

第1章 Visual Studio 2005 概述



本章学习要点

- 了解 Visual Studio 2005 的特性;
- 了解控制台应用程序的创建方法。

1.1 C#基础知识

在过去的 20 年里，C 和 C++ 已经成为商业软件开发领域中使用最广泛的语言。它们为开发人员提供了十分灵活的操作，但同时也牺牲了一定的效率。与 Microsoft Visual Basic 等其他语言相比，同等级别的 C/C++ 应用程序往往需要更长的时间来开发。由于 C/C++ 语言的复杂性，许多开发人员都试图寻找一种新的语言，希望能在功能与效率之间找到一个更为理想的权衡点。

对于 C/C++ 用户来说，最理想的解决方案无疑是在快速开发的同时又可以调用底层平台的所有功能。为了解决这一问题，Microsoft 推出了 C#。

1.1.1 C#简介

C# 是一种现代的面向对象语言，它使开发人员能够快速便捷地创建基于 Microsoft .NET 平台的解决方案。这种框架使 C# 组件可以方便地转换为 XML 网络服务，从而使任何平台的应用程序都可以通过 Internet 调用它。

C# 增强了开发人员的效率，同时也致力于消除编程中可能导致严重后果的错误。C# 使 C/C++ 开发人员可以快速进行网络开发，同时也保持了开发人员所需要的强大性和灵活性。

1.1.2 .NET Framework 平台体系结构

C# 程序在 .NET Framework 上运行，.NET Framework 是 Windows 的一个必要组件，包括一个称为“公共语言运行时”(CLR)的虚拟执行系统和一组统一的类库，CLR 是 Microsoft 的公共语言基础结构 (CLI) 的一个商业实现。CLI 是一种国际标准，是用于创建语言和库在其中无缝协同工作的执行和开发环境的基础。

.NET Framework 提供了一个统一、面向对象且层次化的可扩展的类库集。Visual Studio.NET 中的所有编程语言，包括 Visual C++、Visual C#、Visual J#、Visual Basic 及各种脚本语言都可以使用这个类库，C# 中的各种数据类型就建立在这个类库之上。

1.1.3 面向对象的程序设计方法

面向对象的程序设计方法是当前程序设计的大势所趋，这种设计方法通过类、对象、继承

和多态等机制形成了一个完善的编程体系。面向对象编程（OOP）将程序设计中的数据和对数据的操作作为一个不可分割的整体，通过由类生成的对象来组织程序。对象包含属性与方法，能识别和响应一定的事件。

1. 类和对象

现实生活中的类是人们对客观对象不断认识而概括出来的抽象概念，而对象则是现实生活中的一个个实体。例如，学生是正在学校接受教育的一群人的总称，即一个类。而对象则具体到一个个不同的人，如张三、李四。

在实际的编程过程中，每当创建一个新对象时，它必须基于一个类。例如，要创建一个按钮，则必须基于一个类，放置在窗体上能使用的按钮就是一个对象，称为“类的一个实例”。

2. 属性

属性包含一些与对象有关的状态信息，如名称、颜色或位置等。C#中的对象属性可以看做表现对象特征的数据扩展。

例如，在面向对象的编程中，控件对象的常用属性有名称、文本及是否可见等，修改这些属性的值可以改变控件的状态。

3. 事件

事件就是预先定义并能够被对象识别的动作。当用户执行一个操作，如单击一个按钮时产生一个鼠标单击事件，在键盘上按一个键时产生一些击键事件等。

4. 方法

与对象有关的行为、操作称为“方法”，它是面向对象编程的动词。一些典型的方法有 Close、Show 和 Clear 等。例如，学生类可以定义学习、参加考试等方法。

1.2 Visual Studio 2005 的新特性

Visual Studio.NET 是 Microsoft 公司推出的最新程序开发工具，和以前的 Visual Studio 相比具有质的飞跃。而 Microsoft Visual Studio 2005 则在此基础上有了进一步的发展。Microsoft Visual Studio 2005 支持新的应用程序开发平台，即 Microsoft .NET Framework 2.0 版本，并把 Microsoft 公司所有语言的开发环境整合在一起，即所有语言使用同一套工具，在同一个集成开发环境（IDE）中进行开发，并且改进环境中所包含的各种特性。Visual Studio 2005 有以下新特性。

1) 创新的四种语言

各有特色和创新的四种语言（Visual Basic、Visual C++、Visual C#和 Visual J#）使编程体验更加丰富。

2) IDE 的改进

IDE 的改进提供了个性化，是开发的高生产力的保证。Visual Studio 2005 的 IDE 性能稳定且强大，主要特征有统一的语言开发环境；更方便使用的工具窗口，如资源管理窗口；使用标签化窗口极大地提高了屏幕利用率，当要切换窗口时，只需单击窗口标签即可。IDE 中的各种窗口还具有自动隐藏的特性。

3) .NET Framework 2.0

从 .NET Framework 1.0 平台升级到 .NET Framework 2.0 平台，.NET Framework 2.0 是一个可靠的应用平台，为构建安全、高性能及关键业务的解决方案提供了坚实基础。



4) SQL Server 2005 解决方案的开发

SQL Server 2005 集成到 Visual Studio 2005 的 IDE 中,可以对数据库对象进行开发和调试。

5) 企业开发和软件生命周期管理

Visual Studio Team System 可以使软件开发团队在开发过程中降低工作复杂度,是提高生产和协作能力的高效率、集成性、可扩展的软件生命周期管理工具。

1.3 安装 Visual Studio 2005

在 Windows XP 操作系统中安装 Visual Studio 2005 的步骤如下。

(1) 将 Visual Studio 2005 的安装光盘放入光驱,安装程序将进入安装启动界面,选择“安装 Visual Studio 2005”选项。

(2) 如果该计算机是第一次安装 Visual Studio 2005,那么在安装前首先检测是否安装所需组件。如果没有满足基本要求,则提示需要安装组件。单击其中的“Windows XP Service Pack2”链接,开始安装过程。

(3) 所有组件安装完成后,在打开的窗口中单击“下一步”按钮开始安装 Visual Studio 2005。

(4) 输入 Visual Studio 2005 的产品密钥。选中“我接受许可协议中的条例”复选框,然后单击“下一步”按钮。

(5) 在打开的窗口中,用户可以选择要安装的功能。选中“默认值”单选按钮,使用默认设置安装;选中“完全”单选按钮将全部安装;选中“自定义”单选按钮可以选择需要安装的选项,它适用于高级用户。如果是初学者,建议选中“默认值”单选按钮;如果硬盘足够大,可以选中“完全”单选按钮。在该界面中,还允许选择安装路径。Visual Studio 2005 默认的安装路径为“C:\program Files\Visual Studio8\”。一般不建议将 Visual Studio 2005 安装在 C 盘中,单击其后的“浏览”按钮可以改变安装的路径。可以直接在“路径”文本框中输入路径,完成所有设置后单击“安装”按钮。

安装 Visual Studio 2005 时系统会自动安装所有程序,安装时间由于机器的性能不同而有所区别,在安装过程中会重启多次。

完成安装后重新启动计算机,安装结束。

1.4 IDE 介绍

Visual Studio 2005 提供了具有丰富工具的 IDE,它拥有强大的功能。用户可以使用它快速有效地创建各种 C#项目,项目中甚至可以包含来自不同语言的模块。

Visual Studio 2005 还是一个自动化程序很高的系统,用户可以定制符合自己习惯的开发环境,如定义工具栏,其中包括工具栏的布局和按钮等。

1.4.1 开始页面

Visual Studio 2005 的“起始页”页面为用户提供了一个中心位置来设置 IDE 参数,阅读文档并执行其他操作。在默认情况下,每次启动 Visual Studio 2005 都将进入如图 1-1 所示的页面。

左边窗格的中上部显示了最近打开过的项目,单击这些项目链接可以直接打开对应项目。可以打开已经有的项目(网站)或者创建一个新的项目(网站),单击链接可以执行相应的操作。

右边窗格中显示了 Visual Studio 2005 的新特性等内容，单击其中的链接可以查看相应的内容。

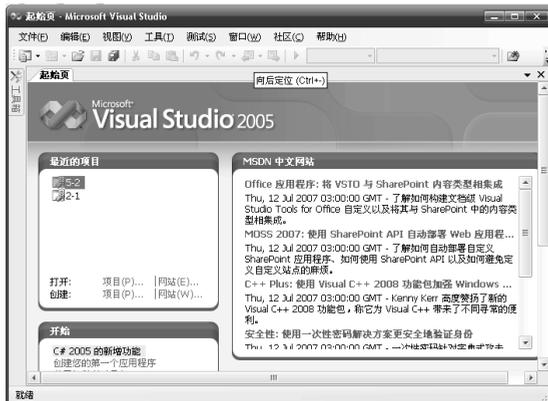


图 1-1 “起始页”页面

1.4.2 解决方案资源管理器

解决方案资源管理器是用户和解决方案之间的一个接口，它提供了有关解决方案中对象的实时信息，并且允许用户管理这些对象。用户可以在该窗口中完成许多文件和项目的管理任务，例如，通过拖动来复制或移动对象、删除对象、生成解决方案和项目、设置启动项目、添加项目、添加项、打开文件，以及查看对象的属性等。

1. 创建新的解决方案资源管理器

Visual Studio 2005 使用解决方案和项目来组织应用程序所包含的各种文件，一个解决方案可以包含一个或多个项目。

为创建新的解决方案和项目，选择【文件】 【新建】 【项目...】选项，打开如图 1-2 所示的“新建项目”对话框。在其中选择要建立的项目类型以及相对应的模板，然后在“名称”和“位置”文本框中输入新项目的名称和保存的位置。这里要注意选中“创建解决方案的目录”复选框，单击“确定”按钮。



图 1-2 “新建项目”对话框

2. 解决方案资源管理器

创建一个项目后进入 Visual Studio 2005 开发主界面（例如，创建一个 Window 应用程序，项目名称为“HelloWorld”），在右上角可以看到解决方案资源管理器。使用 Visual Studio 2005 开



发的所有应用程序都是通过解决方案和项目来进行组织和管理的，而解决方案和项目的管理操作都可以通过解决方案资源管理器来完成。

解决方案资源管理器的工具栏中有以下几个按钮。

- (1) “属性”按钮：单击该按钮，显示查看选中项目的属性。
- (2) “显示所有文件”按钮：单击该按钮，显示所有文件。
- (3) “刷新”按钮：单击该按钮，刷新解决方案资源管理器中的文件。
- (4) “查看代码”按钮：单击该按钮，查看该文件中的代码。
- (5) “查看类关系图”按钮：单击该按钮，显示创建的类之间的关系。

3. 添加/删除项

添加到方案和项目中的项目可以是任何类型的文件，包括源代码文件及文本文件等。能够添加新项目的位置有解决方案节点、项目节点及其下的文件夹节点，选择的添加位置不同，可以添加的文件类型也会不同。用户可以创建新的文件，也可以添加已有的文件。要在解决方案或项目中添加项，可执行以下步骤。

(1) 选择要添加项的节点，然后选择【项目】 【添加新项】选项或者【项目】 【添加现有项】选项打开相应的对话框。在这里选择【项目】 【添加新项】选项，打开“添加新项目”对话框。

(2) 选择新创建的文件类型，选择默认的一类文件即可。输入新文件的名称“text”，单击“添加”按钮，添加一个新项。在解决方案资源管理器中可以查看此文件。

(3) 右击要删除的项，在弹出的快捷菜单中选择【删除】选项将删除项。

4. 设置项目的属性

在 Visual Studio 2005 中，项目的属性可以分为两类，即配置无关属性和配置相关属性。常见的项目配置有两种，即为调试应用程序和为最终发行程序设置的配置。

无关属性就是哪些影响所有项目配置的属性，而相关属性则指那些只影响指定项目配置的属性。配置无关属性在“属性”对话框中实现，通常包括项目的名称、保存位置，以及项目所对应的策略文件等。

配置相关属性则在项目属性页对话框中实现，通常包括程序集的名称、启动对象及输出类型等属性。在这个对话框中，用户可以针对不同的项目配置不同的属性值。例如，针对调试配置和发行配置设置不同的优化属性。要打开项目属性页对话框，在解决资源管理器中选择一个项目节点，然后单击解决方案资源管理器工具栏中的“属性”按钮即可。

用户可以根据需要设置其中的选项，在选择项目模板创建项目时，Visual Studio 2005 将根据配置无关属性及相关属性为用户设置各属性的值，所以用户不需要过多设置。

1.4.3 菜单栏

Visual Studio 2005 开发主界面菜单栏中包含了 Visual Studio 2005 的大多数功能，其中的菜单随不同项目及不同文件动态变化。

菜单栏提供了用于开发、调试和保存应用程序需要的所有选项，除了提供标准的【文件】、【编辑】、【视图】、【窗口】和【帮助】菜单项之外，还提供了编程专用的功能菜单项，如【项目】、【生成】及【调试】等。

- (1) 【文件】菜单项：包括文件操作的全部选项，如新建工程、打开工程及保存工程等。
- (2) 【编辑】菜单项：包括正文编辑和控件的操作选项。

(3)【视图】菜单项：包括显示或隐藏 IDE 中的各种窗口，如代码编辑器、解决方案资源管理器和输出窗口等。

(4)【项目】菜单项：用于添加各种项目，如添加 Windows 窗体、用户控件、组件、类和新项等。

(5)【生成】菜单项：主要用于生成解决方案。

(6)【调试】菜单项：包括常用程序查错的工具及程序启动、全部中断和全部终止等选项。

(7)【数据】菜单项：用于在程序中添加新数据源，连接到数据库。

(8)【工具】菜单项：包括连接到设备、连接到数据库、连接到服务器、选择工具箱，以及外接程序管理等选项。

(9)【测试】菜单项：主要用于测试，包括新建测试、加载源数据文件和创建新测试列表等选项。

(10)【窗口】菜单项：包括控制窗口布局的选项。

(11)【帮助】菜单项：包括获取帮助信息的选项。

1.4.4 工具栏

菜单栏下面是标准工具栏，其中提供了多个常用命令的快速访问按钮。单击某个按钮，即可执行相应的操作。和菜单栏一样，Visual Studio 2005 的工具栏也是动态变化的。随着工作的不同，工具栏中的按钮也不尽相同。

工具栏中的主要按钮如下。

(1)“新建项目”按钮：新建项目、网站及文件等。

(2)“添加新项”按钮：在解决方案中添加项目及网站等。

(3)“打开文件”按钮：打开项目及文件等。

(4)“保存”按钮：保存当前打开的项目或文件。

(5)“全部保存”按钮：保存所有未保存文件。

(6)“剪切”按钮：剪切选中内容到剪贴板中。

(7)“复制”按钮：复制选中内容到剪贴板中。

(8)“粘贴”按钮：粘贴剪贴板中的内容。

(9)“撤销”按钮：撤销上次操作。

(10)“重复”按钮：重复上次操作。

(11)“向后定位”按钮：回到后一个操作的文件。

(12)“向前定位”按钮：回到前一个操作的文件。

(13)“启动调试”按钮：开始调试程序。

(14)“在文件中查找”按钮：在文件中实现查找和替换功能。

(15)“解决方案资源管理器”按钮：打开解决方案资源管理器。

(16)“属性面板”按钮：打开“属性”面板。

(17)“对象浏览器”按钮：打开对象浏览器。

(18)“工具箱”按钮：打开工具箱。

(19)“起始页”按钮：打开起始页。

(20)“其他窗口”按钮：打开其他窗口。

1.4.5 工具箱

工具箱是 Visual Studio 2005 中的重要工具，每一个开发人员都必须非常熟悉，其中提供了开发



Windows 窗口应用程序所必需的控件。通过工具箱，开发人员可以方便地进行可视化窗体设计。

如图 1-3 所示为工具箱的外观，展开其中的“所有 Windows 窗体”标签，可以看到所有 Windows 窗体控件。由于 Windows 窗体控件过多，将在本书的第 10 章中详细介绍工具箱。



图 1-3 工具箱外观

1.4.6 代码编辑器

“代码编辑器”是用来显示和编辑程序代码的窗口，支持 Visual Studio 2005 中所有的编程语言，它提供了强大的代码编辑功能。在 Visual Studio 2003 的基础上，Visual Studio 2005 又提供了以下新增功能。

- (1) 自动换行。
- (2) 渐进式搜索。
- (3) 向后/向前定位按钮。
- (4) 剪贴板循环。
- (5) 代码的大纲显示。
- (6) 折叠到定义。
- (7) 折叠块/全部折叠。
- (8) 智能感知功能。
- (9) 为行编号。
- (10) 超链接。
- (11) 编码问题指示器。

可以使用以下方法打开“代码编辑器”窗口。

- (1) 单击解决方案资源管理器窗口中的“查看代码”按钮。
- (2) 选择【视图】 【代码】选项或者按 F7 快捷键。
- (3) 右击要添加代码的项，在弹出的快捷菜单中选择【查看代码】选项，如图 1-4 所示。
- (4) 如果创建的是 Windows 应用程序，则可以双击窗体的任何地方或者窗体中的控件，打开“代码编辑器”窗口。
- (5) 如果创建的是 Windows 应用程序，则右击窗体，在弹出的快捷菜单中选择【查看代码】选项。

打开“代码编辑器”窗口后，用户可以在其中输入或修改代码。代码编辑器如图 1-5 所示。



图 1-4 【查看代码】选项

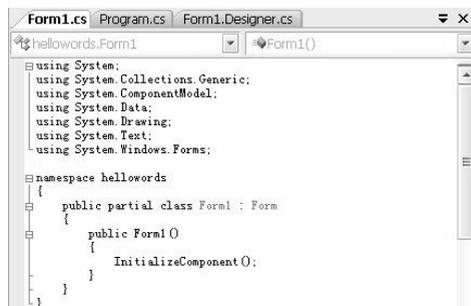


图 1-5 代码编辑器

在代码编辑器的上方有两个下拉列表。

(1)“类型”下拉列表：其中包括当前窗体及其包含的所有类的对象名。

(2)“成员”下拉列表：其中包括所选类的所有成员。

代码编辑器可以分为如下 3 个区域。

(1) 代码编辑区：用户输入或修改代码的区域，在其中可以使用编辑器提供的各种功能。例如，使用不同的颜色显示代码的不同部分、自动换行功能，以及智能感知功能等。

(2) 选定内容边距：选定内容边距用来选择代码内容，当鼠标指针移动到该边距内时，指针形状变成箭头形状。单击可选定整行代码，拖动可以选择相邻的代码。

(3) 指示器边距：除了显示一些标识（如断点、书签和快捷方式标识）外，还可以在其单击操作中设置断点。

1.4.7 对象浏览器

Visual Studio 2005 为用户提供了对象浏览器，它允许用户查看不同组件中的对象（包含命名空间、类、结构、接口、类型和枚举等）和对象的成员（包含属性、方法、事件、变量、常量和枚举项等）。可以查看的对象组件不仅仅是用户解决方案中的项目，还可以是用户项目所引进的组件及外部组件，如 .NET 框架组件和 COM 组件。

要打开对象浏览器，可按 Ctrl+Alt+J 组合键，或选择【视图】 【对象浏览器】选项。

除了浏览对象及其成员的信息外，还可以在对象浏览器中查看对象或成员的定义或声明代码，指定所要浏览的组件并查找特定的符号。

对象浏览器可以分为以下 4 个部分。

1) 工具栏

例如，使用其中的“前进/后退”按钮，可以回到曾经浏览过的位置。使用“浏览”下拉列表可以选择浏览范围，浏览范围指在对象浏览器窗口中显示和查找那些组件的对象。可以选择两种浏览范围，即活动项目和选择的组件。

2)“对象”窗格

该窗格位于对象浏览器的左边，显示当前浏览范围中的组件的容器对象，如命名空间、类、结构、接口、类型和枚举等。

3)“成员”窗格

该窗格位于对象浏览器的右边，显示在“对象”窗格中选中的对象成员，这些成员可以包括属性、方法、事件、变量、常数和枚举项。

4)“说明”窗格

该窗格位于“对象浏览器”的底部，显示当前选中的对象或成员的详细信息。

1.4.8 “属性”面板

使用“属性”面板可以查看和更改位于编辑器和设计器中选中对象的设计属性及事件，也可以编辑和查看文件、项目及解决方案属性。用户可以选择【视图】 【属性】选项或者按 F4 快捷键打开“属性”面板。该面板中显示的内容会随着用户的选择而动态改变，从而及时反映所选对象的属性。

“属性”面板中显示了编辑字段的不同类型，这取决于特定属性的需要。这些编辑字段包括编辑框、下拉列表，以及到自定义编辑器对话框的链接，属性以灰色显示则是只读的。

“属性”面板由以下几部分组成。

(1) 对象名称下拉列表：其中列出当前选中的一个或多个对象，选择一个对象可以显示其



属性；选择多个对象时，只显示所有选定对象的通用属性。

(2) 工具栏：可以使用工具栏中的按钮设置“属性”面板的显示方式和显示内容，即按分类顺序及字母顺序。按分类顺序是指按类别列出选定对象的所有属性及属性值，可能折叠类别以减少可见属性数。展开或折叠类别时，可以在类别名左边看到加号(+)或减号(-)，类别按字母顺序列出；按字母顺序则是指按字母顺序排列出选定对象的所有设计时的属性和事件。

单击属性按钮显示对象的属性，单击事件按钮显示对象的事件。仅当窗体或控件设计器在一个 Visual C#项目托管扩展的上下文中处于活动状态时，此“属性”面板中的工具栏控件才可用。此外，还可以单击属性页按钮打开选定项的“属性页”对话框。

(3) 显示属性部分：用来显示对象的属性及事件等。

(4) “说明”窗格：显示属性的属性类型和简短说明。

1.4.9 “命令”窗口

选择【视图】 【其他窗口】 【命令窗口】选项或者按 Ctrl+Alt+A 组合键，打开命令模式的“命令”窗口。

在“命令”窗口中可以直接输入并执行各种命令，其中包括菜单中或未在菜单中出现的命令，该窗口可以使 C#开发人员在开发程序时提高效率。

1.4.10 “即时”窗口

在“命令”窗口中输入【immed】命令将打开“即时”窗口，也可以选择【调试】 【窗口】 【即时】选项或者按 Ctrl+G 组合键打开“即时”窗口。

“即时”窗口用于调试应用程序，在其命令行中输入“?”后跟表达式或变量，按 Enter 键可以显示其值。

1.4.11 “任务列表”窗口

选择【视图】 【其他窗口】 【任务列表】选项或者按 Ctrl+Alt+K 组合键打开“任务列表”窗口。

“任务列表”窗口的作用是帮助用户组织和管理应用程序开发过程中的日常操作任务，其中显示的任务大体上可以分为两类，即用户输入的任务项和 IDE 自动生成的任务项。用户可以在“任务列表”窗口中记录一些以后打算执行的任务，在完成任务之后，可以为其做上标记或者直接删除。

单击“任务列表”窗口中的“创建用户任务”按钮，可以添加一些提示性的任务，该窗口中显示的每一个任务项都包含优先级、类型图标、检查框和说明信息等。

1.5 使用命名空间

对于一个大型项目而言，通常有更多的名称、更多的已命名数据、更多的已命名方法，以及更多的已命名类。如果要有效地管理它们，则需要使用 C#提供的命名空间(Namespace)。

所有的 C#类都位于一个命名空间中，C#语言使用命名空间组织系统类型或用户自定义的数据类型。如果没有明确地声明一个命名空间，则用户代码中定义的类型将位于一个未命名的全局命名空间中。从管理的角度上来看，可以把用户代码中定义的类型比做操作系统中的文件，而把命名空间作为文件夹，这个全局命名空间中的类型对于所有的命名空间都是可见的。



不同命名空间中的类型可以具有相同的名称，但是同一个命名空间中的类型名不能相同。用户在编写 C#语言程序时，通常要先声明一个命名空间，然后在这个命名空间中定义自己的类型。命名空间的声明非常简单，其语法如下所示。

```
namespace 名称[.名称] {
    定义自己的类型
    .....
}
```

例如，在一个名为“ HelloWorld ”的命名空间中创建一个名为“ HelloWorld ”的类，代码如下所示。

```
namespace HelloWorld
{
    class HelloWorld { ..... }
}
```

要使用某个命名空间中的类型时，可以使用 using 关键字来指定命名空间，例如：

```
using System;
using System.Collections.Generic;
using System.Text;
```

using 语句说明了即将使用的类所属的命名空间，因此在后面的代码中无需在每一个类前面说明这个类所在的命名空间了。由于上述 3 个命名空间之中包含了使用较为普遍的类，所以每次新建一个项目时，Visual Studio 2005 都会自动添加这些 using 语句。

表 1-1 列出了 .NET 框架中常用的命名空间。

表 1-1 .NET 框架中常用的命名空间

命名空间	类的描述
System	定义通常使用的数据类型和数据转换的基本 .NET 类
System.Collections	定义列表、队列和数组和字符串表
System.Data	定义 ADO .NET 数据库结构
System.Drawing	提供对基本图形功能的访问
System.IO	允许读写数据流和文件
System.Net	提供对 Windows 网络功能的访问
System.Net.Sockets	提供对 Windows 套接字的访问
System.Runtime.Remoting	提供对 Windows 分布式计算平台的访问
System.Security	提供对 CLR 安全许可系统的访问
System.Text	编码 ASCII、Unicode、UTF-7 和 UTF-8 字符
System.Threading	多线程编程
System.Timers	在指定的时间间隔内引发一个事件
System.Web	浏览器和 Web 服务器功能
System.Windows.Forms	创建使用标准的 Windows 图形接口的基于 Windows 的应用程序
System.Xml	提供对 XML 文档的支持

1.6 Main 方法

Main 方法是一个特殊的方法，在整个程序中只存在一个名为“ Main ”的方法。该方法是程



程序的入口点，程序将在此创建对象和调用其他方法。代码如下所示。

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Project_1-1
{
    class Program
    {
        static void Main(string[] args) //Main 方法
        {
            Console.WriteLine("HelloWorld!");
        }
    }
}
```

其中 `Console.WriteLine("HelloWorld!");`即为 Main 方法要运行的代码，即在控制台屏幕上显示“HelloWorld!”。

如果在上面的程序中添加另一个 Main 方法，那么在编辑时，Visual C# 2005 将显示错误提示。Main 方法是一个特殊的方法，它具有如下特点。

(1) 程序的入口，程序控制在该方法中开始和结束。

(2) 在类或结构（将在后面章节中讲解）的内部声明中，Main 方法必须是静态的，即必须声明为 static，而且可以具有 void 或 int 返回类型。声明 Main 方法时可以使用或不使用参数。

1.7 使用帮助

Visual Studio 包含一个广泛的 Help 帮助工具，其中包含所有的 Microsoft Developer Network (MSDN) 库，MSDN 中包含了许多 VS IDE、.NET Framework 及 C#等参考资料，能够很好地使用帮助中的知识来解决编程过程中的实际问题是一个优秀开发人员应该具备的素质。MSDN 可以在安装 Microsoft Visual Studio 2005 时一起安装，也可以单独安装。在 Microsoft Visual Studio 2005 集成环境中，选择【帮助】选项启动 Microsoft Visual Studio 2005 帮助，在其中可以获得大量的资料。也可以通过访问 Internet 上的 MSDN 来获得相关信息，网站地址为 <http://msdn2.microsoft.com/zh-cn/default.aspx>。使用帮助的缺点是查找想要的主题需要花费很长的时间。

1.8 项目实践

项目名称：创建控制台应用程序。

项目内容：创建控制台应用程序，并在控制台上输出“Hello World”。

项目目的：

(1) 熟悉 Microsoft Visual Studio 2005 集成环境；

(2) 掌握控制台输入方法。

项目步骤：

(1) 选择“程序” “Microsoft Visual Studio 2005” “Microsoft Visual Studio 2005”选



项，默认情况下进入“起始页”页面。选择【文件】 【新建项目】选项。

(2) 打开“新建项目”对话框。在左边“项目类型”中选择“其他语言” “Visual C#”选项，在右边“模板”部分选择“控制台应用程序”，然后在“名称”文本框中输入文件名为“HelloWorld”。

(3) 选择【文件】 【全部保存】选项，打开“保存项目”对话框。在“位置”文本框中输入程序放置路径，注意如果此文件夹不存在，系统将会自动创建该文件夹。确定选中“创建解决方案的目录”复选框，单击“确定”按钮，这样就保存了该项目的所有文件。

(4) 进入开发环境后，选择【视图】 【解决方案资源管理器】选项，打开解决方案资源管理器。检查其中列出的文件，可以看到解决方案名为 HelloWorld，它只有一个项目。项目下面包括两个文件，分别是 Program.cs 和 AssemblyInfo.cs。

(5) 打开 Program.cs 的代码编辑器，在其中显示程序自动生成的一些代码。在 Main 方法中加入 Console.WriteLine("Hello World!")。

代码如下。

```
using System;
using System.Collections.Generic;
using System.Text;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

1.9 复习与提示

本章介绍了 C#和.NET Framework 的基础知识，以及如何安装 Microsoft Visual Studio 2005。通过创建一个 C#程序“Hello World”，说明如何在应用程序中与用户进行交互、如何通过 System 预定义的类 Console 提供的方法来输入输出、如何编辑 C#源文件，以及 C#应用程序的基本组成部分。

1.10 上机实验

【实验】创建控制台应用程序

【实验要求】

创建控制台应用程序，并在控制台上输出“Hello World”；分别打开、关闭和移动“解决方案资源管理器”、“属性”、“代码设计器”等窗口；恢复 IDE 为默认窗口选项。

【实验目的】

熟悉 Microsoft Visual Studio 2005 开发环境。