第1章 C#与Visual Studio简介

学习目标

(1) 了解.NET Framework、C#的基础知识，了解面向对象程序设计的基本思想。

(2) 熟悉Visual Studio集成开发环境的界面，为后续使用Visual Studio进行程序开发打下基础。

(3) 能够编写简单的控制台应用程序和Windows窗体应用程序。

1.1 知识梳理

1.1.1 .NET Framework

.NET Framework是Microsoft为开发应用程序而创建的一个富有革命性的新平台。这样描述最有趣的地方是它的含糊不清，但这是有原因的。首先，注意它没有说“在Windows操作系统上开发应用程序”。尽管.NET Framework的Microsoft版本运行在Windows操作系统上，但以后将推出运行在其他操作系统上的版本，例如Mono，它是.NET Framework的开发源代码版本(包含一个C#编译器)，该版本可以运行在几个操作系统上，包括各种Linux版本和Mac OS。另外，还可以在个人数字助手(PDA)类设备和一些智能电话上使用Microsoft .NET Compact Framework(基本上是完整.NET Framework的一个子集)，使用.NET Framework的一个主要原因是它可以作为集成各种操作系统的方式。

另外，上面给出的.NET Framework定义并没有限制应用程序的类型。这是因为本来就没有限制。.NET Framework可以创建Windows应用程序、Web应用程序、Web服务和其他各种类型的应用程序。

.NET Framework主要包含一个非常大的代码库，可以在客户语言(如C#)中通过面向对象编程技术(OOP)来使用这些代码。这个库分为不同的模块，这样就可以根据希望得到的结果来选择使用其中的各个部分。例如，一个模块包含Windows应用程序的构件，另一个模块包含联网的代码块，还有一个模块包含Web开发的代码块。一些模块还分为更具体的子模块，例如在Web开发模块中，有用于建立Web服务的子模块。

其目的是，不同的操作系统可以根据自己的特性，支持其中的部分或全部模块。例如，PDA支持所有的核心.NET功能，但不需要某些更深奥的模块。部分.NET Framework库定义了一些基本类型。类型是数据的一种表达方式，指定其中最基础的部分(例如32位带符号的整数)，以便使用.NET Framework在各种语言之间进行交互操作。这称为通用类型系统(Common Type System，CTS)。除了支持这个库以外，.NET Framework还包含.NET公共语言运行库(Common Language Runtime，CLR)，它负责

管理用 .NET 库开发的所有应用程序的执行。

.NET Framework 的核心是其运行库的执行环境,称为公共语言运行库(CLR)或 .NET 运行库。公共语言运行库是 .NET Framework 的基础,负责提供代码管理,包括处理加载程序、运行程序的代码以及提供所有支持服务的代码。同时还强制性地实施类型安全检查,事实上,CLR 在应用程序的开发阶段与运行阶段都在起作用。

.NET兼容语言程序的编译、运行是一个全新的过程,需要经过两次编译方可变成可执行的机器指令代码。

1. 编译

源程序首先被编译成微软中间语言(Microsoft Intermediate Language,MSIL 或 IL)代码。它是一种不依赖任何 CPU 指令体系结构的汇编语言,可以快速地被转换成内部机器代码(由 JIT 编译)。编译好的 MSIL 形式的软件包是由多个叫作“装配件”的代码单元组成,它包含有中间语言代码,其中的一个“装配件”被指定为可执行程序,具有主程序的启动入口,其他“装配件”被组织成“库”,全部过程由 CLR 管理完成。

2. 执行

在执行应用程序时,CLR 首先载入中间语言代码,并将其编译为适合当前运行环境的机器代码,第一个被加载的是含有程序入口的主程序“装配件”,经编译后运行它,当需要调用其他方法时,再载入其他的“装配件”并编译、运行。当“装配件”或方法被第一次调用执行后,它就被缓存起来,以便再次调用。每一个方法只被编译一次,方法一经编译,它的入口地址就被编译好的可执行代码的地址代替。如果一个方法从来没有被调用过,则与之对应的中间语言代码就不会被编译器转换成机器代码。由于中间语言代码只在需要的时候才被编译,使得应用程序获得了很高的执行效率。这一过程就是通常所说的即时编译(Just In Time,JIT)。在这期间,CLR 会检查代码的类型和安全性,例如,运行代码时,CLR 会监视内存的使用,确保编译器不会访问无效的代码内存地址。

代码管理是 CLR 的基本原则,在 .NET环境下运行或被 CLR 管理的任何代码都被称为托管代码,外部的其他代码也可运行在 Windows 环境上,这些代码被称为非托管代码。

使用中间语言和 JIT 编译有许多好处,它能够生成独立于任何操作系统平台的代码。这种代码形式只包含对 .NET核心类型和基本类库中函数的调用,即使是在 Windows 平台下开发的应用程序,也不包含对 Windows API 和 Intel CPU 基本函数的调用。另一方面,CLR 的 JIT 编译器可将中间语言代码编译成符合运行环境的执行代码。这就说明,任何操作系统平台只要嵌入了合适的 .NET Framework,它就能支持中间语言运行。

一般来说,.NET Framework 的部署可以通过安装 Visual Studio 开发环境来完成,也可以集成在 Windows 操作系统内部,随操作系统一同被安装进计算机。.NET Framework 各版本的详细信息如表 1-1 所示。

表 1-1 .NET Framework 各版本详细信息

版本	发行日期	Visual Studio	Windows 默认安装
1.0	2002-02-13	Visual Studio .NET 2002	Windows XP Media Center Edition Windows XP Tablet PC Edition
1.1	2003-04-24	Visual Studio .NET 2003	Windows Server 2003
2.0	2005-11-07	Visual Studio 2005	
3.0	2006-11-06		Windows Vista Windows Server 2008
3.5	2007-11-19	Visual Studio 2008	Windows 7 Windows Server 2008 R2
4.0	2010-04-12	Visual Studio 2010	
4.5	2012-02-20	Visual Studio 2012 RC	Windows 8 RP Windows Server 8 RC

1.1.2 C#简述

1. 认识 C#

C#(读作 C Sharp)是微软公司发布的一种面向对象的、运行于 .NET Framework 之上的高级程序设计语言。C#是一种安全的、稳定的、简单的、优雅的,由 C 和 C++ 衍生出来的面向对象的编程语言,它在继承 C 和 C++ 强大功能的同时去掉了一些它们的复杂特性(例如没有宏以及不允许多重继承)。C#综合了 VB 简单的可视化操作和 C++ 的高运行效率,以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为 .NET开发的首选语言。C#是面向对象的编程语言。它使得程序员可以快速地编写各种基于 Microsoft .NET平台的应用程序,Microsoft .NET提供了一系列工具和服务来最大程度地开发利用计算与通信领域。

C#使得 C++ 程序员可以高效地开发程序,且因可调用由 C/C++ 编写的本机原生函数,因此绝不损失 C/C++ 原有的强大功能。因为这种继承关系,C#与 C/C++ 具有极大的相似性,熟悉类似语言的开发者可以很快地转向 C#。

2. C#能用来做什么

在学习本书的过程中,读者将慢慢体会到,C#语言可用来编写各种各样的应用程序。归纳起来,使用 C#语言及 .NET Framework,可以编写下列类型的应用程序:

(1) Windows 应用程序。创建引人注目的、快速的、强大的桌面应用程序,包括那些能够在 Web 上动态改变自身的应用程序。

(2) Web 应用程序。利用 .NET和 C#提供的新特征,如 Web 部件、主控页、主题和其他一些特征,能够在最短的时间内创建令人赞叹的、强大的、丰富的、特征全面的 Web 应用程序。

(3) Web 服务。在面向服务的架构(Service-Oriented Architecture, SOA)这一工业标准之上,通过添加业务逻辑和数据来构建 Web Services。

(4) 数据驱动的应用程序。从各种不同类型的数据源中获取和使用数据,整合来自多个数据源的数据,通过 Web 或 Windows 接口将数据显示给用户。

(5) 可移动应用程序。创建能够自动识别可移动操作平台的 Web 应用程序,这些 Web 应用程序能够适应掌上电脑和移动电话的计算及存储能力。

(6) 移动客户端应用程序。创建以掌上电脑为目标平台的应用程序,它们不仅可以与其他桌面应用程序通信,而且还可以与 Internet 通信,使用 Web 服务,并与 SQL 数据库交互。

1.1.3 Visual Studio

Visual Studio(简称 VS)是微软公司的开发工具包系列产品。VS 是一个基本完整的开发工具集,它包括了整个软件生命周期中所需要的大部分工具,如 UML 工具、代码管控工具、集成开发环境(IDE)等。所写的目标代码适用于微软支持的所有平台,包括 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight 及 Windows Phone。

Visual Studio 是目前最流行的 Windows 平台应用程序的集成开发环境。最新版本为 Visual Studio 2013 版本,基于 .NET Framework 4.5.1 。追溯其发展历史,微软于 2005 年发布了 VS 2005,于 2007 年发布了 VS 2008,此后又相继在 2010 年、2012 年、2013 年发布了 VS 2010 、VS 2012、VS 2013。

本书使用的开发环境为 VS 2008,其标志如图 1-1 所示。VS 2008 在以下 3 个方面为开发人员提供了关键改进。

图 1-1 VS 2008 标志

1. 快速的应用程序开发

为了帮助开发人员迅速创建先进的软件,VS 2008 提供了改进的语言和数据功能,例如语言集成的查询 (LINQ),各个编程人员可以利用这些功能更轻松地构建解决方案以分析和处理信息。VS 2008 还使开发人员能够从同一开发环境内创建面向多个 .NET Framework 版本的应用程序。开发人员能够构建面向 .NET Framework 2.0、3.0 或 3.5 的应用程序,意味它们可以在同一环境中支持各种各样的项目。

2. 高效的团队协作

VS 2008 提供了帮助开发团队改进协作的、扩展的和改进的服务项目,包括帮助将数据库专业人员和图形设计人员加入到开发流程的工具。

3. 突破性的用户体验

VS 2008 为开发人员提供了在最新平台上加速创建紧密联系的应用程序的新工具,这些平台包括 Web、Windows Vista、Office 2007、SQL Server 2008 和 Windows Server

2008。对于 Web、ASP.NET、AJAX 及其他新技术使开发人员能够迅速创建更高效、交互式更强和更个性化的新一代 Web 体验。

1.1.4 面向对象程序设计

面向对象程序设计(Object Oriented Programming,OOP)是一种计算机编程架构,它的一条基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。

所谓"对象"在显式支持面向对象的语言中,一般是指类在内存中装载的实例,具有相关的成员变量和成员函数(也称方法)。而类(Class)用来描述具有相同的属性和方法的对象的集合,它定义了该集合中每个对象所共有的属性和方法,对象是类的实例。

面向对象程序设计不同于传统的面向过程程序设计,它大大地降低了软件开发的难度,使编程就像搭积木一样简单,是当今计算机编程的一股势不可挡的潮流。面向对象程序设计达到了软件工程的 3 个主要目标:重用性、灵活性和扩展性。为了实现整体运算,每个对象都能够接收信息、处理数据和向其他对象发送信息。

面向对象程序设计主要包括以下几方面基本概念。

(1) 组件:数据和功能一起在运行着的计算机程序中形成的单元,组件在 OOP 计算机程序中是模块和结构化的基础。

(2) 抽象性:程序有能力忽略正在处理中信息的某些方面,即对信息主要方面关注的能力。

(3) 封装:也叫作信息封装,确保组件不会以不可预期的方式改变其他组件的内部状态。只有在那些提供了内部状态改变方法的组件中,才可以访问其内部状态。每类组件都提供了一个与其他组件联系的接口,并规定了其他组件进行调用的方法。

(4) 多态性:组件的引用和类集会涉及其他许多不同类型的组件,而且引用组件所产生的结果得依据实际调用的类型。

(5) 继承性:允许在现存的组件基础上创建子类组件,统一并增强了多态性和封装性。典型地来说就是用类来对组件进行分组,而且还可以定义新类为现存的类扩展,这样就可以将类组织成树形或网状结构,这体现了动作的通用性。

面向对象程序设计的特征是:封装、继承、多态、抽象。

1.2 任务实现

1.2.1 任务 1:安装 Visual Studio 2008

【任务描述】

安装 Visual Studio 2008 至计算机,为后续开发程序做好准备。

【任务实施】

(1) 双击 Visual Studio 2008 的安装文件,显示界面如图 1-2 所示。

图 1-2　Visual Studio 2008 开始安装时的界面

(2) 单击其中的“安装 Visual Studio 2008”链接，进行安装，安装程序自动加载安装组件，如图 1-3 所示。

图 1-3　安装程序正在加载安装组件

(3) 在如图 1-4 所示的界面单击“下一步”按钮。

(4) 在如图 1-5 所示的界面选择“我已阅读并接受许可条款”，接着单击“下一步”按钮。

图 1-4 在此界面单击“下一步”按钮

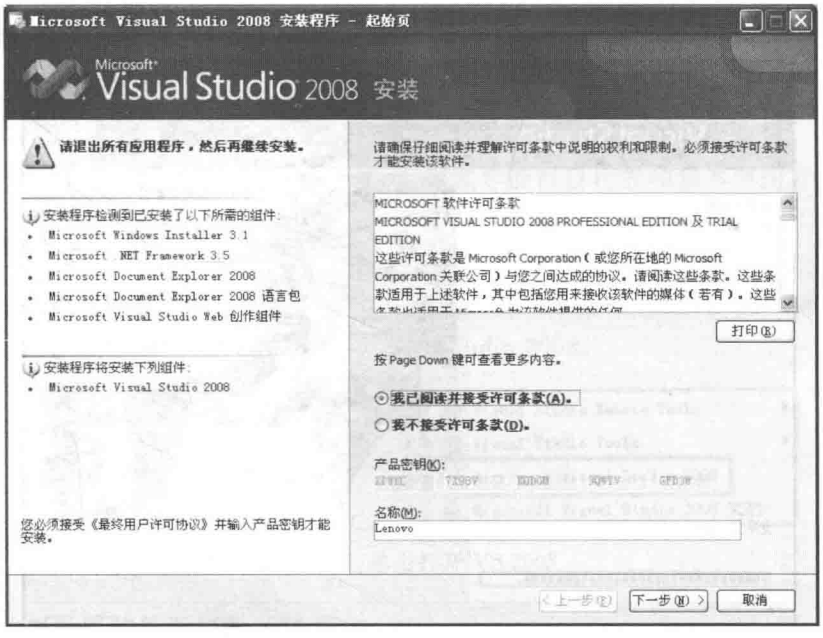

图 1-5 选择“我已阅读并接受许可条款”

(5) 在如图 1-6 所示的界面左栏选择要安装的功能(默认值、完全、自定义)，用户可根据自身需要灵活选择。在右栏可以选择安装的路径，右下方给出了安装软件所需的磁盘空间，并列出了计算机各磁盘分区的剩余空间大小，给用户以提示和参考。上述选择均

完毕后，单击“安装”按钮。

图 1-6　选择要安装的功能和安装路径

(6) VS 2008 已经开始自动安装，如图 1-7 所示。

图 1-7　自动安装

(7) VS 2008 安装完成，如图 1-8 所示。

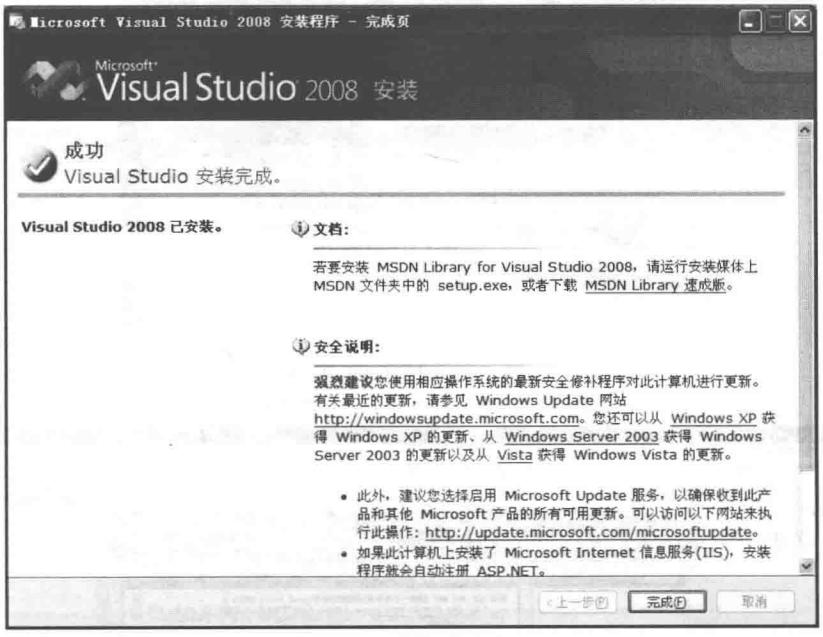

图 1-8 安装完成

1.2.2 任务 2：熟悉 Visual Studio 2008

【任务描述】

熟悉 Visual Studio 2008 的操作界面，掌握其界面栏目构成和基本使用方法。

【任务实施】

(1) 从操作系统的“开始”菜单中找到“Microsoft Visual Studio 2008”，向右展开，单击图 1-9 中圈出的菜单项，就可以打开 Visual Studio 2008。

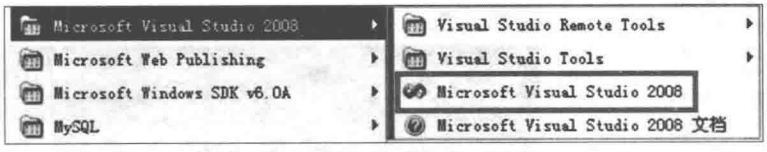

图 1-9 单击打开 VS 2008

(2) 首先看到的界面如图 1-10 所示。

(3) 接下来是 VS 2008 的起始页，如图 1-11 所示。

(4) 选择最上方“文件”菜单下的“新建项目”命令，就可以根据用户需要，新建满足需求的项目，如图 1-12 所示。

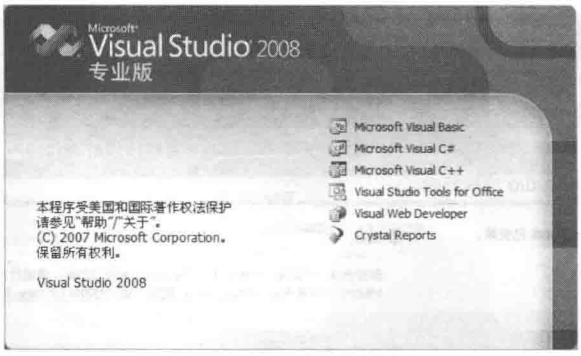

图 1-10　打开 VS 2008

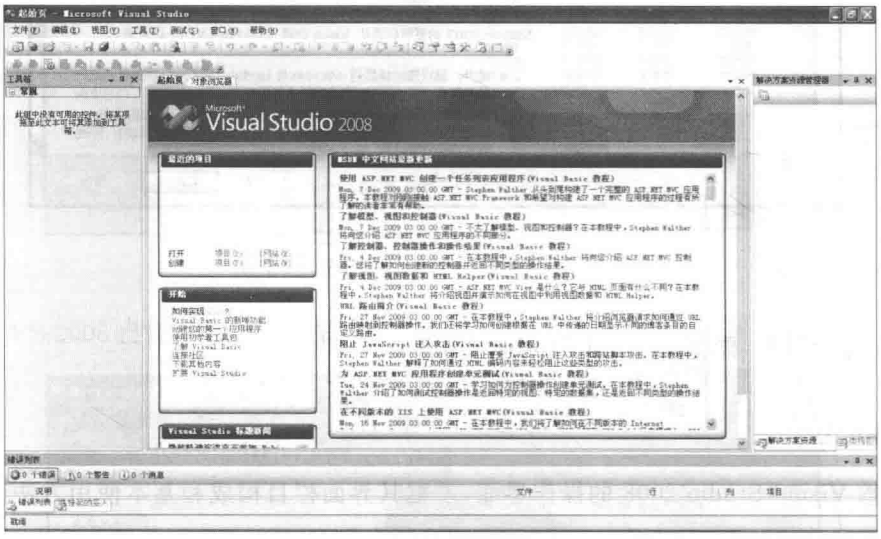

图 1-11　VS 2008 起始页

图 1-12　新建项目

(5) 接下来可见到如图 1-13 所示窗口，左侧列出了可供选择的几种编程语言以及其下包括的项目类型，当编程语言选择了 Visual C# 之后，右侧会列出可供选择的项目模板，右上方则显示项目是基于 .NET Framework 哪一版本。

图 1-13 项目模板

(6) 当选中项目模板中的“控制台应用程序”后，窗口下方会出现项目名称一栏，在这里可以修改项目的命名，然后单击“确定”按钮，如图 1-14 所示。

图 1-14 选择控制台应用程序模板