,这个文件就是工程文件。因此要想利用 VC++6.0 编写程序,首先要建立工程文件。下面我们就做一个实例,完成输出“Hello VC++6.0!”的一个程序。通过这个程序要掌握在 VC++6.0 下建立“Win32 Console Application”工程文件的方法,掌握工程文件的编辑、编译、运行的方法。从第1章到第10章,所有示例中所建的工程文件均是“Win32 Console Application”类型的工程,所以一定要掌握以下所述操作的过程。

(1)单击“开始|所有程序|Microsoft Visual Studio 6.0|Microsoft Visual C++ 6.0”,打开“Microsoft Visual C++”应用程序窗口,如图 1.1 所示。

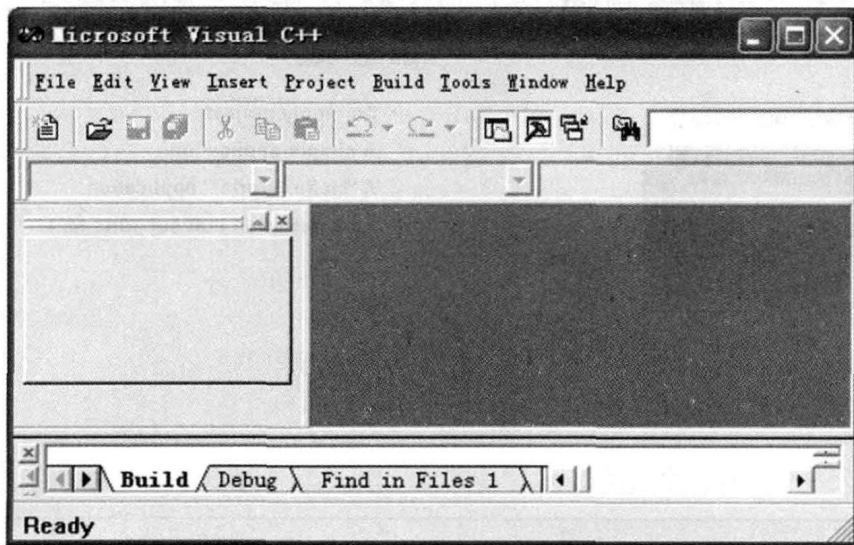


图 1.1 VC++6.0 应用程序窗口

(2)单击“File|New”或按下组合键“Ctrl+N”,打开“New”对话框,如图 1.2 所示。利用这个对话框可创建 VC++6.0 项目中所需要的文件。

(3)单击“Projects”选项卡,这个选项卡中列出了 VC++6.0 可以创建的工程种类。在列表框中选择“Win32 Console Application”选项,在“Project Name”文本框中输入“HelloVC”,作为工

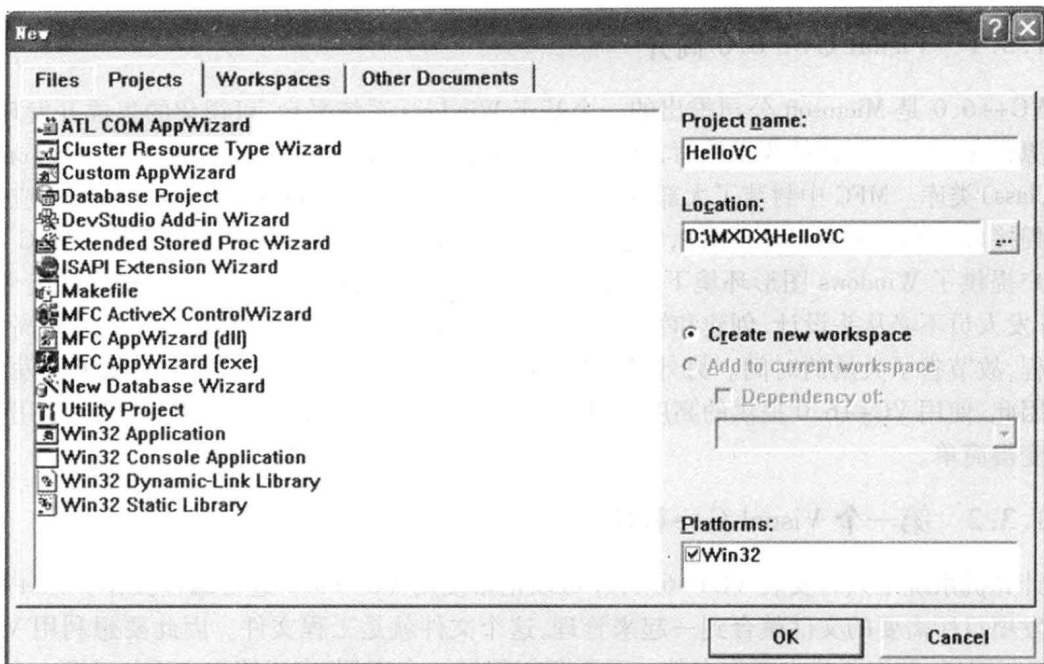


图 1.2 新建工程对话框

程文件的文件名。在“Location”文本框中输入路径或单击后面的按钮进行选择路径,确定工程文件的存放位置。选择“Create new workspace”单选按钮,单击“OK”按钮,打开“Win32 Console Application-step 1 of 1”对话框,如图 1.3 所示。

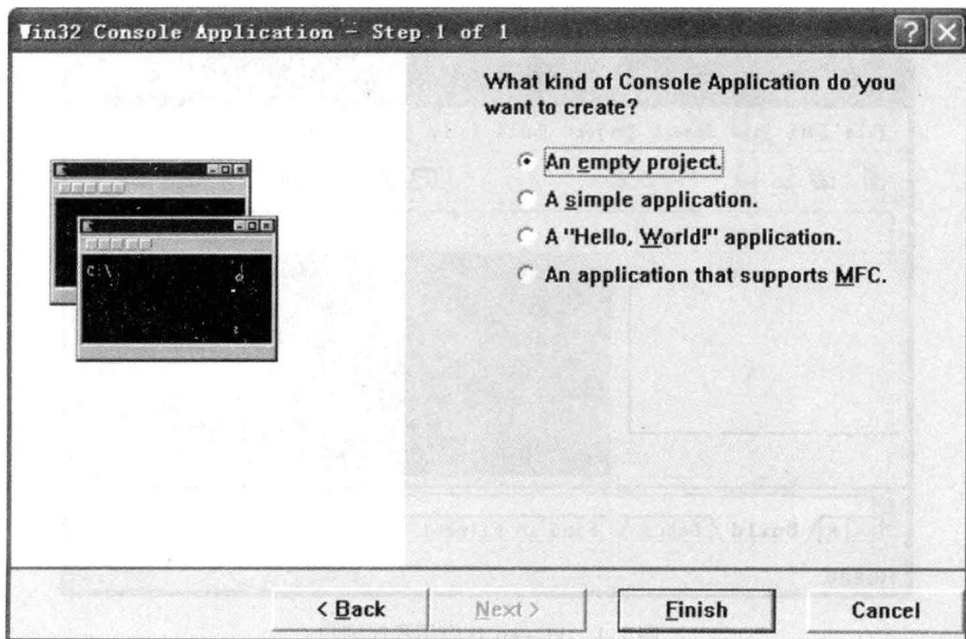


图 1.3 新建工程对话框 Step 1

(4)有 4 个单选按钮,可以选择工程的类型。选择“An empty project”单选按钮,单击“Finish”按钮,会打开“New Project Information”对话框,显示要建的新项目的基本信息,单击“OK”按钮,新建一个空白的工程文件。还需要继续在工程中添加相应的文件,本例添加一个

HelloVC. cpp文件。

(5)单击“File|New”或按下组合键“Ctrl+N”,打开“New”对话框,选择“files”选项卡,如图1.4所示。

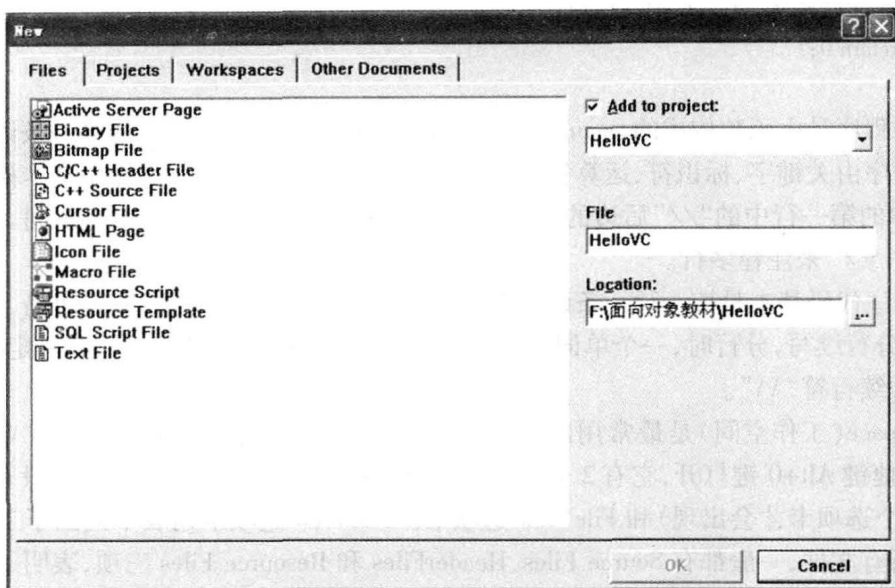


图 1.4 新建文件对话框

“file”选项卡中列出了在 VC++6.0 工程文件里可以添加的文件,如头文件,源文件等。选择“C++ Source File”表示选择要添加 cpp 文件,在“File”文本框中输入源文件名 HelloVC(可以不用输入扩展名),单击 OK 按钮,返回操作的主界面。接下来可以在 VC++6.0 的应用程序窗口中进行工程文件的管理、编辑,工程的编译、运行和调试等的操作,具体界面及标识如图1.5所示。



图 1.5 VC++6.0 的操作主界面

在右侧的文档编辑窗口中,列出的是 HelloVC. cpp 文件的内容,在编辑区的空白处输入如下内容。