计 算 机 基 础 课 程 系 列 教 材

Visual C++ .NET 程序设计教程

本书配有
教学课件

郑阿奇 主编
丁有和 刘 毅 编著

机械工业出版社
China Machine Press

计 算 机 基 础 课 程 系 列 教 材

Visual C++ .NET 程序设计教程

郑阿奇 主编
丁有和 刘 毅 编著

机械工业出版社
China Machine Press

本书以Visual C++ .NET 2003为平台，从全新的角度介绍Visual C++ .NET 2003开发编程和应用。首先介绍Visual C++ .NET开发环境，然后介绍托管C++基础和托管C++面向对象编程。在此基础上，系统介绍Windows窗体和对话框、常用控件、菜单、工具栏和状态栏、GDI+与图像处理、文档界面模型和文件操作、数据库等。本书第一部分是Visual C++ .NET 2003教程，每章后有习题，第二部分为实验，最后还有综合应用实习。为了方便教学需要，本书配有教学课件和配套的应用程序实例。只要阅读本书，结合实验指导进行练习和实习，就能在较短的时间内基本掌握Visual C++ .NET 2003及其应用技术。

本书可作为大学本科、高职高专有关课程的教材，也可作为广大Visual C++ .NET用户的自学参考书。

图书在版编目（CIP）数据

Visual C++.NET程序设计教程 / 郑阿奇主编. －北京：机械工业出版社，2005.8
（计算机基础课程系列教材）
ISBN 7-111-16938-7

Ⅰ.V… Ⅱ.郑… Ⅲ.C语言－程序设计－教材 Ⅳ.TP3

中国版本图书馆CIP数据核字（2005）第079841号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
策划编辑：温莉芳
责任编辑：傅志红
北京中兴印刷有限公司印刷 · 新华书店北京发行所发行
2005年8月第1版第1次印刷
787mm × 1092mm 1/16 · 21.75印张
印数：0 001-4 000册
定价：30.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

前　言

微软公司.NET平台提供的软件开发和部署具有强大功能，包括独立于特定语言和平台的能力，使开发者用他们最擅长的语言（如Visual Basic .NET、Visual C++ .NET、C#等）就可以完成相同的软件项目。.NET程序能够驻留在多个平台上，或者在多个平台之间进行通信，从而促进了Web服务在Internet上的传输。它创造了软件开发的新范型，提高了编程效率，缩短了开发时间。

Visual C++ .NET 2003作为Visual Studio .NET 2003的重要组成部分，在保留原Visual C++编程能力的基础上，加入了.NET环境中一系列激动人心的新特性。从教学的角度讲，虽然不是以追求最新内容为目标，但是以应用为目的学习C++开发和编程课程，教授Visual C++ .NET 2003课程是适宜的。

本书从全新的角度介绍Visual C++ .NET 2003，从而真正掌握它的核心和精彩之处。而且，本书提供的部分实例有一些闪光点，这是读者非常想了解而在其他同类书中不一定能够找到的，这些都有利于教学和应用。

本书第一部分是Visual C++ .NET 2003教程，每章后有习题，第二部分为实验，最后还有综合应用实习。为了方便教学，本书配有教学课件和配套的应用程序实例，需要者登录华章网站下载。

实际上，本教程不仅适合于教学，也非常适合于Visual C++ .NET 2003的各类培训，以及供开发应用程序的用户学习和参考。只要阅读本书，结合实验指导进行练习和实习，就能在较短的时间内基本掌握Visual C++ .NET 2003及其应用技术。

本书由丁有和（南京师范大学）和刘毅（南京审计学院）编写，郑阿奇（南京师范大学）对全书进行统编、定稿。刘建、刘中、郑进、周怡君、李莉等其他同志对本书的编写提供了帮助，在此一并表示感谢!

编　者

2005年6月

目　录

第二部分 实 验

第三部分 附 录

第一部分 教 程

第1章 Visual C++.NET开发环境

随着Internet的发展，Web应用平台的无关性变得越来越重要，Java语言的出现与应用，推动了软件技术的发展与变革，同时也给软件业带来了巨大的冲击。为了应对挑战和实现统一的目的，2000年微软公司推出了它的.NET重要战略计划。Microsoft.NET策略是将Internet本身作为构建新一代操作系统的基础，是Internet和操作系统设计思想的合理延伸，它不仅会改变开发人员开发应用程序的方式，而且使开发人员能创建出各种全新的应用程序。

为了配合.NET应用程序的开发，微软公司推出了新一代的开发平台Visual Studio.NET，它是快速建立企业级Web应用程序和高性能桌面应用程序的集成开发环境。在这个开发环境中，包括Visual C++、Visual Basic、Visual C#和Visual J#等开发工具。

本书以Visual Studio.NET 2003中文企业版作为编程环境，所有程序均在Windows 2000 /XP中调试运行。

1.1 开发环境简介

当Visual Studio.NET 2003中文企业版安装成功后，操作系统“程序”菜单中就会包含“Microsoft Visual Studio.NET 2003”菜单项。在使用Visual Studio.NET进行项目开发之前，先来浏览一下它的开发环境。

1.1.1 概述

单击Windows操作系统任务栏的“开始”菜单，选择“程序”中的“Microsoft Visual Studio.NET 2003”，然后选择“Microsoft Visual Studio.NET 2003”菜单，就能运行Visual Studio.NET，并出现如图1-1所示的开发环境。

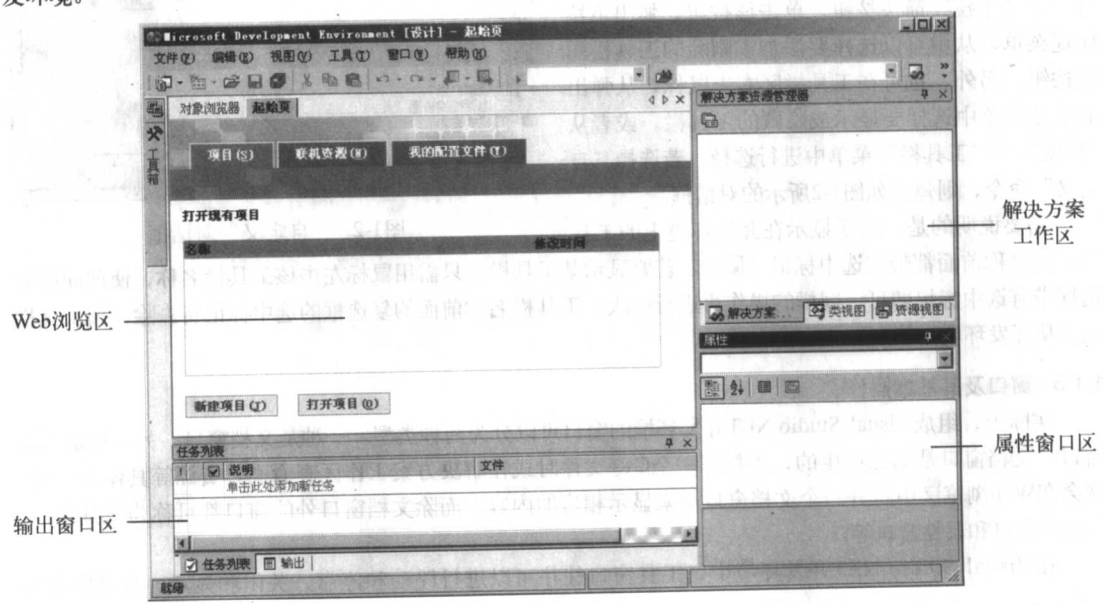

图1-1 Microsoft Visual Studio.NET开发环境

Visual Studio.NET开发环境具有和Office XP相似的平面化界面，并且它将IE浏览器嵌入其中，可以直接浏览网页。

从Visual Studio.NET开发环境中可以看出，它除了具有和Windows窗口一样的标题栏、菜单栏、工具栏和状态栏外，最主要的是还有不一样的窗口区。窗口区由Web浏览区、解决方案工作区、输出窗口区、属性窗口区以及其他窗口组成。

Web浏览区窗口是一个多功能文档窗口，各种程序代码的源文件、资源内容、文档、Web页面等都可以通过该窗口显示。默认时，该浏览窗口显示的是起始页面。

解决方案工作区是由“解决方案资源管理器”窗口、“类视图”窗口及“资源视图”窗口等组成，用来显示解决方案中的一些信息，包括类、解决方案项目以及资源等。

输出窗口区是由“任务列表”窗口、“命令”窗口、“输出”窗口及“查找”窗口等组成，用来显示任务、编连信息以及查找内容等。

属性窗口区是由“属性”窗口、“动态帮助”窗口及“收藏”窗口等组成，用来显示各种对象的属性、动态的帮助项目以及用户收藏的页面地址。

1.1.2　开发环境的菜单和工具栏

在开发环境界面的上方排列着一系列的菜单，每一个菜单下都有各自的菜单命令。由于是中文界面，因而绝大多数命令功能均可望文知义，故这里不再重复。需要说明的是，随着开发环境当前状态的改变，菜单栏以及菜单中的命令项也随之变化。例如，当新建或打开一个项目时，开发环境自动增加“项目”、“生成”和“调试”菜单，而当文档窗口没有任何源文件时，“编辑”菜单中的许多命令项都是灰色的，用户不能使用它们。

菜单栏下面是工具栏。工具栏是一系列工具按钮的组合，当鼠标停留在工具栏按钮上时，按钮凸起，如果光标停留时间稍长一些，就会出现一个小的弹出式工具提示窗口，用来显示该按钮的名称。工具栏上的按钮通常和一些菜单命令相对应，为经常使用的命令提供一种执行的快捷方式。

在Visual Studio.NET第一次运行时，开发环境只显示一个“标准”工具栏。在每个工具栏的右方有一个“下拉”箭头按钮，单击该按钮，弹出下拉快捷菜单，从中可以选择要添加或删除的工具栏图标按钮。另外，也可在工具栏区右击鼠标，从弹出的快捷菜单中选择要显示或隐藏的工具栏，或者从“视图”→“工具栏”菜单中进行选择。若选择“自定义”命令，则弹出如图1-2所示的对话框。

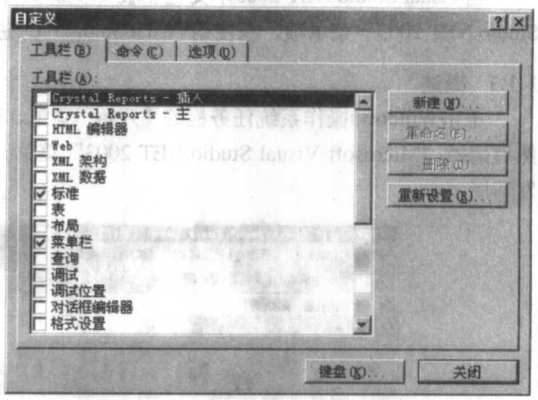

图1-2　“自定义”对话框

需要说明的是，对于显示在开发环境上的工具栏，其名称前面都带有选中标记（☑）。若要显示某工具栏，只需用鼠标左击该工具栏名称，使前面的复选框带有选中标记即可；同样的操作再进行一次，工具栏名称前面的复选框的选中标记将去除，该工具栏就会从开发环境中消失。

1.1.3　窗口及其基本操作

实际上，组成Visual Studio.NET开发环境的窗口可以分为两种类型：一种是文档窗口，另一种是工具窗口。文档窗口是动态产生的，当打开一个C++文件时或在解决方案工作区查看类、资源等具体内容时，就会在Web浏览区中打开一个文档窗口用来显示相应的内容。而除文档窗口外的窗口都可称为工具窗口，如输出窗口和属性窗口等。

在Visual Studio.NET开发环境中，工具窗口往往可以进行浮动和停靠、关闭和显示、自动隐藏等操作。

1. 浮动和停靠

Visual Studio.NET刚开始运行时，窗口区中的各种窗口均处于停靠状态，任何时候用鼠标双击窗口标题栏，都会在浮动和停靠间进行切换。若用鼠标单击某个窗口不放，可将其拖放到整个窗口区的任何位置，这个位置可以是屏幕的任何位置。被拖放的窗口既可单独显示在开发环境界面中的某处，也可与窗口区的其他窗口构成一组。

需要说明的是，若单击某个工具窗口后，选择“窗口”菜单中“可停靠”项，可将工具窗口显示在Web浏览区，同样的操作可将该工具窗口恢复到原来的位置。

2. 关闭和显示

在窗口区中，每个活动窗口的标题栏处都有自动隐藏和关闭按钮，如图1-3所示。单击关闭按钮后，窗口被关闭，但可通过选择“视图”菜单下的菜单命令恢复显示相应的窗口。

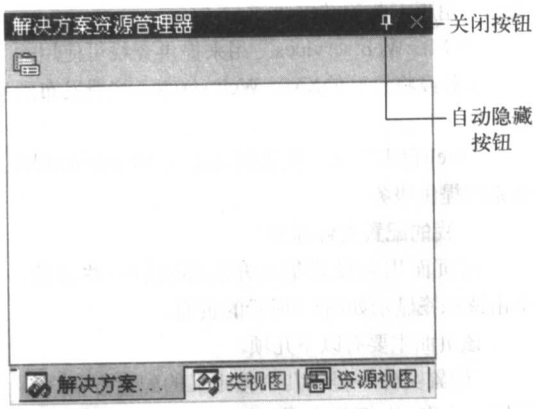

图1-3 窗口标题栏右侧的工具按钮

3. 自动隐藏

自动隐藏是Visual Studio.NET新增的界面特性，它和任务栏的自动隐藏相类似。自动隐藏的功能够使窗口显示的内容更多，凡是自动隐藏的窗口区，都会在屏幕靠左的侧边最小化，并只显示窗口名称标签，参看图1-1左边的“工具箱”。当用户将鼠标移动到这样的窗口名称标签时，该窗口就会自动滑出。当该窗口具有输入焦点时，它不会自动隐藏，一旦失去焦点，它又滑向屏幕的侧边，呈最小化状态。

需要说明的是，当用鼠标右击窗口的标题栏时，会弹出一个快捷菜单，其菜单命令依次为“可停靠”、“隐藏”、“浮动”、“自动隐藏”。这些命令的功能与“窗口”同名菜单命令的功能相一致。

1.1.4 起始页面

Visual Studio.NET的起始页面提供一个中心位置来设置集成开发环境的参数、阅读和编辑文档以及进行其他操作。在默认情况下，每次启动Visual Studio.NET时都会显示该页面，如图1-4所示。

起始页面上边是一些操作标签项，它指出能够在起始页面中进行操作的种类，每一个操作标签项对应于相应操作种类的详细操作。

1. 项目标签

项目标签页面是起始页面首先出现的页面，该页面列出一些最近打开的项目，可以单击这些项目链接来直接打开对应的项目，其作用和选择“文件”→“最近的项目”菜单命令一样。单击[新建项目]或[打开项目]按钮会打开相应的对话框，从中可以进行相应的具体操作。

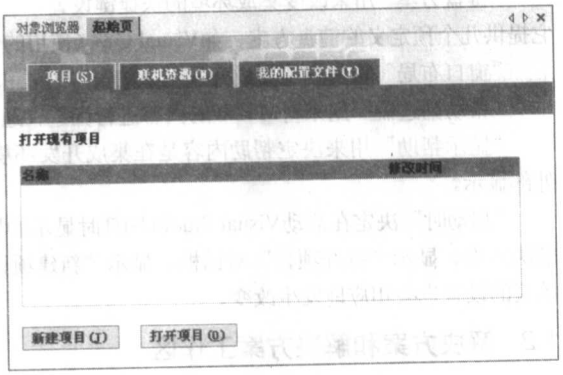

图1-4 起始页面

2. 联机资源标签

若本机已连接到Internet或当前处于联机状态，则单击该标签显示如图1-5所示的页面。

在页面左边是一些操作类型的列表:

“新增功能”是一些介绍Visual Studio.NET新特性以及其他新技术的超级链接。

“网上社区”列出一些和Visual Studio.NET相关的共享代码以及微软公司开发者新闻组等内容，这些新闻组包含大量讨论Visual Studio.NET的电子邮件。单击这些链接会打开Outlook Express，然后就可以订阅新闻组和收发电子邮件。

“标题新闻”用来链接访问Internet最新发布的MSDN消息。

“联机搜索”用来使用Microsoft的搜索引擎，在Internet中进行搜索。

“下载”用来提供Visual Studio.NET相关的免费和试用版下载等链接。

“XML Web services”用来提供查找可使用的Web服务或将自己的XML Web services注册发布到社区中。

“Web宿主”是一些提供商通过.NET企业级服务系统提供服务。

3. 我的配置文件标签

该页面用来设置集成开发环境的一些参数，单击该标签显示如图1-6所示的页面。

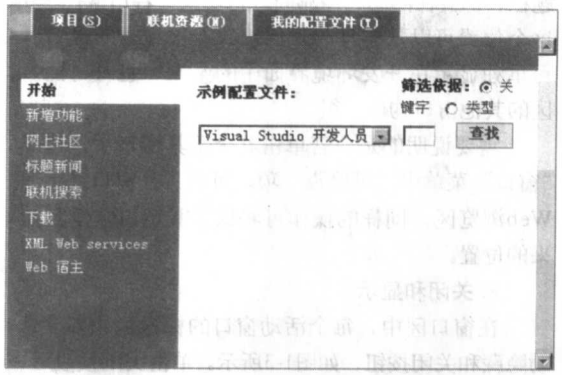

图1-5　联机资源标签页面

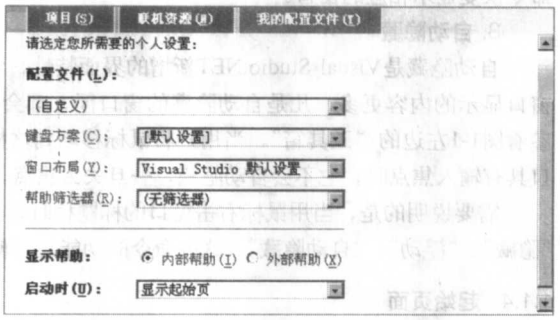

图1-6　我的配置文件标签页面

该页面主要有以下几项：

“配置文件”提供几个预定义的配置方案，包括Visual Studio开发人员、Visual Basic开发人员、Visual C++开发人员、Visual InterDev开发人员、VS宏开发人员、学生开发人员、Visual C#开发人员、Visual J#开发人员等，这些配置文件定义了适合各自环境的键盘方案、窗口布局和帮助筛选器。可以在这些配置文件中选择一个满足自己要求和使用习惯的配置方案，例如C++程序员一般会选择Visual C++开发人员配置方案，这个配置方案针对集成开发环境进行一些配置，使之更有利于C++程序的开发。

“键盘方案”用来改变集成环境的快捷键设置，它提供几个预定义的键盘方案，如Visual C++ 6使用的快捷键设置和Visual Studio默认设置等。

“窗口布局”用来调整集成开发环境的窗口组成和管理方式。

“帮助筛选器”用来对MSDN的内容进行筛选，通过设置帮助筛选器可以只显示自己感兴趣的内容。

“显示帮助”用来决定帮助内容是在集成开发环境内部显示还是通过独立的帮助程序在集成开发环境外部显示。

“启动时”决定在启动Visual Studio.NET时显示的界面类型。可选择的有显示起始页、加载最近加载的解决方案、显示“打开项目”对话框、显示“新建项目”对话框、显示空环境等。当改变该项设置时，起始页的设置也会相应地发生改变。

1.2　解决方案和解决方案工作区

在Visual Studio.NET中，解决方案工作区用来管理应用程序项目、组织文件、查看源代码和源文件，它主要由“解决方案资源管理器”、“类视图”和“资源视图”3个窗口组成。

在具体讨论这些操作和概念之前，先来创建一个.NET项目：选择“文件”→“新建”→“项目”菜单，在弹出的“新建项目”对话框中选择“Visual C++项目”节点下的.NET类型，这时在模板列表中显示出.NET所有的应用程序项目模板。选择“Windows窗体应用程序”模板类型（该类型用于创建可执行的Windows窗体应用程序）。单击[浏览]按钮将项目工作文件夹定位到自己的文件夹，如“C:\Visual C++.NET程序”，并在“名称”框中输入项目名Ex_1_Form1，结果如图1-7所示。单击[确定]按钮，系统开始创建，并回到Visual Studio.NET的主界面。

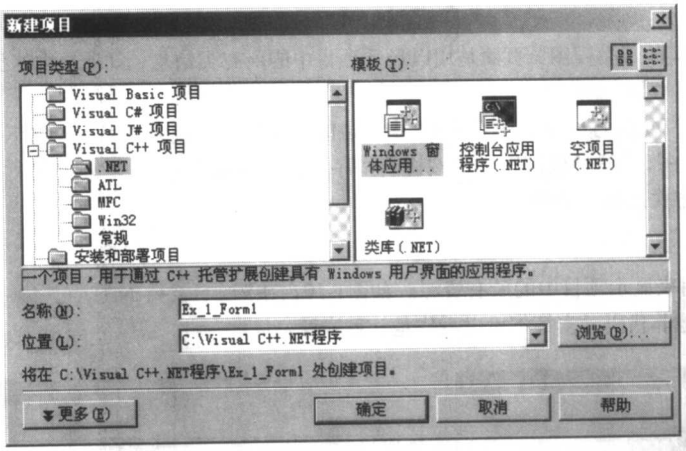

图1-7 “新建项目”对话框

1.2.1 解决方案基本概念

Visual Studio.NET使用解决方案这个概念来表示一个平时经常提及的应用程序项目，用户可以通过方案资源管理器对项目进行管理、组织文件、查看源代码及资源文件等。

从概念来说，解决方案是一个容器，它可以包含若干个应用程序及其相关项目。它的最大特点就是能管理与应用程序项目相关的多种类型的外部文件，如HTML、图标、菜单资源文件以及C++文件等，它的概念拓展了应用程序项目的内涵和外延。但要注意的是:

（1）Visual Studio.NET既可创建一个空的解决方案，也可以创建一个只含有一个应用程序项目或只含有一两个外部文件的解决方案。但当用户创建一个应用程序项目时，系统会自动将该应用程序项目添加到同名的解决方案中。例如，若用户创建一个Ex_1_Form1应用程序项目，那么该项目就会包含在Ex_1_Form1解决方案中。

（2）由于解决方案的提出，相关的应用程序项目文件的扩展名和组织也随之发生变化。以创建的.NET窗体应用程序Ex_1_Form1为例，Ex_1_Form1解决方案是保存在Ex_1_Form1文件夹中，其相应的扩展名为.sln（solution，解决方案）；而应用程序项目Ex_1_Form1的扩展名为.vcproj（Visual C++ Project）。

1.2.2 解决方案资源管理器

“解决方案资源管理器”是将方案中的所有文件以“树”的形式分类显示。这些文件包括所引用的项目文件、C++源文件（.cpp）、头文件（.h）、资源文件等，单击节点名称图标前的符号“+”或“-”或双击图标，将显示或隐藏节点下的相关内容。如图1-8所示。

图1-8 “解决方案资源管理器”页面

每一个解决方案项目文件在该页面中都有相应的“树节点”，例如，所有的C++源文件都在“源文件”节点中。用户不仅可以在树节点中移动文件，而且还可以创建新的树节点以及将一些特殊类型的文件放在该树节点中。

若创建一个新的树节点，可在根节点处右击鼠标，从弹出的快捷菜单中选择“添加”→“新建文件夹”命令，这时就会创建一个默认名为“NewFolder1”的节点。这个节点名称可根据自己需要重新命名。

从上可以得知，“解决方案资源管理器”是用户与解决方案之间的双向接口。它提供有关打开解决方案中各项的实时信息，并可让用户管理这些内容。例如，用户可以查看各文件的当前状态，允许用户将某些项从一个项目拖放到另一个项目，添加或删除项以及显示解决方案、项目或文件的属性等。

1.2.3 类视图

解决方案工作区的“类视图”页面是用以显示项目中的所有类信息，如项目中的类名、相关宏、常数、全局函数和变量等，如图1-9所示。

单击“类视图”页面上方的工具图标中右边的下拉按钮，就会弹出相应的下拉菜单，从中可以选择“按字母顺序排序”、“按类型排序”、“按访问排序”或“按类型分组”，来决定类视图页面节点的显示方式。默认的显示方式是按类型排序。

1.2.4 资源视图

“资源视图”用来显示项目中的所有资源，如图1-10。在Visual C++.NET中，每一个图片、字符串值、工具栏、图标或其他非代码元素等都可以看作是一种资源。

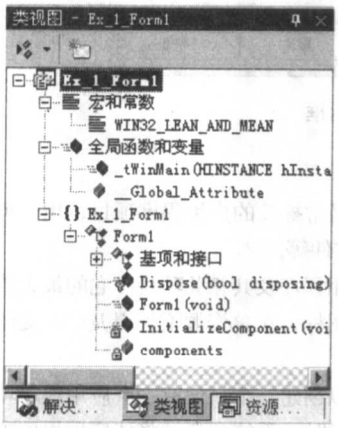

图1-9　“类视图”页面

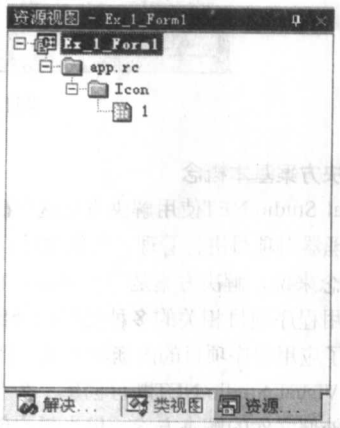

图1-10　“资源视图”页面

需要说明的是，在上述显示的页面中，项目中不同类别内容的节点名称前面的图标是不同的。例如在类成员函数的图标中，使用紫色方块表示公共成员函数，使用紫色方块和一把钥匙表示私有成员函数，使用紫色方块和一把锁表示保护型成员函数；又如用蓝绿色图标表示成员变量等。

1.2.5 设置解决方案属性

解决方案的属性设置既可以在其属性窗口中进行也可以在解决方案属性页对话框中完成。当单击“解决方案资源管理器”页面中的根节点“解决方案“Ex_1_Form1”(1 项目)”时，就会在“解决方案资源管理器”下面的属性窗口（默认位置）中显示其属性，如图1-11所示。若单击“解决方案资源管理器”页面顶部的“属性”按钮（），还将打开解决方案属性页对话框，如图1-12所示。

通过上述方式可以确定解决方案的启动项目、指定项目依赖特性、定义和编辑解决方案与项目的活动配置。

“启动项目”是指解决方案中首先被生成和运行的项目。在一个解决方案中，可以存在一个或多个启动项目。对于多个启动项目，可以在解决方案属性页对话框中的相应项目的“操作”列表中选择“开始执行（不调试）”、“启动”和“无”来确定该项目在启动调试器时是否执行以及是否调试。

“项目依赖项”是指某一个项目的生成依赖于其他项目，也就是说，在生成该项目之前必须先生成它所依赖的项目。

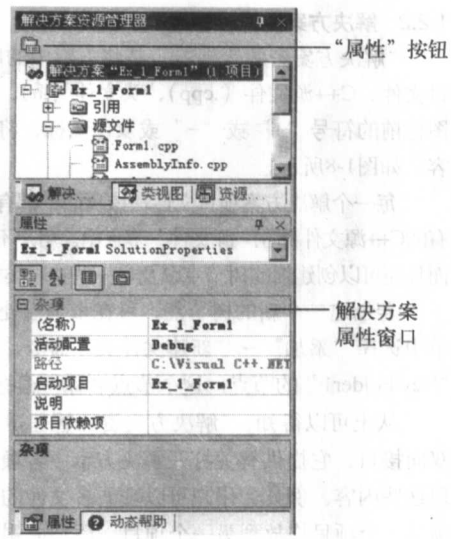

图1-11　解决方案属性窗口

“配置属性”用来指定如何生成解决方案和项目的调试版本（Debug）和发行（Release）版本。调试版本是为了便于进行调试，它所生成的程序中包含了大量的调试信息，而发行版本是用来生成最终的应用程序，它对所生成的代码进行充分优化，使生成的代码更小、速度更快。

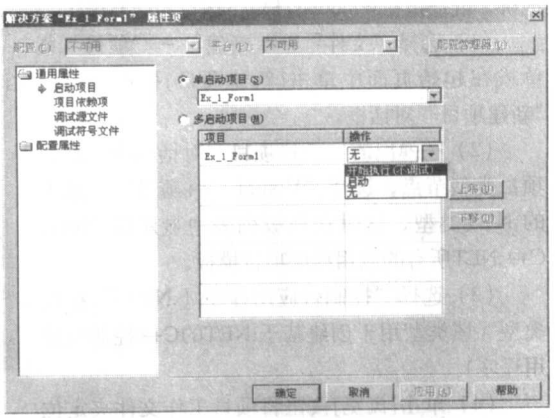

图1-12 解决方案属性页对话框

1.2.6 生成解决方案

生成解决方案时，可在菜单栏中先选择“生成”→“配置管理器”菜单，弹出如图1-13所示的对话框，从中可以选择当前活动的解决方案配置或在“配置”列中指定某一个项目的配置。选择要生成的版本，然后在“生成”菜单中选择相应的命令进行生成。若有多个项目需要生成，则可选择“批生成”菜单命令，弹出如图1-14所示的对话框，从中可以设定当前解决方案中所有项目的所有生成配置。选择后，单击[重新生成]按钮将重新开始生成所有选择的项目配置，而单击[生成]按钮则只生成那些更新过的项目配置。[清理]按钮用来删除选定的项目配置和生成过程的所有中间文件和输出内容。

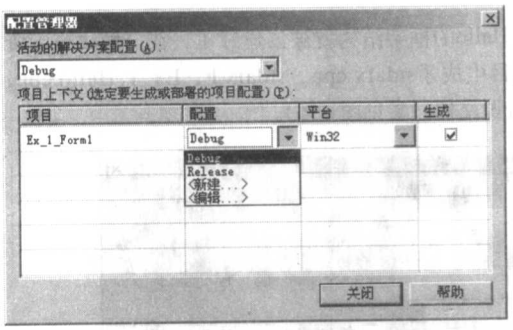

图1-13 “配置管理器”对话框

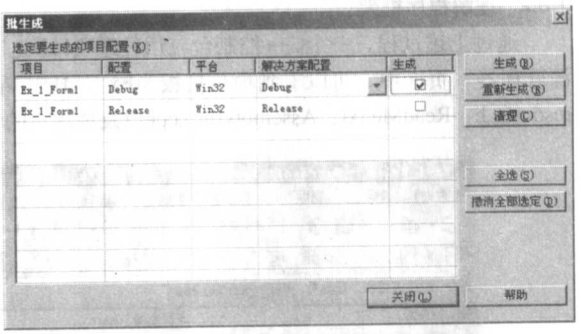

图1-14 “批生成”对话框

1.3 创建一个Visual C++.NET程序

前面介绍了许多关于开发环境的操作和概念，这里以C++控制台应用程序为例，进一步了解开发环境的使用过程，其步骤为：创建程序框架→添加或修改代码→编译并运行。

1.3.1 创建程序框架

在Visual C++.NET中，项目模板能快速创建一些常用的应用程序类型框架，如基于MFC的一般Windows应用程序、基于ATL的应用程序以及基于.NET的应用程序等。

MFC是早期的Windows应用程序开发方式，为了帮助用户处理各种经常使用又复杂繁琐的Windows操作，Visual C++设计了一套基础类库（Microsoft Foundation Class Library，简称MFC）。MFC把传统Windows API编程规范中的大多数内容封装成为各种类，使程序员从繁杂的编程中解脱出来，提高了编程和代码效率。

ATL是ActiveX Template Library的缩写，它是一套C++模板库。使用ATL能够快速地开发出高效、简洁的代码，同时对COM（Component Object Model，组件对象模型）组件的开发提供最大限度的代码自动生成以及可视化支持。为了方便使用，从Microsoft Visual C++ 5.0开始，微软公司把ATL集成到Visual C++开发环境中，1998年9月推出的Visual Studio 6.0集成了ATL 3.0，现在的Visual Studio.NET集成了ATL 7.0。

在Visual C++ .NET中，创建一个基于.NET的C++控制台应用程序可按下列步骤进行。所谓控制台应用

程序是指那些需要与传统DOS操作系统保持某种程序兼容，同时又不需要为用户提供完善界面的程序。简单地讲，就是指在Windows环境下运行的DOS程序。

（1）选择“文件”→“新建”→“项目”菜单或在起始页面中单击[新建项目]按钮，弹出“新建项目”对话框。

（2）在弹出的“新建项目”对话框中，展开项目类型节点，选择“Visual C++项目”节点下的.NET类型，这时在模板列表中显示出Visaul C++.NET所有的应用程序项目模板。

（3）选择“控制台应用程序（.NET）”模板类型（该类型用于创建基于.NET的C++控制台应用程序）。

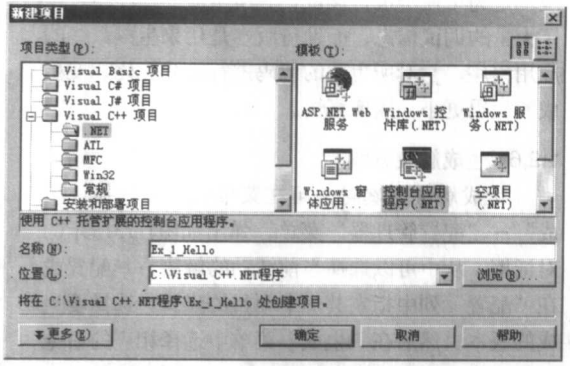

图1-15　“新建项目”对话框

（4）单击[浏览]按钮将项目工作文件夹定位到自己的文件夹，如“C:\Visual C++.NET程序”，并在“名称”框中输入项目名Ex_1_Hello，结果如图1-15所示。

（5）单击[确定]按钮，系统创建Ex_1_Hello项目。

1.3.2　理解程序框架

通过上述过程，一个C++控制台应用程序项目Ex_1_Hello的框架结构就被创建好了。在“解决方案资源管理器”页面中，可以看到项目模板为Ex_1_Hello项目生成了stdafx.cpp、stdafx.h、Ex_1_Hello.cpp、resource.h、ReadMe.txt、AssemblyInfo.cpp以及该项目的资源文件（参见图1-16）。

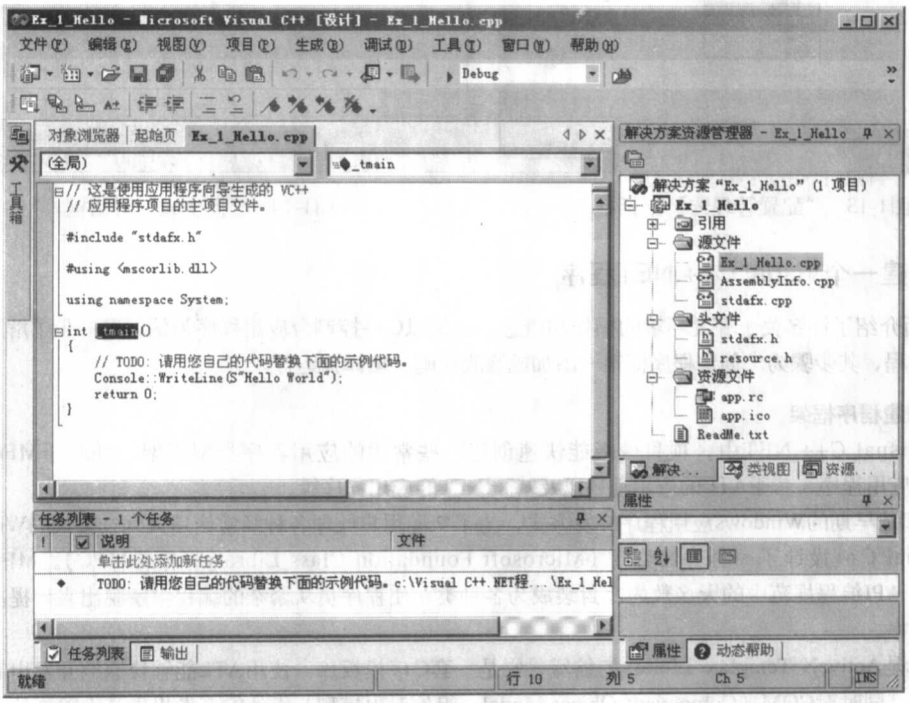

图1-16　Ex_1_Hello.cpp文件内容

其中，stdafx.cpp是一个只有一条语句（#include "stdafx.h"）的空文件；stdafx.h是Visual C++.NET为每个项目配置的用来预编译的文件，在stdafx.h文件中可以加入应用程序所需要的头文件；ReadMe.txt是

Visual C++.NET为每个项目配置的说明文件，它包括对项目模板产生文件类型的说明以及操作的一些技巧；resource.h是用来定义资源标识符的头文件；app.rc是项目Ex_1_Hello的资源文件；AssemblyInfo.cpp文件用来控制程序集的属性，程序集用来构成基本部署单元、版本控制、重新使用、激活范围和安全权限。

实际上，Ex_1_Hello.cpp才是项目模板产生的“真正”具有实际意义的程序源代码文件，用户几乎所有的代码都是添加在这个文件中。

在“解决方案资源管理器”窗口中，展开所有的“源文件”节点，双击“Ex_1_Hello.cpp”，该文件内容就会在开发环境左边的文档窗口显示出来，并用蓝色来表示代码中的关键字，用绿色表示注释，如图1-16所示。

代码“#using <mscorlib.dll>”中， #using关键字是预处理指令，用来将一个元数据文件输入到C++程序中，这些元数据文件可以是DLL、EXE、OBJ等。mscorlib.dll是.NET框架的一个核心类库。

代码“using namespace System;”中，using和namespace是Visual C++.NET关键字，用来使用System命名空间。命名空间是“类型”的一种逻辑命名方案，.NET使用该命名方案用于将类型按相关功能的逻辑类别进行分组，利用命名空间可以使开发人员更容易在代码中浏览和引用类型。System是.NET框架根命名空间，包含最基本的类型，约有100个类，如：用于数据流输入/输出的System::IO、用于支持GDI+的2D图形绘制的System::Drawing、用于文本处理的System::Text以及基于Windows窗体应用程序设计的System::Windows::Forms等，每个类型都包含若干个类。

代码中，_tmain函数是程序的入口点，它是用于UNICODE（双字节字符集，即用16 位大小来表示的字符集）的main主函数，每一个基于.NET的C++控制台应用程序都必须包含一个且只能包含一个这样的主函数。

主函数中的Console是System命名空间中的一个类，它用于控制台应用程序的标准输出和输入。WriteLine是Console类的一个成员函数，用来向控制台输出一行文本。域运算符“::”用来通知编译系统该函数所属的类。WriteLine（S“Hello World”）将“Hello World”字符串显示在控制台窗口中，S前缀用来指明后面的字符串是一个字符串常量。除WriteLine 外，Console类中类似的函数还有Write、Read、ReadLine。Write函数表示向控制台输出文本但不换行，Read和ReadLine分别表示从控制台读入若干字符和一行文本。

程序中的“/*.....*/”之间的内容或以“//”开始一直到行尾的内容是注释，其目的只是为了提高程序的可读性，对编译和运行不起作用。正是因为这一点，所以注释的内容既可以用汉字来表示，也可以用英文来表示，只要便于理解就行。需要说明的是，/*.....*/注释方式可以出现在程序中的任何位置。

1.3.3 添加并修改代码

现将_tmain函数中的代码改写如下，该程序要求输入两个整数，计算两数之和，并显示结果：

```
int _tmain()
{
    String  *strNum1,                          // 用户第一个输入
            *strNum2;                          // 用户第二个输入
    int     nNum1,                             // 第一个整型变量
            nNum2,                             // 第二个整型变量
            nSum;                              // 两数字之和
    Console::Write(S"请输入第一个数字: ");
    strNum1 = Console::ReadLine();             // 获取第一个数字(字符串)
    Console::Write(S"请输入第二个数字: ");
    strNum2 = Console::ReadLine();             // 获取第二个数字(字符串)

    // 将数字从String*转换成整型数字
    nNum1 = Int32::Parse(strNum1);
    nNum2 = Int32::Parse(strNum2);

    // 求两个数之和
    nSum = nNum1 + nNum2;
    // 显示结果
```