高等学校教材

Visual C++ 程序开发基础

张基温 贾中宁 李 伟 编著

高等教育出版社

高 等 学 校 教 材

Visual C++程序开发基础

张基温 贾中宁 李 伟 编著

高 等 教 育 出 版 社

内容提要

本书主要介绍在 Visual C++环境下用 C++编程的基本概念和基本方法。全书共分 5 章：第 1 章主要介绍程序设计的一般概念、C++数据类型和运算的基本知识以及用 AppWizard 进行程序开发的一般方法；第 2 章介绍用 C++语言设计过程化程序的基本方法；第 3 章介绍 C++中常用的派生数据类型——数组和指针等；第 4 章介绍有关类与对象的基本概念和方法，第 5 章介绍 Visual C++的核心——MFC 及其应用。

本书深入浅出、例题丰富、概念清楚，不要求以 C 语言或 C++为先修，既可以作为高等学校本、专科有关专业的教材，也适合工程技术人员自学参考。

图书在版编目(CIP)数据

Visual C++程序开发基础/张基温等编著. —北京：高等教育出版社，2001

ISBN 7-04-008912-2

Ⅰ. V… Ⅱ. 张… 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(2001)第 07827 号

Visual C++程序开发基础

张基温 贾中宁 李伟 编著

出版发行	高等教育出版社		
社 址	北京市东城区沙滩后街 55 号	[illegible]0009	
电 话	010-64054588	010-[illegible]014048	
网 址	http://www.hep.edu.cn http://www.hep.com.cn		
经 销	新华书店北京发行所		
印 刷	中国青年出版社印刷厂		
开 本	787×1092 1/16	版 次	2001 年 3 月第 1 版
印 张	19	印 次	2001 年 3 月第 1 次印刷
字 数	470 000	定 价	23.90 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

前　言

程序是计算机的灵魂，程序设计是计算机生命的源泉。纵观历史，计算机的进步基本上是沿3条主线不断前进的：一是计算机元器件的进步，二是计算机体系结构的进步，三是程序设计、开发方法和工具的进步。开发方法和开发工具作为程序设计的两个车轮，相辅相成地将程序设计不断带向新的水平。

从程序设计方法上看，程序设计已经经过了面向过程、面向数据、面向对象、面向资源几个阶段。从程序的表述上看，程序设计经过了如下阶段：代码编程、表格编程和图形编程。代码编程是使用字符编码书写源程序，既可以书写面向过程的程序，又可以书写面向对象的程序。表格编程是使用表格或表示图表设计程序逻辑，通过编译工具自动产生程序代码，主要用于面向数据的程序设计。图形编程是由于图形用户界面（GUI）的出现而产生的程序开发平台，主要使用图形单元构件来构造应用程序。随着GUI技术的进一步发展和系统提供的构件的丰富，工具软件包越来越多，系统资源的利用成为程序开发的关键，面向资源的程序开发应运而生。这条发展路线遵循着一个“代码隐藏”的原则。Visual C++正是循着这样一条轨迹展现出的时代产品。它把Windows统一而漂亮的界面风格、面向对象的程序设计方法和面向资源的环境结合在一起，形成一个功能强大而复杂的C++编译器。它能提供简单而高效的操作方式、高效的内存管理、与设备无关的图形接口、数据共享和多任务的运行机制，同时又提供了一系列功能强大的开发工具和内容丰富的开发资源。

Visual C++是功能非常强大的程序设计语言，又是公认的难于入门的编程工具。目前介绍它的书籍要么是洋洋百万字以上的手册，要么是要求有C++基础的读者才能看懂的“指南”。这使它的推广受到了很大影响。没有C++基础甚至没有任何编程基础的人能不能使用Visual C++进行程序开发呢？这一问题便成了我们写这本书的动力。

本书共分为5章。第1章主要介绍C++的基本数据类型、基本运算和Visual C++程序的输入/输出方法。学习了这一章，读者就可编写简单的Visual C++程序了。以此为基础，后面的各章将逐步扩充读者的程序设计知识和能力。第2章介绍在C++程序中组织过程的基本机制：流程的基本控制结构、函数、变量的存储属性和编译预处理。学习完这一章，将具备构造复杂和较大过程的能力。第3章主要介绍数组和指针，学习完这一章，将可以用数组来组织同类数据，并且还将掌握C语言中应用极为普遍的数据类型——指针的能力。这一章还简单地介绍了枚举类型。学习完前3章，就可以具备用Visual C++进行过程化程序设计的基本能力。第4章介绍C++的面向对象机制：类的定义、对象的生成、继承与派生、虚函数。学完这一章，读者将获得面向对象程序设计的基本能力。第5章介绍MFC的基本概念和基本方法。

需要说明的是，本书不是一本Visual C++的手册，而是一本在Visual C++环境下进行程序设计教学的基础教材。本书把重点放在程序设计的基本方法上，想要进一步掌握Visual C++更

强大功能的读者，恐怕还需要参考其他 Visual C++的手册。

由于 Visual C++建立在 C++的基础上，而 C++建立在 C 的基础上，因此介绍 Visual C++，必然涉及 C++和 C 的一些概念，为便于区分这些概念的来处和使用范围，书中分别使用了 Visual C++(VC++)、C++和 C 的字样。另外程序中需由用户输入的内容采用白体，以与系统提供的内容（加粗）相区别。

本书由张基温、贾中宁、李伟编写。朱建明、尹晓峰、姚威参加了部分调试工作。

作为一种全新的编写尝试，本书在编写过程中，需要解决编程知识交迭、内容取舍合理等相当多的问题。这些问题的处理能否适合大多数读者的需求，还需要来自读者的反馈信息，特别希望同行和有关专家提出宝贵意见，以便能把本书修改得更好。

编 者

2000 年 8 月

目　录

第 1 章　Visual C++程序设计入门

1.1　引　　言

1.1.1　程序开发的一般过程

众所周知，计算机由硬件和软件两部分组成。硬件是计算机的物理组成，包括电路板、电子元件、接插件以及电源、外壳等。光有硬件，计算机还不能工作。计算机的工作是由程序控制的。程序是对计算机完成工作的方法的描述。不同的程序控制计算机完成不同的功能：有的用于系统管理，有的提供一种工作环境，有的完成一项具体的应用任务。一般说来，安装在计算机中的程序的总和就称为该计算机的软件。在不严格的场合下，人们也把完成一组相关任务的程序称为一个软件。

一般说来，一个程序的开发需要经过如下步骤：

1. 分析问题，建立问题的模型

模型是对问题进行抽象的结果。它忽略了人们所不关心的方面或不必要的细节，保留了所关心的和重要的内容，以使人们深刻理解问题的本质，掌握解题的基本途径。如对“原来有 3 头牛，又买了 2 头，现共有几头牛”这样的问题，可以抽象为：“3+2”，这是一个忽略了具体事物本身，只研究其数量关系的模型。

当前，在程序开发时所使用的建模方法主要有两种：过程模型和对象模型。

面向过程的模型认为程序是实现一些功能，每个功能通过一系列的操作实现。例如：在 3 个数（a,b,c）中求最大数的过程如图 1.1 所示，其中 max 代表最大数。求两个正整数 u 与 v 的最大公约数的过程如图 1.2 所示，其中 u 为大数，v 为小数。

面向对象的模型认为世界是由各种各样的对象(object)所组成。对象是现实世界中的有意义的事物,可以是物质的,也可以是一种概念，一般说来对象的外在特征表现为行为和属性,不同对象间的相互作用和通讯构成了现实世界；复杂对象由简单对象组成。因此基于对象的分析是一种结构分析，它围绕着现实世界中的对象来构造系统模型，而不是围绕功能—过程来构造系统模型。

分类是对复杂系统进行结构分析的主要方法。通过分类对组成系统的对象进行抽象，可以简化系统的模型，并容易理解系统的实质。属于同一类型的对象有类似的属性、共同的行为（操作）、与别的对象有类似的关系、共同的语义。例如，不管在哪个坐标点上，划出一个什么样（半径，颜色）的圆，都有相似的属性（半径，圆心坐标，颜色）、共同的行为（被平移，被放大，被显示，被选择等）。具有一组类似性质的对象，用对象类（class）来描述。

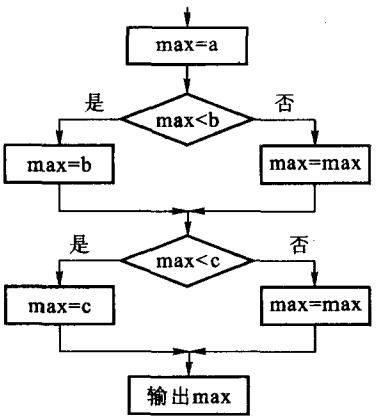

图 1.1　求最大数流程

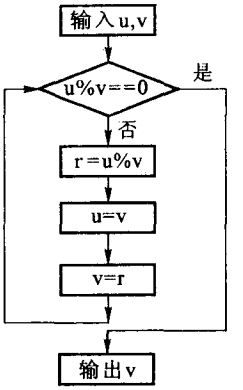

图 1.2　求最大公约数流

类具有层次关系。在图 1.3 所示“图形”(figure)类层次结构图中,每个对象类由类名、属性、方法（行为）三部分来描述。在类层次结构中，父类是子类的超集，即父类所描述的对象包含了其任一子类所描述的对象。或者说，子类比父类具有更多的属性，以表明子类对象的特殊性。如一个圆类比一般二维图共同属性要多一个属性（直径 diameter)。也可以说，父类派生了子类，或子类继承了父类的性质。在类层次结构中，位于最上层的类称为该类层次结构的基类。本例的

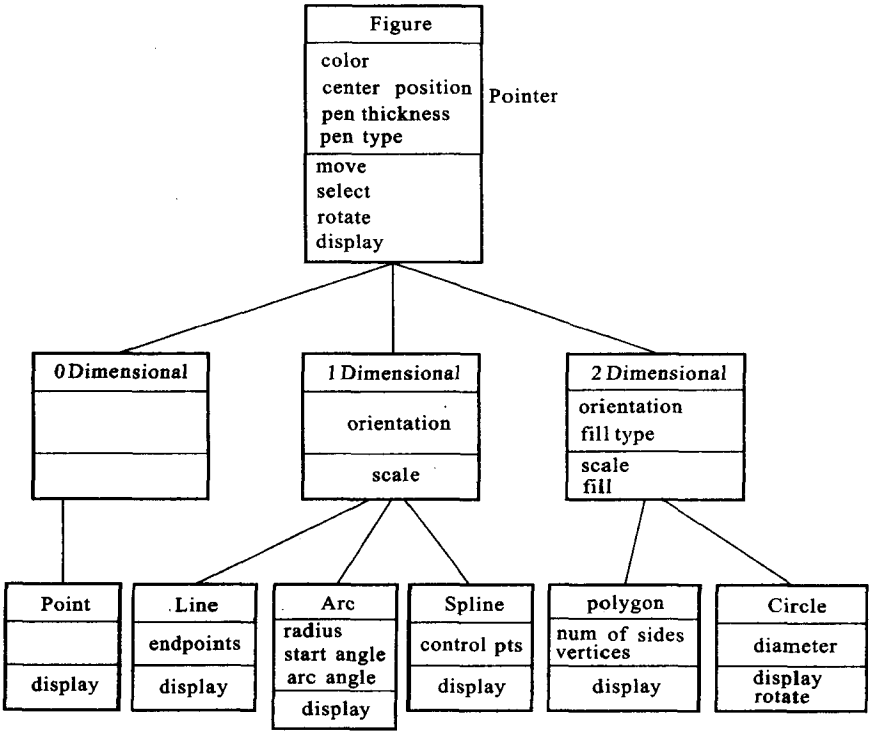

图 1.3　Figure 类层次结构

类层次结构中的基类是 pointer。

在对系统进行结构分析时,为了深刻理解对象行为,常常要对对象进行动态分析，建立动态对象的模型，做为对象模型的补充。

不同的模型要采用不同的方法，因而程序开发就有面向过程和面向对象两种方法。

2. 表现模型

表现模型就是用一种符号系统——语言来描述模型。或者说是用一种语言来写程序。这种语言就称为程序设计语言。

目前计算机程序设计语言很多，从方便机器理解还是方便人理解的角度，可将程序设计语言分为低级语言（面向机器的语言）和高级语言两大类。从所适合的描述模型来看，高级程序设计语言可分为面向过程的程序设计语言和面向对象的程序设计语言两大类。目前应用最为普遍的 C 语言就是一种面向过程的程序设计语言。而 C++是在 C 语言的基础上汲取面向对象的机制，形成的一种多范型的程序设计语言。本书所介绍的 Visual C++（后面简称 VC++）则是一种基于 C++和 Windows 图形用户接口风格的高级程序设计语言。

把程序输入计算机并进行修改、存储的过程称为编辑。编辑又分为源程序编辑和资源编辑。源程序是用文本方式(字符、数字、运算符等)显示的符号序列的集合，又称为源代码。资源是在程序中使用的菜单、对话框、工具条、按钮、位图等部件，它们在外观上表现为某种图形或图像，在计算机内部则是以二进制方式存储的数据文件。源程序编辑和资源编辑都是用编辑工具——编辑器完成的。虽然源程序编辑和资源编辑采用不同的编辑器，但在 VC++中，这些工具和编译器、连接器以及调试器等工具都被组装在一个集成开发环境——Developer Studio 中。

3. 程序的编译（或解释）与连接

现在的计算机还只能解释并执行用 0，1 码书写的程序，不能直接解释执行源代码。为了让机器执行程序，必须先将源代码翻译成机器能直接理解的 0，1 码语言描述的代码。翻译方法分为编译和解释两种。编译犹如笔译，是一块一块地进行翻译，先不考虑执行。解释则有点像口译，是翻译一句，执行一句。

编译后用 0，1 码表示的程序文件称为目标程序文件。目标程序文件往往还不是完整的程序。这是因为，对于大的程序或是为了某种需要，一个程序往往要被分成多个模块进行分别设计、分别编辑、分别编译、分别调试；并且几乎所有的程序都要用到系统提供的某些模块。为了让程序能完成预定的任务，必须将目标文件、资源文件以及所用到的系统提供的模块连成一个整体。这一过程称为程序文件的连接。经连接的程序文件，才是可执行程序文件。

程序在编译、连接过程中，也可能发现错误，为此要重新进入编辑器进行编辑。

4. 程序的测试与调试

经编译、连接的程序文件，生成可执行文件，就可以让计算机执行了。但是，并不是就可以得到预期的结果而交付用户使用了，因为程序仍然会存在某些错误。因此，在交付用户使用之前，

还要对程序进行测试。所谓测试，就是用一些数据（称为测试数据）来试运行程序，来找出程序中的错误。

程序测试的正确指导思想是：

① 以任何程序都存在错误为前提；

② 测试的目的是找出程序中的错误，而不是证明程序的正确；

③ 为了保证测试成功，需要很好地设计一组测试用例，试运行程序；

④ 一般说来，测试都不可能是完全的，成功的测试是能够找出更多错误的测试。

测试发现错误后，还要进一步找出错误的位置，并纠正错误。这一过程称为程序的调试。为了方便程序的调试，许多系统提供有程序调试器，提供单步执行等手段，让调试者尽快地找出程序中的错误。

找出错误后，还要再一次进行程序的编辑—编译—连接—测试—调试。

5. 程序的维护

程序交付用户使用之后，并不是万事大吉了。由于多种原因，还可能要对程序进行修改。交付之后对程序的修改称为程序的维护。维护程序的原因主要有：

① 原来的程序没有完全满足用户要求；

② 用户要求改变；

③ 程序中遗留有错误，在运行中被发现。

程序的维护可以由开发者进行，也可能是由别人进行。为能便于程序的维护，开发者应当提供必要的技术资料，并且要保证程序的可读性好——能让人看懂。

当然，为了让用户正确地使用程序，还应当提供用户说明书。现在，这些内容多以“Read me”和“Help”的形式提供给用户。

图 1.4 为程序的一般开发过程。

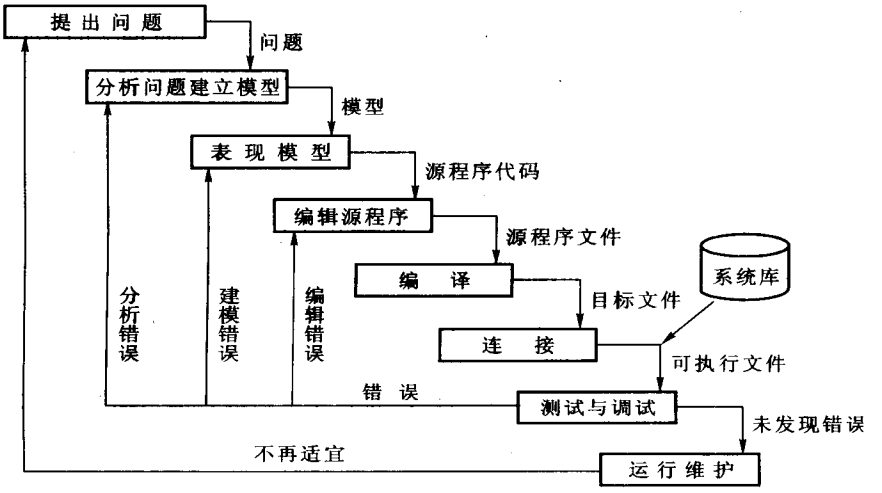

图 1.4　程序开发的一般过程

1.1.2　Windows 程序的特点

所谓 Windows 程序，具有两重含义：一是指能够在 Windows 环境下执行的程序；更进一层的意思是具有 Windows 风格的程序。具体地讲，Windows 风格的程序具有以下几个特点：

1. 友好、标准的 Windows 用户界面

随着应用程序的不断发展，用户界面的地位越来越重要。一个应用程序的用户界面设计得好，用户使用起来就方便。一般说来，Windows 的用户界面有如下特色：

(1) 使用窗口作为用户与应用程序的通信工具

窗口是屏幕上一块受控制的图形化矩形区，以提供应用程序与用户交互的场所。Windows 允许用户同时运行多个应用程序。这时键盘、鼠标、显示器是共享的，每个程序独享的只是屏幕上的一个矩形的区域——窗口。窗口的意义不仅表明程序在屏幕上占一隅之地，更重要的是，Windows 程序依赖窗口而存在：有了窗口，程序才能有条件地接收来自用户的信息，才能把自己的工作进程和有关信息展示给用户。一个 Windows 应用程序可能会有多个窗口，但其中有一个其赖以生存的窗口，这个窗口称为主窗口。每个 Windows 程序的第一个任务就是建立这个主窗口。主窗口消亡，标明程序结束。

除此之外。窗口还用于组织所有其他需要的用户界面对象，例如菜单、光标、图标、对话框、工具条、消息条、滚动条等。当然，菜单、工具条、消息条、滚动条、对话框以及按钮也是窗口。

典型的窗口可分为两个部分：由系统绘制的“非客户区”和由应用程序绘制的“客户区”。可以放置在非客户区的组件有标题条、系统菜单、应用程序菜单、可缩放的边界和滚动条等。这些组件大多是可选的。

(2) 操作的简便性、直接性、可辨性

每一个 Windows 应用程序都可缩小为一个图标，被激活时，将展开为一个窗口。用户通过菜单、工具条、对话框等可视化的窗口组件简单、直接地操作程序。例如，使用菜单或带有按钮、复选框控件的对话框直接方便地指示程序执行特定的任务；使用带有编辑框控件的对话框简便地给程序输入数据……

(3) 一致性

应用程序界面的一致性以及应用程序界面与内部的一致性。

(4) 用户控制性

用户可以控制程序执行的全过程，而不是像 DOS 程序那样，只要一开始执行，用户就只能等待结果，中间不可控制。它能在用户的控制之下按照用户的意图执行指定的任务。

(5) 反馈性

能表明用户对程序的操作以及操作企图，如在一个菜单中，鼠标指针移向某一命令，该命令条的颜色就发生变化，以保证操作的正确性。

(6) 宽容性

选择重要的操作之后，并不立即执行，还给用户提供选择“确认”或“撤消”的最后机会。

2. 数据输出与设备无关

在 Windows 应用程序中，不管处理的数据是文本还是图形，都是以图形方式进行输出处理的，即采用了统一的图形用户接口。当需要输出数据时，不管实际的输出设备是屏幕、打印机还是绘图仪，在程序中使用的都是同一种方法（使用统一的函数）。

3. 事件消息机制

Windows 与其他传统程序有一个很大的区别就是它的运行过程处于被动状态：程序初始化之后，并不知道下一步要做什么，而是处于等待某一事件。一般说来，事件来自 3 个方面：

① 来自鼠标或键盘；

② 来自菜单、工具按钮或对话框控件；

③ 来自 Windows 本身，如恢复一个被其他窗口覆盖的窗口等。

一个事件发生后，系统将通知 Windows，并告诉它事件的类型、发生时间以及与屏幕相关事件的位置。当 Windows 知道事件发生后，就把相关信息（如位置和时间等）组织成一个数据结构，与消息一起发送给相关的程序。这种运行方式称为事件驱动系统或消息驱动。

4. 特殊的数据——资源

通常，一个程序元件可分为两大类：操作码和数据。这里，数据一般是指程序操作的对象——变量或常量。但是，一个典型的 Windows 应用程序还具有另外一种特殊的数据——资源。资源决定程序用户界面的形象特征。一个程序的资源定义界面元素有：

① Accelerator（快捷键）

② Dialog（对话框）

③ Icon（图标）、Cursor（光标）和 Bitmap（位图）

④ Menu（菜单）

⑤ String Table（字符串表）

⑥ Toolbar（工具栏）

当 Windows 调用一个应用程序时，就从程序的可执行文件中读取操作代码并初始化操作数据（变量），同时对它们分配内存。资源数据保存在资源文件中，是在程序运行时（程序创建窗口、显示对话框或装载位图时）被读取，而不是程序装载时被读取。

使用资源一方面可以使 Windows 应用程序的外观和功能更加标准化、更具有一致性，另一方面可以减少应用程序的长度，减轻程序员的工作量。

5. 使用句柄进行实体管理

在 Windows 程序中要用到实体——程序、文件、对象、程序实例、对象实例、各种表（快捷键表、设备描述表等）以及资源等。为了对这众多的实体进行管理，Windows 在内部为它们建立了一个索引表，以便系统查找。索引表的索引，就称为“句柄”（Handle）。句柄的值是一个数字（在 Win16 中是 16 位，在 Win32 中是 32 位）。在程序中一般用宏名进行标识。表 1.1 是一些

公共 Windows 对象及其句柄。

表 1.1 公共 Windows 对象及其句柄

对　象	句　柄	对　象	句　柄
快捷键表	HACCEL	图表	HICON
位图	HBITMAP	菜单	HMENU
画刷	HBRUSH	粘贴板	HPALETTE
光标	HCURSOR	画笔	HPEN
设备描述表	HDC	区域	HRGN
文件	HFILE	窗口	HWND
字体	HFONT		

在 Windows 应用程序中，句柄的应用是很频繁的，Windows 使用句柄索引来存取保存在响应表中的信息，应用程序通过句柄能够访问相应对象的信息。

1.1.3 集成开发环境 Developer Studio

简单地说，Visual C++是从 C++和 Windows SDK 的基础上发展而成的。DOS C++脱胎于 DOS C，但却还没有实现换骨，仅仅是在面向过程的 C 之上，又添加了一些面向对象的机制。程序的许多方面还保留着过程化的影子：

① 每一个 C++程序与 C 程序一样，都由一系列的函数组成，并且所有的函数都直接地或间接地通过一个称为主函数 main（）的函数调用而运行；

② 对象之间的作用也通过函数调用而发生；

③ 程序设计的主要工作就是设计一系列的函数。

基于 SDK 的 Windows C++在实现可视化的同时又向面向对象迈进了一大步。它一方面引入了 Windows 的看家本领——窗口；同时引入了事件消息机制。但是，如同 DOS C++一样，每一个应用程序都要有一个起始函数 WinMain（）（相当于 DOS C++的 Main（）），并且需要繁多的编程规则。

基于 MFC 的 Visual C++从根本上改变了上述不足。Visual C++的 Developer Studio 是 Microsoft 公司为开发 Windows 应用程序而创建的一种集成开发环境，它不但集成有编辑、编译、连接、调试等程序开发工具，更重要的是提供了全方位的自动编程服务工具——编程向导 Wizard（包括 AppWizard 和 ClassWizard）。使用了编程向导，可以大大提高程序开发的效率。Developer Studio 对使用 ODBC/ADD 访问数据库、ActiveX/OLE 编程、网络通信和 Internet 编程都能提供良好的支持，是最有力的开发工具之一。详细了解 Developer Studio，需要参考、查阅有关技术手册。下面仅对本书将涉及的几点作简要介绍。

1. Developer Studio 的特点

(1) 由于每个程序都有大致相同的结构，这些相同的部分用同样的代码描述。为减轻程序员的负担，AppWizard 可以自动生成通用的程序框架代码，并把一部分每个程序都用得到的

代码而程序员无须看见的部分（如与 main()和 WinMain()相当的程序入口 AfxWinMain()）隐藏起来，只开放程序员需要知道的部分。程序员的工作就是在这个框架中添入自己的代码。下面的程序 1.1 就是一个输出函数框架。其中：以“//”开头的行称为注释行，用于向阅读程序者作一些说明或提示。第 2 个注释行提示程序员下一行就是写自己的代码的开始位置。

程序 1.1

```
// COutTextView drawing    以 "// "开头的程序行是注释行,仅对程序进行说明，以便阅读
void COutTextView::OnDraw(CDC* pDC)
{
    COutTextDoc* pDoc = GetDocument();    //  分号表明一个语句结束
    ASSERT_VALID(pDoc);

    // TODO:add draw code for native data here  ——提示程序员从此开始增添代码
}
```

(2) 程序的开发过程，是由应用程序向导 AppWizard 引导完成的，便于学习掌握。

(3) 提供了一套功能强大的基础类库，即微软基础类库 MFC（Microsoft Foundation Class）。MFC 中包含许多类，每个类中又包含着许多完成相关具体任务的成员函数（完成某种任务的程序模块）。设计程序时可以根据目的在相关类中快速查找相应的成员函数，从而大大简化了 Windows 编程。例如：

① 要处理文档，可以在 CDocument（文档类）类中查找相应函数。

② 要输出文本或图形，可以在 CDC（设备环境类 DC：Device Context）类中查找相应函数。

③ 要设计对话框，可以在 CDialog（对话框类）类中查找相应函数。

④ 要处理文件，可以在 CFile(文件处理类)类中查找相应函数。

(4) 引入了文档视图结构。

文档和视图是 MFC 的两类对象：一个文档（Document）对象就是用户处理的数据单位，用来维护、管理数据，包括数据的读取、存储与修改；视图（View）对象用来接收并显示数据，将这些数据交给文档类来处理。也就是说，数据来自文档，而用户通过视图的过滤来查看和编辑数据。按照 MFC 程序的结构和风格，应该在视图中显示数据，在文档类中定义、处理数据。于是，“完整”的程序代码主要被分散到视图和文档两个类中了。为了不同的需要和方便，用户往往需要用多种不同的方式查看和编辑特定的文档，例如在字处理时，用户有时希望以大纲方式显示文本，有时则希望用页面方式显示文本；在表处理时，有时希望以条形图来显示数据，有时则希望以轮廓图来显示数据。这样一个文档对象就会对应多个视图对象，修改文档内容使各视图都统一修改。MFC 将数据的文档与视图相隔离，可以给程序的开发带来很大的方便。

AppWizard 可以建立 3 种类型的应用程序结构：

① MDI(Multiple Document Interface)：多文档界面，允许同时打开多个文档。

② SDI(Single Document Interface)：单文档界面，一次只允许打开一个文档。

③ Dialog based：基于对话的应用程序，不支持文档视图结构，仅显示一个简单的对话框，主要用于编写小程序。

图 1.5 和图 1.6 分别是 MDI 和 SDI 类型应用程序的例子。MDI 类型的应用程序多用于像 Word ,Excel 那样的规模比较大的应用程序。

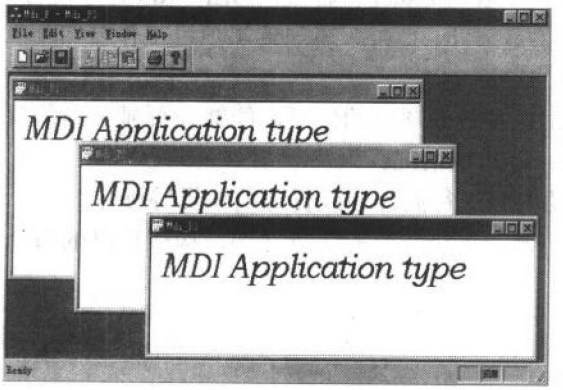

图 1.5　多文档界面应用程序的外观

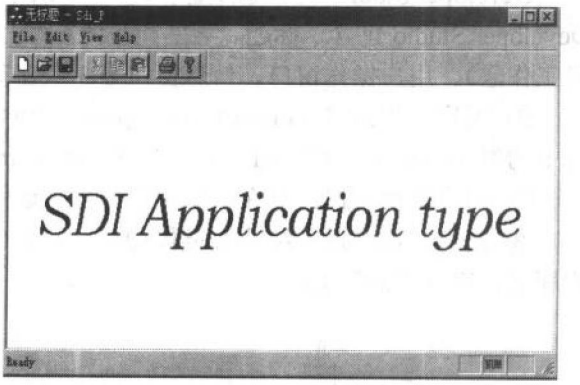

图 1.6　单文档界面的外观

(5) 提供强有力的资源管理功能。资源是 Windows 应用程序的重要组成部分。每创建一个 Windows 应用程序后，当打开该程序时，通过 Workspace 进入 Xxx resources(Xxx 为应用程序名)，就进入了该程序的资源编辑器。使用资源编辑器，程序员可以对资源进行编辑，还可以根据需要新增一些资源。

(6) 在 MFC 中用指针来标识对象。为了跨越 MFC 与 Windows，MFC 维护着一个查询表集合——句柄映像，用以在 Windows 对象与 C++/MFC 对象间的一一对应关系。

每个句柄都只对创建对象的系统组件有意义。MFC 创建的句柄映像有两种类型：永久性的和临时性的。永久性的句柄映像应用于通过 MFC 创建的 Windows API 对象。对于不是通过 MFC 创建的 Windows API 对象，则使用临时性的句柄映像。

(7) 以项目作为应用程序开发的基本单位。从软件工程的角度，每个程序的开发工作，都是一项工程项目（Project)：它要涉及计算机以及相关专业等领域的知识及其应用，还要使用代码生成、编辑、编译、链接、调试等一系列的工具，并且所生成的程序不是仅由一个文件构成，而需要一组相互关联的源文件和一些资源文件的支持。在集成化开发环境中，把这一组相互关联的源文件、资源文件以及支撑这文件的类的集合称为一个工程项目（Project，后面简称做项目)。项目是应用程序开发的基本单位，用户的工作都是围绕项目进行的。开发一个应用程序时，首先要确定一个项目名，这个项目名就是最后生成的应用程序（可执行文件）名。项目经编译、连接，生成可执行文件后，便可认为该任务的最终完成。Visual C++ 支持项目管理，并以项目工作区的形式来组织文件、项目和项目配置。在项目工作区中，给出了构成项目的源文件、资源文件以及支撑这些文件的类。项目工作区是由项目工作区文件（项目名. DSW）进行管理的，这个文件中包含了工作区的定义和项目中所包含的文件以及支持类的所有信息。

2. Developer Studio 界面

Developer Studio 工作时的窗口如图 1.7 所示。像 Word 、Excel 等标准的 Windows 程序一样，Developer Studio 包含了标题、菜单、工具条、状态条等部件。此外，还配置了项目工作窗口（项目工作区）、用户操作窗口、信息窗口等 3 个窗口：

① 项目工作窗口（Project work space）：包含正在开发的项目的信息，能够分页显示当前项目中的类（Class）、文件（Files）、资源（Resource）等构造情况。

② 用户操作窗口：对在项目工作窗口中选中的内容进行源程序或资源的编辑等操作。

③ 信息窗口：进行编译及链接操作时，给出编译、链接情况的信息（如，构成项目的每个程序名，错误信息等）。

图 1.7　工作中的 Developer Studio

“Developer Studio”在菜单中包含了其所有的功能。也可以将常用部分放在工具条上。针对项目的操作大多集中在菜单“Project”和“Build”上，其中有项目的设置（Setting）、应用程序的生成（包含程序和资源的编译、连接直至生成执行文件的一系列步骤）、调试（Debug）、运行程序等。

1.1.4　MFC AppWizard 生成一个简单程序的过程

1. 使用 MFC AppWizard[exe] 制作程序框架

使用 MFC AppWizard[exe]制作程序框架的操作步骤如下：

(1) 准备：确定项目名和项目目录