

第 1 章 C#语言开发环境

【教学提示】 Visual Studio.NET 是一个集成开发环境，从程序设计、代码编译和调试，到最后形成发布程序的全部工作都可以在这个集成环境中完成。为了帮助用户快速、高效地开发应用程序，Visual Studio.NET 提供了许多功能强大的开发工具和丰富的菜单命令。本章将对 Visual Studio.NET 中的 C#集成环境的各个组成部分做概要的描述。

【教学要求】 在学习 C#语言之前，不仅要清楚 Visual Studio.NET 的总体构成，了解其具有的工具，还要知道其集成环境。本章主要介绍 Visual Studio.NET 集成环境的一些基础知识。通过本章学习，读者应该了解 C#语言的特点及其发展趋势、C#语言简介、C#的应用领域；了解 Visual Studio.NET 具有的工具，熟悉其集成环境；掌握安装 Visual Studio 2010 的硬件要求和安装步骤；熟悉 Visual Studio 2010 开发环境，包括编辑器和 Windows 窗体设计器、解决方案资源管理器和项目设计器、编译器、调试器和错误列表窗口、工具箱和属性窗口等；学会使用 Visual Studio 2010 帮助系统；掌握 C#程序设计过程。

1.1 C#语言概述

随着微软.NET 平台的逐步推广，C#作为一种基于该平台的面向对象编程语言，自面世以来便以其易学易用、功能强大的特点得到了广泛的应用，赢得了越来越多的程序开发人员的喜爱；它与 C++、Java 语言类似，但又有所改进，是一种简单、现代、高效、面向对象、类型安全的新型编程语言，开发人员可以通过它编写在 .NET Framework 上运行的各种安全可靠的程序，不但可以开发数据库管理系统，而且也可以开发多媒体应用程序和网络应用程序。而 Visual Studio 开发平台则凭借其强大的可视化用户界面设计，使程序员从复杂的界面设计中解脱出来，让编程成为一种享受。本书中涉及的程序都是通过 Visual Studio 2010 开发环境编译的。

1.1.1 C#语言的特点

C#是一种面向对象的编程语言，主要用于开发可以在 .NET 平台上运行的应用程序。C#语言是从 C 和 C++语言派生出来的，其语言体系都构建在 .NET 框架上，并且能够与 .NET 框架完美结合。C#语言具有以下突出的特点。

(1) 语法简洁

C#语言主要从 C 和 C++语言继承而来，摒弃了 C 和 C++语言中一些比较复杂而且不常用的语法元素，同时吸收了 Java 语言和 Delphi 语言的优点，它在设计时考虑了多数实际应用的需求。

C#语言取消了指针，不允许直接对内存进行操作，使代码运行在安全的环境中。

字符串定义和操作汲取了 Delphi 语言的 string 数据类型方便性的特点。“::”和“->”这两个域操作符和参照操作符也被“.”操作符所取代。

语法中的冗余是 C++语言中的常见的问题，如 const 和 #define、各种各样的字符类型等。

C#对此进行了简化，只保留了常见的形式，而其他冗余形式从它的语法结构中被清除了出去。

(2) 面向对象

C#语言是彻底的面向对象程序设计语言，与 C++语言相比，它把面向对象提高到了另一个层次。C#语言具有面向对象程序设计语言所应有的一切特性——封装、继承和多态。

C#语言只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。C#语言中没有了全局函数，没有了全局变量，也没有了全局常数，一切都必须封装在一个类之中。这样，代码将具有更好的可读性，并且减少了发生命名冲突的可能。

(3) 类型安全

在 C++语言中拥有一个指针，就能自由地把它强制转换为任何类型，只要内存支持的操作，它就可以做。这并不是企业级编程语言的类型安全。

C#语言实施最严格的类型安全，以保护自己及垃圾收集器 (Garbage Collector)。C#语言中变量的规则如下。

不能使用没有初始化的变量。对于对象的成员变量，编译器负责清零；而局部变量，则由用户程序负责清零。当用户程序使用一个没有初始化的变量时，编译器会提示怎么做。优点是能够避免由于使用不经初始化的变量计算结果而导致的错误，而用户还不知道这些奇怪的结果是如何产生的。

C#语言取消了不安全的类型转换，不能把一个整型强制转换成一个引用类型（如对象），而当向下转换时，C#验证这种转换是正确的（也就是说，派生类真的是从向下转换的那个类派生出来的）。

边界检查是 C#语言的一部分，再也不会出现这种情况：数组实际只定义了 $n-1$ 个元素，却超额使用了 n 个元素。

算术运算有可能溢出终值数据类型的范围。C#语言允许在语句级或应用程序级检测这些运算。在允许检测溢出的情况下，当溢出发生时将会抛出一个异常。

在 C#语言中，被传递的引用参数是类型安全的。

(4) 与 Web 的紧密结合

.NET 中新的应用程序开发模型意味着越来越多的解决方案需要与 Web 标准相统一，例如超文本置标语言 (Hypertext Markup Language, HTML) 和 XML。由于历史的原因，现存的一些开发工具不能与 Web 紧密地结合。SOAP 的使用使得 C#语言克服了这一缺陷，大规模深层次的分布式开发从此成为可能。

由于有了 Web 服务框架的帮助，对程序员来说，网络服务看起来就像是 C#的本地对象一样。程序员能够利用他们已有的面向对象的知识与技巧开发 Web 服务。仅需要使用简单的 C#语言结构，C#控件将能够方便地为 Web 服务，并允许它们通过 Internet 被运行在任何操作系统上的任何语言所调用。举个例子，XML 已经成为网络中数据结构传递的标准，为了提高效率，C#语言允许直接将 XML 数据映射成为结构。这样就可以有效地处理各种数据。

(5) 版本处理

在过去的几年中，几乎所有的程序员都至少有一次不得不涉及众所周知的“DLL 异版本调用异常”。该问题起因于多个应用程序都安装了相同 DLL 名字的不同版本。有时，老版本的应用程序可以很好地和新版本的 DLL 一起工作，但是更多的时候它们会中断运行。现在的版本问题真是令人头痛。

C#语言的 NGWS runtime 将对所写的应用程序提供版本支持。C#语言可以很好地支持版本

控制。尽管 C#语言不能确保正确的版本控制，但是它可以为程序员保证版本控制成为可能。有这种支持，一个开发人员就可以确保当他的类库升级时，仍保留着对已存在的客户应用程序的二进制兼容。

(6) 兼容性和灵活性

C#语言并没有存在于一个封闭的世界中。它允许使用最先进的 NGWS 的通用语言规定 (Common Language Specification, CLS) 访问不同的 API。CLS 规定了一个标准，用于符合这种标准的语言内部之间的操作。为了加强 CLS 的编译，C#编译器检测所有的公共出口编译，并在通不过时列出错误。

当然，C#语言还能够访问一些旧的 COM 对象。NGWS 运行时提供对 COM 透明的访问。OLE 自动化是一种特殊的机制，使用 C++语言创建 OLE 自动化项目的人已经喜欢上各种各样的自动化数据类型，C#语言支持它们，而没有烦琐的细节。最后，C#语言允许用 C 原型的 API 进行内部操作。可以从应用程序访问任何 DLL 中的入口点 (有 C 原型)。用于访问原始 API 的功能称为平台调用服务 (Platform Invocation Services, 缩写 PInvoke)。

尽管 C#代码的默认状态是类型安全的，但是可以声明一些类或者仅声明类的方法是非安全类型的。这样的声明允许使用指针、结构，静态地分配数组。安全码和非安全码都运行在同一个管理空间内，这意味着当从安全码调用非安全码时不会陷入列集 (Marshalling)。

1.1.2 C#语言与.NET 的关系

.NET Framework 是微软公司近年来主推的应用程序开发框架，该框架提供跨平台和跨语言的特性，C#是其主要的开发语言。使用.NET 框架，配合微软公司推出的 Visual Studio 集成开发环境，开发人员可以比以往更轻松地创建出功能强大的应用程序。

.NET Framework 的一个重要的控件就是公共语言运行时 (Common Language Runtime, CLR)，它是.NET Framework 的基础。可以将公共语言运行时看作一个在程序执行时管理代码的底层环境，它提供内存管理、线程管理和远程处理等核心服务，还强制实施严格的类型安全检查以及可以提高应用程序安全性和可靠性的其他代码正确性控制机制。事实上，代码管理的概念是公共语言运行时的基本原则。以公共语言运行时为目标的代码称为托管代码，而不以公共语言运行时为目标的代码称为非托管代码。

.NET Framework 的另一个主要控件就是框架类库，它支持多种应用程序的开发，这些应用程序包括控制台应用程序、Windows 应用程序、ASP.NET 应用程序、Windows 服务、XML Web 服务、Web 窗体。

C#语言是专门为与.NET Framework 一起使用而设计的。C#就其本身而言只是一种语言，尽管它用于生成面向.NET 环境的代码，但它本身不是.NET 的一部分。在安装 Microsoft Visual Studio 2010 的同时，.NET Framework 也被安装到本地计算机中。在.NET 中，C#语言的使用比例远远大于其他编程语言。

1.2 安装 Microsoft Visual Studio

下面将详细介绍如何安装 Microsoft Visual Studio 2010，其步骤如下。

(1) 将 Microsoft Visual Studio 2010 安装盘放到光驱中，光盘自动运行后会进入安装程序

界面；如果光盘不能自动运行，可以双击 setup.exe 可执行文件，应用程序会自动跳转到如图 1-1 所示的 Microsoft Visual Studio 2010 安装页面。该页面上有两个安装选项，即安装 Microsoft Visual Studio 2010 和检查 Service Release，一般使用第一个安装选项。



图 1-1 Microsoft Visual Studio 2010 安装页面

(2)单击第一个安装选项“安装 Microsoft Visual Studio 2010”，进入如图 1-2 所示的 Microsoft Visual Studio 2010 安装向导页面。



图 1-2 Microsoft Visual Studio 2010 安装向导页面

(3)单击“下一步”按钮，进入如图 1-3 所示的安装程序之起始页。在页面左侧显示的是关于 Microsoft Visual Studio 2010 安装程序所需组件的信息，右侧显示用户许可协议。

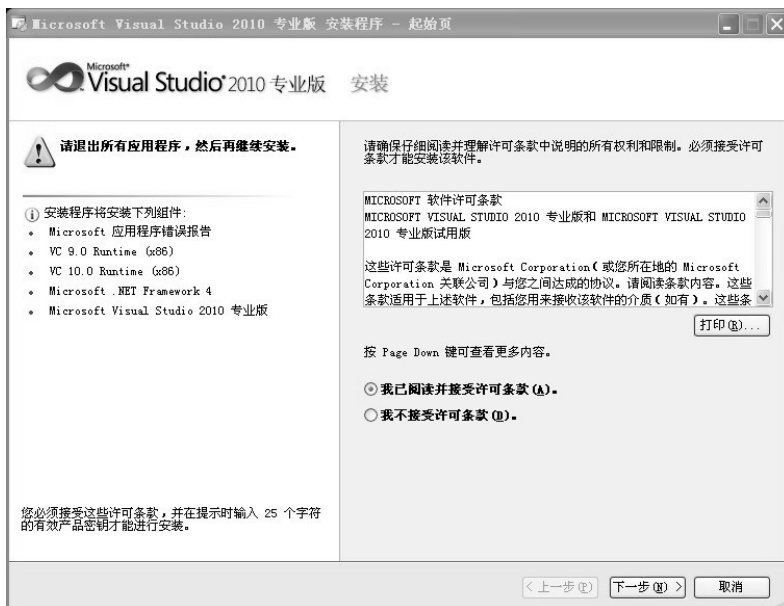


图 1-3 安装程序之起始页

(4) 选中“我已阅读并接受许可条款”单选按钮，单击“下一步”按钮，进入安装程序之选项页，用户可以从中选择要安装的功能和产品安装路径（产品安装默认路径为“C:\Program Files\Microsoft Visual Studio 10.0\”），也可以根据需要设置不同的安装路径，如图 1-4 所示。

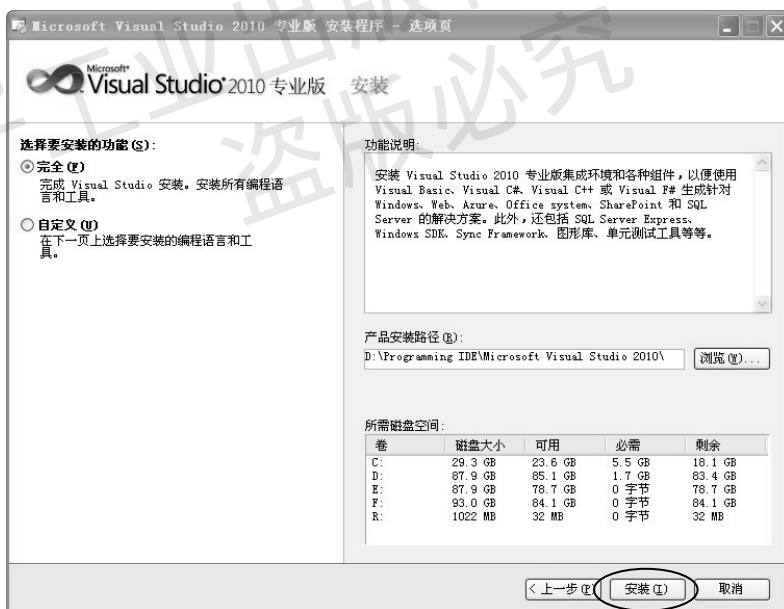


图 1-4 安装程序之选项页

说明：在“选择要安装的功能”栏中，用户可以选择“完全”和“自定义”两种安装方式。如果选择“完全”，安装程序将安装系统的所有功能；如果选择“自定义”，用户可以选择希望安装的项目，增加了安装程序的灵活性。通常，推荐用户选择“完全”。

(5) 单击“安装”按钮，进入如图 1-5 所示的安装程序之安装页，显示正在安装控件。



图 1-5 安装程序之安装页

(6) 安装完毕后，单击“下一步”按钮，进入安装程序之完成页，单击“完成”按钮，至此完成 Microsoft Visual Studio 2010 开发环境的安装。

1.3 Microsoft Visual Studio 集成开发环境

Microsoft Visual Studio 是创建和测试项目的平台，Microsoft Visual Studio 集成开发环境 (IDE) 由各种工具组成，包括允许直观创建窗体的窗体设计器、用于输入和修改代码的智能编辑器、把 C#语句翻译成中间机器代码的编译器、帮助查找和修正程序错误的调试器、用来查看对象的属性和方法的对象浏览器及帮助工具等。

1.3.1 Microsoft Visual Studio 主窗口

如图 1-6 所示为 Microsoft Visual Studio 的主窗口，其中的所有窗口都可以变为可停靠或浮动、隐藏或可见，也可以移动到新位置。若要更改窗口的行为，可以单击该窗口标题栏中的向下箭头或图钉图标，然后从中进行选择。若要将停靠窗口移动到新位置，可以拖动标题栏，直至出现一个滴管图标，按住鼠标左键的同时，将指针移至新位置该图标的上方。将指针放在左侧、右侧、顶端或底端图标的上方，可使该窗口停靠在指定一侧。将指针放在中间图标的上方，可使该窗口成为选项卡式窗口。在放置指针时，将出现一个蓝色半透明的矩形框，它指示该窗口将停靠在新位置的什么地方。

Microsoft Visual Studio 主窗口主要通过菜单栏和工具栏进行操作。如果所使用的版本是中文版的，则可以直接通过中文菜单项进行操作；对于工具栏，只要将指针移至工具栏中的某选项上，就会出现该工具栏选项的中文提示（也称为 Hint），所以这里不再对菜单栏和工具栏功能一一做介绍。

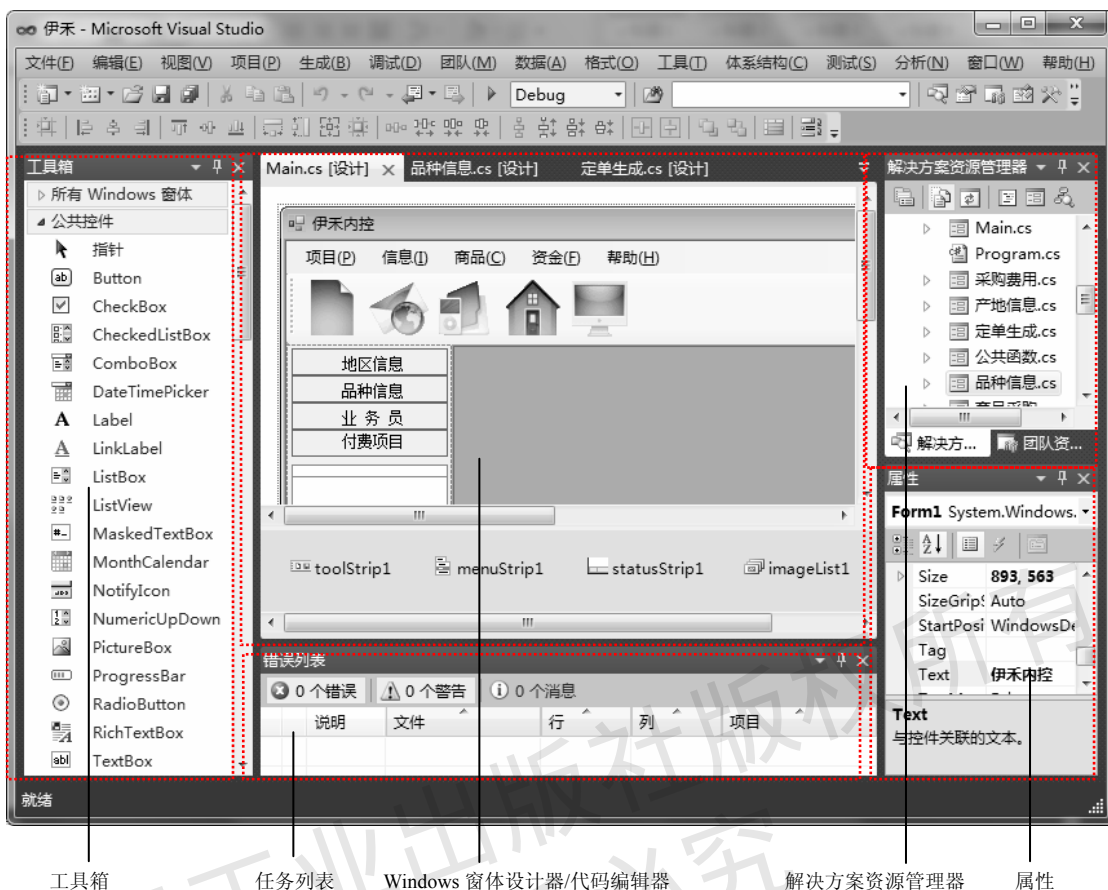


图 1-6 Microsoft Visual Studio 主窗口

1.3.2 代码编辑器与 Windows 窗体设计器

在 Windows 窗体程序设计过程中，代码编辑器用来编写程序代码，Windows 窗体设计器用于控件布局及外观的设计。由于代码编辑器和 Windows 窗体设计器合用一个窗口，因此不能同时看得到，需要通过按 F7 键（或 Shift+F7 组合键），或者选择菜单项“视图”→“代码”（或“设计器”），在代码编辑器和 Windows 窗体设计器之间进行切换。为简便起见，本书中将这两个窗口统称为“编辑器/设计器窗口”。

1. 代码编辑器

代码编辑器是用于编写源代码的字处理程序。就像 Word 对句子、段落和语法提供广泛支持一样，C#代码编辑器也为 C#语法和 .NET Framework 提供广泛支持。这些支持可以分为 5 个主要的类别。

- ① IntelliSense: 在将 .NET Framework 类和方法输入编辑器中时，不断对其基本文档进行更新，同时还具有自动代码生成功能。
- ② 重构: 随着基本代码在开发项目过程中的演变，智能重构基本代码。
- ③ 代码段: 可以浏览的库，其中包含了频繁重复的代码模式。
- ④ 波浪下画线: 当输入内容时，对拼写错误的单词、错误的语法以及警告情况的可见通知。
- ⑤ 可读性帮助: 大纲显示和着色。

2. Windows 窗体设计器

Windows 窗体设计器用于为窗体添加控件，对它们进行排列，并为其事件编写代码。可以使用 Windows 窗体设计器完成以下操作。

① 向某个窗体添加控件、数据控件或基于 Windows 的控件。

② 双击设计器中的窗体并为该窗体编写 Load 事件中的代码，或双击窗体上的控件并为该控件的默认事件编写代码。

③ 通过选择某个控件并输入一个名称来编辑该控件的 Text 属性。

④ 通过使用鼠标或方向键移动选定的控件来调整该控件的位置。同样，使用 Ctrl 键和方向键可更精确地调整控件的位置。最后，使用 Shift 键和方向键调整控件的大小。

⑤ 按住 Shift 键并单击或按住 Ctrl 键并单击可选择多个控件。当对齐或操控控件大小时，如果按住 Shift 键并单击，则第一个选定的控件是主导控件。如果按住 Ctrl 键并单击，则最后选定的控件是主导控件，因此每次添加新控件时，主导控件会随之更改。或者，也可以单击窗体，拖动鼠标，在想要选择的控件周围画一个矩形选择框，以选择多个控件。

Windows 窗体设计器为创建基于 Windows 的应用程序提供了一种快速的开发解决方案。在该设计器中可以进行可视化的、基于客户端的窗体设计。可从“工具箱”中将控件拖动或绘制到其图面上。

添加到 Windows 窗体设计器中的非可视化控件将放置在位于设计图面下方的“控件栏”中，以便于访问而不干扰可视化设计空间。

此外，在 Windows 窗体设计器中右击会显示一个快捷菜单，可以使用其中的命令切换到代码编辑器，将控件锁定到窗体上，或查看窗体的属性。

1.3.3 解决方案资源管理器和项目设计器

1. 解决方案资源管理器

解决方案资源管理器以分层树视图的方式显示项目中的所有文件。如果使用“项目”菜单项将新文件添加到项目中，将看到这些文件显示在解决方案资源管理器中。除文件外，解决方案资源管理器还显示项目设置，以及对应用程序所需的外部库的引用。

2. 项目设计器

右击解决方案资源管理器中的“Properties”节点，然后在快捷菜单中选择“打开”命令，将访问“项目设计器”，如图 1-7 所示。使用其中的属性页可以修改生成选项、安全要求、部署详细信息以及许多其他项目属性。

若要访问单个文件的属性，应使用属性窗口。

在项目设计器中可以集中管理项目的属性、设置和资源。与其他设计器（如窗体设计器或类设计器）一样，项目设计器在 Microsoft Visual Studio IDE 中作为单一窗口出现。

1.3.4 编译器、调试器和错误列表窗口

1. 编译器

C# 编译器没有窗口，因为它不是交互式工具，但可以在项目设计器中设置编译器选项。

选择菜单项“生成”→“生成”，IDE 将调用 C#编译器。如果生成成功，则在状态窗格中将显示“生成成功”消息。如果存在生成错误，将在编辑器/设计器窗口的下方出现“错误列表”窗口。双击其中某个错误可以转到源代码中相应的问题行。按 F1 键可以查看针对突出显示的错误的帮助文档。



图 1-7 项目设计器

2. 调试器

调试器具有多个不同的窗口，这些窗口随着应用程序的运行而显示不同变量的值和类型信息。在调试器中调试时，可以使用代码编辑器指定在某一行暂停执行，以及每次一行单步执行代码。

3. 错误列表窗口

“错误列表”有助于用户加快应用程序开发的过程。在“错误列表”窗口中，可以进行以下操作。

- ① 显示在编辑和编译代码时产生的错误、警告和消息。
- ② 查找 IntelliSense 所标出的语法错误。
- ③ 查找部署错误、某些静态分析错误以及在应用“企业级模板”策略时检测到的错误。
- ④ 双击任意错误信息项打开出现问题的文件，然后转移到错误位置。
- ⑤ 筛选显示哪些项，以及为每项显示哪些列的信息。

要显示“错误列表”窗口，选择菜单项“视图”→“错误列表”。

1.3.5 工具箱

“工具箱”是一个浮动的树控件，它与 Windows 资源管理器的工作方式非常类似，但没有网格或连接线。可以同时展开工具箱的多个页（称为“选项卡”），整个目录树在工具箱内部滚动。要展开工具箱的任何选项卡，单击它名称旁边的加号(+); 要折叠一个已展开的选项卡，单击它名称旁边的减号(-)。

工具箱中显示可以添加到项目中的项的图标。每次返回编辑器或设计器时，工具箱都会自

动滚动到最近选择过的选项卡和项。当把焦点转移到其他编辑器、设计器或另一个项目时，工具箱中当前选定内容也相应转移。

注意：当编辑控制台应用程序时，不会显示工具箱中的项，因为通常它们的设计不具有图形用户界面。

可以通过在选项卡中重排项或添加自定义的选项卡和项来自定义“工具箱”。

当选择其他设置组合时，将会清除当前的工具箱状态，即，选择其他设置组合会更改现在可用的工具箱选项卡，以及在选项卡中显示的项。

1.3.6 属性窗口

使用属性窗口可查看和更改位于编辑器和设计器中的选定对象的设计时的属性及事件。也可以使用属性窗口编辑和查看文件、项目和解决方案的属性。属性窗口可通过“视图”菜单项打开。

属性窗口中显示编辑字段的不同类型，具体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。属性以灰色显示并且是只读的。属性窗口的选项说明见表 1-1。

表 1-1 属性窗口的选项

名称	声明
对象名	列出当前选定的一个或多个对象（只有活动编辑器/设计器中的对象可见）。当选择多个对象时，只出现所有选定对象的通用属性
按分类顺序	按类别列出选定对象的所有属性及属性值。可以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	按字母顺序对选定对象的所有设计时的属性和事件排序。要编辑可用的属性，在它右边的单元格中单击并输入更改内容
属性	显示对象的属性，如图 1-8 (a) 所示
事件	显示对象的事件，如图 1-8 (b) 所示
属性页	显示选定对象的“属性页”对话框或项目设计器。“属性页”对话框显示属性窗口中可用属性的子集、同一集合或超集。使用该按钮可查看和编辑与项目的活动配置相关的属性

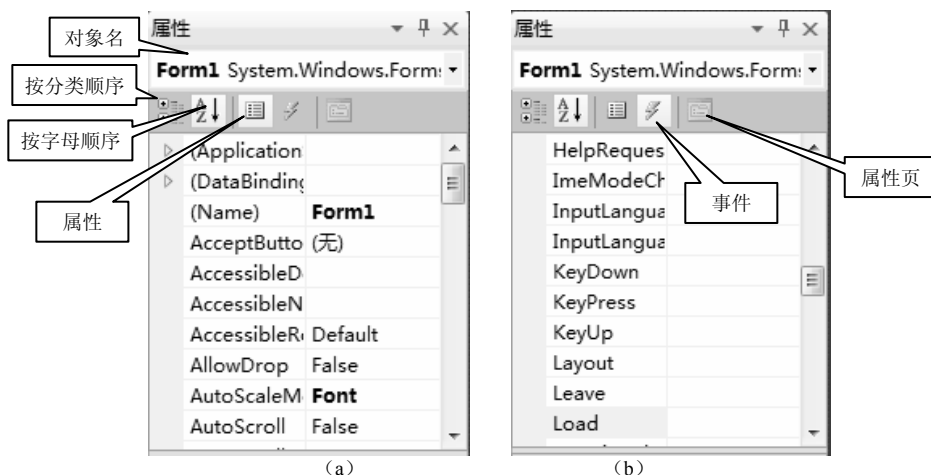


图 1-8 属性窗口

1.4 Visual C#项目

准备好开始编写代码后，第一步是设置项目。项目包含应用程序的所有原始资料，不仅包括源代码文件，还包括资源文件，如图标、对程序依赖的外部文件的引用，以及配置数据（如编译器设置）。生成项目时，Visual C#调用 C#编译器和其他内部工具，以使用项目中的文件创建可执行程序集。

1.4.1 创建新项目

选择菜单项“文件”→“新建”→“项目”，打开“新建项目”对话框，如图 1-9 所示。可以看到在左侧项目类型列表中默认选中“Visual C#”下的“Windows”项，在右侧模板列表中，可以选择需要的项目模板。如果展开项目类型列表中的“其他语言”或“其他项目类型”项，可以看到在右侧模板列表中显示不同的项目模板。

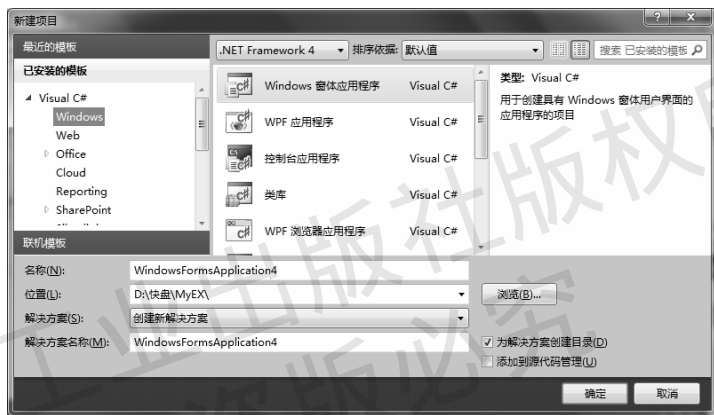


图 1-9 “新建项目”对话框

初学者工具包是另一种类型的项目模板。如果用户安装了初学者工具包，将会看到该工具包在“新建项目”对话框中列出。

选择项目模板并单击“确定”按钮后，Visual Studio 将创建项目，就可以开始编写代码了。项目文件、引用、设置以及资源均显示在主窗口右边的解决方案资源管理器中，如图 1-10 所示。



图 1-10 解决方案资源管理器

1.4.2 项目中的内容

1. 属性

“属性”节点表示应用于整个项目的配置设置，这些设置存储在解决方案文件夹下的.csproj文件中。这些设置包括编译选项、安全性和部署设置，以及其他更多设置。可以使用项目设计器对项目进行修改。

2. 引用

在项目的上下文中，引用仅标识应用程序运行所需的二进制文件。通常，项目引用标识 DLL 文件，如 .NET Framework 类库文件之一。它也可以引用 .NET 程序集（称为 shim），使用户的应用程序可以调用 COM 对象或本机 Win32 DLL 上的方法。如果程序创建其他程序集中定义的类的实例，则必须在项目中添加对该文件的引用，才能编译项目。在默认情况下，所有 C# 项目都包含对 mscorlib.dll 的引用，mscorlib.dll 包含核心 .NET Framework 类。选择菜单项“项目”→“添加引用”，可以添加对其他 .NET Framework DLL 或文件的引用。

注意：不要将项目引用的概念与 C# 或其他编程语言中的引用类型的概念相混淆。前者指文件及其在磁盘上的预期位置，后者指使用 class 保留字声明的 C# 类型。

3. 资源

资源是应用程序中包含的数据，但是可以独立于其他源代码进行修改的方式存储。例如，可以将所有字符串作为资源存储，而不是将它们硬编码到源代码中。可以在日后将这些字符串翻译成不同语言，然后将它们添加到交付给客户的应用程序文件夹中，而不必重新编译程序集。Visual C# 定义了 5 种类型的资源：字符串、图像、图标、音频和文件。可以使用“资源设计器”（项目设计器中的“资源”选项卡）添加、移除或编辑资源。

4. 窗体

创建 Windows 窗体项目时，在默认情况下，Visual C# 会将一个窗体添加到项目中。

也可以选择菜单项“项目”→“添加 Windows 窗体”，添加新窗体。每个窗体都有两个文件与其相关联，分别为 Form1.cs 和 Form1.Designer.cs。

Form1.cs（可以任意命名该文件）包含写入的源代码，这些源代码用于配置窗体及其控件（如列表框和文本框），并对事件（如单击按钮和按键）做出响应。在简单的 Windows 窗体项目中，需要在此文件中进行大部分或全部代码编写工作。

Form1.Designer.cs 文件包含从工具箱中将控件拖动到窗体上、在属性窗口中设置属性等操作时，在窗体设计器中写入的源代码。通常，不要手动编辑此文件。

注意：如果创建控制台应用程序项目，它将不包含 Windows 窗体的源代码文件。

5. 其他源代码文件

项目可以包含任意数量的其他 .cs 文件，这些文件可能与特定的 Windows 窗体相关联。在图 1-10 中，Program.cs 包含应用程序的入口点。单个 .cs 文件可以包含任意数量的类和结构定义。选择菜单项“项目”→“添加新项”或“添加现有项”，将新的或现有的文件或类添加到项目中。

1.4.3 修改项目属性

创建项目之后，可以使用项目设计器（见图 1-7）执行各种任务，例如，更改可执行文件的名称、自定义生成过程、添加对 DLL 的引用或加强安全设置。选择菜单项“项目”→“属性”，或在解决方案资源管理器中右击“属性”项，从快捷菜单中选择“打开”命令，可以访问项目设计器。项目设计器将显示在编辑器/设计器窗口中。

在项目设计器中，对各种项目属性进行分组，包含 10 个属性页。项目设计器与 Windows 窗体设计器和代码编辑器位于同一个窗口中。

注意： Visual Studio Team System 中包含用于代码分析的附加属性页。

在图 1-7 中，显示的是“应用程序”属性页。通过单击左侧标签（“生成”、“生成事件”、“调试”等），可以访问相应的属性页。此处输入的项目特定信息存储在.csproj 文件中，该文件在解决方案资源管理器中不可见，但它位于项目文件夹中。使用 Visual C#时，将鼠标指针定位于任意属性页上，然后按 F1 键，即可访问该属性页的帮助。表 1-2 给出对项目设计器中主要属性页的简短说明。

表 1-2 项目设计器中主要属性页说明

属性页	说明
应用程序	更改程序集的名称、输出类型及程序集信息（其中包括版本号和其他资源选项）
生成	更改已编译的程序集存储的位置、条件编译选项、错误和警告的处理方式以及其他设置
生成事件	创建和修改自定义生成步骤
调试	指定在调试器下运行时的命令行参数，以及其他设置
资源	将字符串、图标、图像或其他类型的文件作为资源添加到项目中
设置	存储设置，如数据库的连接字符串，或特定用户要使用的配色方案。在运行时可以动态检索这些设置
引用路径	指定项目中引用的程序集所在位置的路径
签名	指定 ClickOnce 证书选项，并提供程序集的强名称
安全性	指定运行应用程序所需的安全设置
发布	指定将应用程序分发到网站、FTP 服务器或文件位置所使用的选项
代码分析	（仅限 Visual Studio Team System） 用于分析源代码是否存在潜在安全问题、是否遵从 .NET Framework 设计指南，以及更多内容的工具选项

1.4.4 生成和调试

在 Visual C#中，选择菜单项“生成”→“生成”，或按 Ctrl+Shift+B 组合键，将生成可执行应用程序。选择菜单项“调试”→“开始执行”或按 F5 键，可以通过一步操作生成并启动应用程序。

生成过程包括将项目文件输入 C#编译器中，将源代码转化为 Microsoft 中间语言（MSIL），然后将 MSIL 和元数据、资源、清单以及其他模块（如果有的话）进行连接，从而创建一个程序集。程序集是一个可执行文件，扩展名通常为.exe 或.dll。开发应用程序时，有时希望生成调试版本，以进行测试并查看程序的运行状况。最后，在全部无误的情况下，将创建一个发布版本部署给客户。

1.5 C#程序设计过程

C#程序设计是可视化的,大多数开发的程序界面风格类似于其他 Windows 应用程序或 Web 应用程序。在开发过程中,基本上以一个项目文件管理一组窗体单元文件,而各个窗体根据功能需要,安排了不同的控件,运行时通过这些控件与用户交互,这些控件所要实现的功能由其背后的事件(程序代码)实现。

在 C#中创建应用程序的过程一般如下。

1.5.1 新建项目

1. 运行 Microsoft Visual Studio

选择菜单项“开始”→“所有程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”,将启动 Microsoft Visual Studio 并显示“起始页”。

2. 新建项目

选择菜单项“文件”→“新建”→“项目”,打开“新建项目”对话框,如图 1-9 所示。在项目类型列表中选择“Visual C#”下的“Windows”,在模板列表中选择“Windows 窗体应用程序”,如图 1-11 所示。

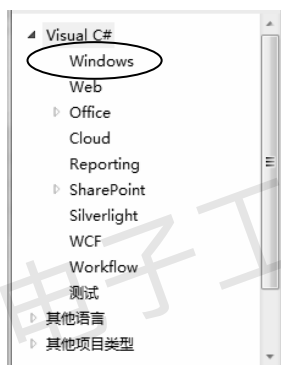


图 1-11 项目类型列表

在“名称”框中输入项目名,在“位置”框中输入项目及程序文件所存放的路径(也可以单击“浏览”按钮指定一个已有的文件夹),在“解决方案名称”框中输入解决方案名。

单击“确定”按钮,进入 Visual C#项目开发环境。

3. 设置环境

环境设置根据每个人的习惯而定,不是必须的,这里简单描述环境的定制,具体使用时根据可需要进行选择。

选择菜单项“窗口”→“重置窗口布局”,在提示框中选择“是”,重置为 IDE 的默认布局。

工具箱默认为自动隐藏,移动鼠标指针指向 IDE 窗口左边的工具箱图标,会显示工具箱窗口,可单击工具箱顶部的图钉图标,将使工具箱窗口固定在屏幕上,而不再自动隐藏。

选择菜单项“工具”→“选项”,打开“选项”对话框,如图 1-12 所示,在对话框左侧列表中,选择要设置的项,在右侧页面中进行具体设置。

1.5.2 定义用户界面

1. 设置窗体

窗体设计器中的新窗体具有所有标准的 Windows 特征,如包括标题栏、“最大化”按钮和“最小化”按钮以及“关闭”按钮。

可以直接在窗体设计器中调整窗体的大小,并在属性窗口中设置窗体的标题栏、颜色等窗体属性。

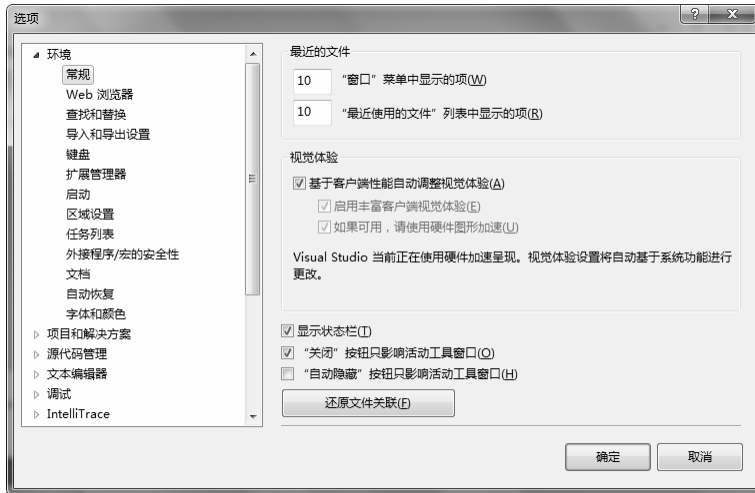


图 1-12 “选项”对话框

2. 在窗体上放置控件

从工具栏箱中单击选择控件（如按钮或文本框等），拖动至窗体上。如果是可视控件，则可以在窗体上调整控件的位置（直接拖放控件）、大小（拖动控件四周的点）等；如果是非可视组件，则系统会把该组件放入窗体下面的列表中。

选定窗体上的控件后，可以按 **Delete** 键删除。

根据需要，可使用参照线来帮助调整控件的大小和对齐，使控件排放整齐美观。

1.5.3 设置属性

属性（Property）是某些描述性的数值，如控件的名称、大小、位置、字体、颜色等，这些数值的设定决定了窗体或控件显示的外观。

根据程序需求可在属性窗口中设置控件的属性，也可以通过程序动态地设置和修改属性。

如果属性窗口不可见，则选择菜单项“视图”→“属性窗口”或按 **F4** 键，使其显示出来。

从窗体上选择控件，或从属性窗口的下拉列表中选择窗体或控件，使之成为当前控件，就可以用属性窗口来设置属性了。根据属性标题的情况，设置属性可分为 4 种情况。

第一种情况是单击属性标题，直接在属性值编辑框中输入属性的数值或字符串，如 **Name**、**Text** 等属性。

注意：如果输入的属性值不符合要求，例如，在应输入数值的属性值中错误输入字符串，**Visual C#**将给出错误信息。

第二种情况是属性值由 **Visual C#**提供可选项。这时，单击属性标题后，显示一个下拉列表框。用户可单击下拉按钮打开下拉列表，从中选择所需的属性值。可选项分为两种：一种是文字可选项，如 **BackgroundImageLayout**、**Enabled** 等属性；另一种是图形加文字形式的可选项，如 **ForeColor**、**Cursor** 等属性。

第三种情况是通过对话框来设置属性值。对于 **Image**、**BackgroundImage**、**Font** 等属性，单击属性标题后，属性值右端会出现“...”按钮，即浏览按钮，单击该按钮，将打开一个相应的对话框，在对话框中选择所需值，即可设置属性值。

第四种情况是属性带有子属性。在这种属性的属性标题前，一般会有一个“+”号，单击该“+”号，可以展开并显示子属性，同时“+”号变成“-”号。可依次对各子属性进行设置。

1.5.4 添加事件

事件（Event）是一种消息处理机制，利用它可以捕捉某种情况的发生，并对其做出相应的反应。例如，单击鼠标、拖动鼠标、双击鼠标或按下键盘等都是典型的事件。

从窗体上选择一个控件，或从属性窗口的对象名下拉列表选择一个窗体或控件，使之成为当前控件，就可以用属性窗口的“事件”选项卡来添加事件了。

添加事件的过程一般是选定控件后，单击属性窗口中的事件选项卡，选择要添加的事件名称，在其右侧编辑框中输入事件的处理（响应）函数名，然后回车即可。

提示：如果单击要添加的事件后，在其右侧编辑框中双击鼠标，将为事件采用默认事件处理函数名，同时进入代码编辑窗口。

默认事件处理函数名是控件名和事件名合并而成的字符串，如 button1 控件的 Click 事件的默认事件处理函数名是 button1_Click。

每个事件都将以 private void 开始，代码编写在 { } 之内，以实现需要的功能。

1.5.5 运行项目

在完成代码的编写之后，即可运行该项目。选择下列三种方法之一，开始项目运行。

- ① 选择菜单项“调试”→“启动调试”。
- ② 单击工具栏中的“启动调试”按钮。
- ③ 按 F5 键，即“启动调试”命令对应的快捷键。

如果程序中存在 C#语法错误，或存在无法解析为已知类型或成员的标识符，生成将失败，并在“错误列表”窗口中出现一个错误列表。可以双击错误信息，跳转到发生错误的代码行，调试修改后重新运行。

通常，C#编译器的错误提示信息会非常明确，并且具有较强的描述性，但如果无法断定问题所在，可以在错误列表中选中该错误信息，并按 F1 键，转到该信息对应的帮助页面。“帮助”页面中还包含其他有用信息。如果问题仍然无法解决，则可以考虑在 C#论坛或新闻组中提出该问题。要访问论坛，可选择菜单项“社区”→“提出问题”。

在运行界面进行操作时，若出现问题或程序的运行结果达不到预期的结果，退出运行界面后，可以使用 Visual C#提供的调试器来进行调试和修改，再重新运行，直到满意为止。

1.5.6 保存文件和关闭项目

应当做好文件保存工作。

1. 保存文件

选择菜单项“文件”→“全部保存”，系统会保存当前的项目、窗体和解决方案文件。

2. 关闭项目

选择菜单项“文件”→“关闭解决方案”，如果自上次修改以来尚未保存，则系统将提示是否保存，根据需要选择保存或不保存。

1.5.7 打开项目

要对一个已有的项目进行修改，需要重新打开项目，可以通过三种方法来打开已保存的项目。

① 选择菜单项“文件”→“打开”→“项目/解决方案”，打开“打开项目”对话框，从中找到项目主文件.sln文件。

② 选择菜单项“文件”→“最近的项目”下的项目文件。

③ 从“起始页”的“最近的项目”中选择项目文件。

打开项目后，如果屏幕上看不到窗体，检查解决方案资源管理器，双击表示窗体的图标，然后单击解决方案资源管理器顶部的“查看设计器”按钮，窗体将出现在 Windows 窗体设计器中。还可以单击“窗口代码”按钮，从而在代码编辑器中显示窗体的代码。

有时需要在文件中查找指定指令，或替换某个标识符，可以按 Ctrl+Shift+F 组合键打开“在文件中查找”对话框，从而在整个项目中执行查找和替换操作。

【例 1-1】 本例介绍如何在 Visual C#集成开发环境中编写 Windows 应用程序。程序实现的功能是：在应用程序的窗体中显示时钟。

具体实现过程如下。

(1) 进入 Visual C#开发环境

选择菜单项“开始”→“所有程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”，启动并显示“起始页”。

(2) 创建项目

为了方便管理文件，可以创建一个“项目”进行统一管理。

① 创建项目文件夹

为了单独存放与项目有关的文件，先创建一个文件夹，如 C:\Mypro，可以通过 Windows 环境中的资源管理器来实现。

② 创建一个新的项目

选择菜单项“文件”→“新建”→“项目”，打开“新建项目”对话框，在项目类型列表中选择“Visual C#”，在模板列表中选择“Windows 窗体应用程序”。

在“名称”框中输入项目名 ex01_01，在“位置”框中设置项目及程序文件所存放的路径 C:\Mypro，在“解决方案名称”框中输入解决方案名。

(3) 向窗体中添加控件

要实现在 Form1 窗体中显示文字，可以添加一个 Label（标签）控件。标签控件在工具箱的“公共控件”选项卡中。可以将鼠标指针移动到“公共控件”选项卡中标签控件上，单击选中该控件，然后移动鼠标指针到 Form1 窗体中再次单击，标签控件就被添加到窗体中。

要实现每隔 1 秒钟将显示的内容改变一次，采用同样的方法，向 Form1 窗体中添加一个 Timer（定时器）控件。

定时器控件是非可视组件，所以组件图标将显示在设计器的下方。

要实现关闭显示，添加一个 Button（按钮）控件，采用同样方法将其添加到窗体中。

(4) 设置控件属性

在 Form1 窗体中选中标签控件，然后在属性窗口中对标签控件的主要属性进行设置，具体如表 1-3 所示。

表 1-3 例 1-1 中标签控件的属性设置

属 性	属 性 值
Text	00:00:00
Font	黑体, 粗体, 初号
ForeColor	0, 0, 192

注意：改变字体属性时，可以单击编辑框右边的浏览按钮，在打开的“字体”对话框中设置字体、大小等，也可以展开 Font 属性，对子属性进行设置。

设置好控件的属性后，可以改变窗体的大小：将鼠标指针移到窗体的边缘，指针的形状会变成双箭头，按下鼠标左键拖动就可以改变窗体的大小。

然后，在 Form1 窗体中选中按钮控件，在属性窗口中将按钮的 Text 属性设置为“停/启”。设置好的窗体如图 1-13 (a) 所示。

(5) 添加代码

① 在 Form1 窗体中选中定时器控件，然后选中属性窗口中的事件 (Events) 选项卡，为定时器控件添加 Tick 事件。双击 Tick 栏右边的编辑框，系统会自动添加该事件，并自动定位到相应的代码处。

在 timer1_Tick 中添加以下代码：

```
private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToLongTimeString(); //显示时间字符
}
```

② 同样，在 Form1 窗体中选中按钮控件，然后选中属性窗口中的事件选项卡，为按钮控件添加单击 (Click) 事件 (即运行时，单击按钮所要运行的程序)。双击 Click 栏右边的编辑框，进入代码编辑器，给按钮控件添加单击事件过程如下：

```
private void button1_Click(object sender, EventArgs e)
{
    timer1.Enabled = !timer1.Enabled;
}
```

(6) 运行程序

选择菜单项“调试”→“开始执行”运行程序。如果程序没有经过编译和链接，则系统会自动进行编译和链接。运行结果如图 1-13 (b) 所示。

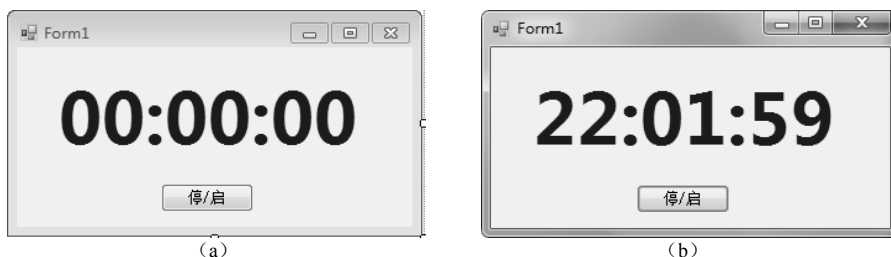


图 1-13 显示时钟例程

说明：例 1-1 通过定时器每秒执行一次 void timer1_Tick 事件，在事件中，通过“label1.Text=DateTime.Now.ToLongTimeString();”语句改变控件 label1 的 Text 属性，也就是将窗体中显示的内容每秒改变一次。而按钮的单击事件通过“timer1.Enabled=!timer1.Enabled;”语句将 timer1 控件的 Enabled 属性在 true 和 false 之间进行切换，以达到停、启的目的。程序代码中“//”符号代表其后为注释文本。

读者可以自己尝试进行以下工作。

- ① 改变窗体中显示内容交替的频率。可以通过设置定时器的 Interval 属性来实现。
- ② 改变窗体标题栏的内容。例如，可以将 Form1 窗体的 Text 属性由“Form1”改为“小闹铃”。
- ③ 改变窗体的大小，显示内容、字体和颜色等。
- ④ 有兴趣的读者还可以让程序在运行过程中发出声音，这样提示的效果会更明显。可以在 void timer1_Tick 事件的末尾添加一条语句实现：

```
private void timer1_Tick(object sender, EventArgs e)
{
    ...
    Console.Beep();    //发声
}
```

- ⑤ 如果想改成每分钟报时发声一次，可以在上述语句中加一条判断语句：
if (DateTime.Now.Second == 0) Console.Beep(); //发声

1.6 Microsoft Visual Studio 的帮助

Microsoft Visual Studio 的“帮助”文档包含在 MSDN library 中，用户可以将 MSDN library 安装在本地计算机中，也可以从 Internet 上的 <http://www.microsoft.com/china/msdn/library> 位置获得。该库的本地版本是格式为 .hxs 的压缩 HTML 文件的集合。可以选择在计算机中安装该库的全部或部分内容，MSDN 完全安装的大小接近 2GB，并且其中包括很多 Microsoft 技术文档。使用称为 Microsoft 文档资源管理器的 Microsoft Visual Studio 帮助浏览器可以查看本地和联机 MSDN 文档。

1. 联机与本地帮助

在“帮助选项”属性页中，可以指定搜索行为（包括 F1 键搜索）的下列选项：

- (1) 首先尝试搜索联机 MSDN library，如果未找到匹配项，再尝试搜索本地文档。
- (2) 首先尝试搜索本地 MSDN library，如果未找到匹配项，再尝试搜索联机文档。
- (3) 只尝试搜索本地 MSDN library。

第一次调用任何搜索时，都会出现这些选项。联机 MSDN 文档可能比本地文档包含更多更新内容。因此，如果在使用 Visual C# 的过程中可以连接 Internet，建议将搜索选项设置为首先尝试搜索联机 MSDN library。本地文档一旦更新即可下载。

可以通过以下几个地址访问联机（基于 Internet）帮助：MSDN Online、MSDN 新闻组及由 Microsoft 赞助的名为 Codezone 社区的开发人员网站。Codezone 社区非常有价值，因为这个团队中包括使用 Visual Studio 和 Visual C# 编写实际应用程序的专业开发人员，其中的内容和建

议是持续更新的，因此能反映 Visual C#编程领域的当前发展趋势、热点及成果。

2. F1 键搜索

F1 键提供区分上下文的搜索功能。在代码编辑器中，将插入点光标定位于保留字或类成员上或紧随其后，并按 F1 键，可以访问 C#保留字和 .NET Framework 类的“帮助”文档。当对话框或任何其他窗口具有焦点时，可以按 F1 键获取该窗口的“帮助”。

F1 键搜索仅返回一页。如果没有找到匹配项，将显示信息性页面，提供一些故障排除提示。

3. 搜索

使用搜索界面，将返回与任何指定的术语或术语集相匹配的所有文档。

在“帮助选项”属性页中，指定除了搜索 MSDN library 以外，是否还要搜索 Codezone 网站。Codezone 内容只可以联机使用。

相同的联机与本地搜索选项可应用于搜索和 F1 键搜索。

在搜索界面中，指定要包括的文档类型，可以缩小或扩大搜索范围。此处有三个选项：语言、技术和主题类型。通常，只选中适用于当前开发方案的那些选项，将获得最佳效果。

4. 索引

索引可以快速找到本地 MSDN library 中的文档。它不是全文搜索，而是只搜索分配给每个文档的索引保留字。索引查找通常比全文搜索更快而且相关性更强。如果不止一个文档中包含了索引搜索框中指定的索引保留字，将打开歧义消除窗口，可以从可能的选项中进行选择。

在默认情况下，索引窗口位于文档资源管理器的左侧。可以从“帮助”菜单访问该窗口。

5. 目录

MSDN library 目录以分层树视图结构显示库中的所有主题。它非常有用，可以浏览文档，了解库中所包含的内容，还可以浏览无法通过索引或搜索找到的文档。通常，通过 F1 键、索引或搜索查找文档时，知道文档在目录中的位置十分有用，这样就可以查看存在哪些与给定主题相关联的其他文档。单击文档资源管理器工具栏中的“与目录同步”按钮，可查看当前显示的页面在 MSDN library 中的位置。

6. 如何实现

“如何实现”是 MSDN library 的筛选视图，其中主要包括称为“如何”或“演练”的文档，这些文档说明如何执行特定任务。可以从文档资源管理器工具栏、“帮助”菜单或“开始”页访问“如何实现”帮助。Microsoft Visual Studio 的每种语言都有其各自的“如何实现”页面，所显示的页面取决于当前的活动项目类型。

7. 动态帮助

“动态帮助”窗口根据代码编辑器中插入点的当前位置，显示到 .NET Framework 和 C#语言的参考文档的链接。

1.7 实验：在 Microsoft Visual Studio 环境中编写 C#程序

【实验目的】

熟悉 Microsoft Visual Studio 集成开发环境，掌握编辑、编译、运行 C#程序的操作。

【实验要求】

创建控制台应用程序，输入程序代码，编译、运行程序，发现并改正错误。

【示范模仿】

实验 1.1 已给出源程序，对其进行编辑、保存、编译和运行等操作。

编写程序 tr0101，程序功能是读入一个整数 n ，求 $0\sim n$ 间的所有偶数。源程序如下：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace tr0101
{
    class Program
    {
        static void Main(string[] args)
        {
            double n,s; //声明变量
            Console.Write("请输入一整数: "); //提示输入整数
            n = Convert.ToDouble(Console.ReadLine()); //输入 n
            s = 1;
            while (s < n)
            {
                if (s % 2 == 0)
                    Console.WriteLine(s); //输出 s
                s++;
            }
            Console.ReadKey(); //按任意键退出
        }
    }
}
```

(1) 进入 Visual Studio 开发环境

选择菜单项“开始”→“所有程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”，启动 Visual Studio。

(2) 创建项目

打开“新建项目”对话框，在项目类型列表中选择“Visual C#”，在模板列表中选择“控制台应用程序”。

在“名称”框中输入项目名 tr0101，在“位置”框中设置项目及程序文件所存放的路径 C:\Mypro\tr0101（也可以单击“浏览”按钮指定一个已有的文件夹），在“解决方案名称”框中输入解决方案名。

(3) 编写代码

系统进入代码编辑环境后，将自动生成代码框架，在 static void Main(string[] args)代码段中

输入相应的代码。

(4) 保存程序

选择菜单项“文件”→“全部保存”，系统保存当前的项目、程序。

(5) 生成程序

选择菜单项“生成”→“生成 tr0101”，进行编译、生成操作。如果程序中有语法错误，则系统会将光标停在代码编辑窗口中错误处，修改后，重新进行生成，直到生成通过。

(6) 运行程序

选择菜单项“调试”→“开始执行”或“启动调试”，运行程序，也可以直接单击工具栏中的“▶”图标运行程序。



图 1-14 实验 1.1 界面

运行结果如图 1-14 所示。

【学以致用】

实验 1.2 编写程序 tr0102，其功能是读入一个整数 n ，求 $0\sim n$ 间的所有奇数。

【探索求新】

实验 1.3 编写程序 tr0103，其功能是读入一个整数 n ，求 $0\sim n$ 间的所有整平方数（如输入 32，则输出 1, 4, 9, 16, 25）。

【挑战新高】

实验 1.4 编写程序 tr0104，其功能是读入一个整数 n ，求 $0\sim n$ 间的所有素数（只有 1 和它本身这两个因数的自然数称为素数）。

习题 1

- (1) 简述 C# 的特点。
- (2) 简述 Microsoft Visual Studio 2010 安装的硬件要求和安装过程。
- (3) 简单介绍 Microsoft Visual Studio 2010 集成开发环境。
- (4) 简述 Microsoft Visual Studio 2010 主窗口的主要组成部分，以及各部分的作用。
- (5) 简述 C# 的程序设计过程。
- (6) 如何通过 Microsoft Visual Studio 2010 帮助系统检索问题？