第1章 .NET 框架与 C#概述

项目背景

在学习C#语言前,必须先对其有一个初步认识。为帮助读者了解C#的开发环境、 C#的基本程序结构,我们在项目实施中,通过一个简单程序的设计,介绍.NET的编程界 面,C#的编程环境以及程序调试的基本方法。在学习中要注意C#语言与C语言和 Java语言的区别,在比较中学习。

项目任务

- (1) 任务 1-1 第一个 C # 程序
- (2) 任务 1-2 注册用户信息

知识目标

- (1) 了解.NET 框架的基本概念、C # 的基本概念。
- (2) 理解 C # 的程序结构。
- (3) 理解命名空间的概念。
- (4) 熟悉控制台程序数据输入/输出的方法。

技能目标

掌握 C#语言编程环境与程序调试的基本方法。

关键词

.NET 框架(.NET framework),应用(application),开发(developer),构建(builder),调试 (debug),控制台(console),项目(project),命名空间(namespace),解决方案(solution),运行时 (runtime),.NET 类库(.NET library),编译器(compiler)

1.1 Visual Studio .NET 简介

Microsoft.NET(以下简称.NET)框架是微软提出的新一代软件开发模型,C#语言 是.NET框架中新一代的开发工具。C#语言是一种面向对象的语言,它简化了C++语 言在类、命名空间、方法重载和异常处理等方面的操作,摒弃了C++的复杂性,更易使用, C#面向对象程序设计与项目实践

更少出错。它使用组件编程,和 VB 一样容易使用。

1.1.1 .NET 概述

1. 什么是.NET

2000年6月22日,微软正式对外宣布.NET 战略。.NET 是微软推出的一个全新的 概念,微软公司总裁兼首席执行官史蒂夫 · 鲍尔默说: ".NET 代表了一个集合、一个环 境、一个编程的基础结构,可以作为一个平台来支持下一代的互联网。.NET 也是一个用 户环境,是一组基本的用户服务,可以作用于客户端、服务器端和任何地方,具有很好的一 致性,并有新的创意。"因此,它不仅是一个用户的体验,而且是开发人员体验的集合。

2. .NET 的组成

.NET 以公共语言运行库(common language runtime, CLR)为基础,实现跨平台和 跨语言开发。.NET 由 5 个主要部分组成。

(1) Windows .NET 操作系统:它是可以运行.NET 程序的操作系统的总称,如 Windows 7/8/10 等,还提供各种应用软件服务(如 IIS、Active Directory 等)。

(2).NET 企业级服务器: 它主要包括 SQL Server 等。

(3).NET Web 服务组件:.NET 提供一系列高度分布、可编程的公共性网络服务, 可以从任何支持 SOAP 的平台上访问.NET 服务组件。

(4).NET 框架: 它是.NET 的核心部分,提供建立和运行.NET 应用程序所需的编辑、编译等核心服务。

(5) Microsoft Visual Studio .NET: 它是为建立基于.NET 框架应用程序而设的一个可视化的集成的开发环境(integrated development environment,IDE)。它为所有的编程语言提供一个简单统一的代码编辑器,包括 XML 编辑器、SQL Server 接口、以图形化的方法设计服务器端构件的设计器、监控远程机器的 Server Explorer 等。

1.1.2 .NET 框架简介

.NET 框架是一个集成在 Windows 中的组件,是一套语言无关的应用程序开发框架,它的主要特色是:简化应用程序的开发复杂性、采用系统虚拟机运行的编程平台、以公共语言运行库为基础,提供一个一致的开发模型,支持多种语言(如 C #、VB、C ++、Python 等)的开发,开发人员可以选择任何支持.NET 的编程语言来进行多种类型的应用程序开发。它支持生成和运行下一代应用程序与 XML Web 服务。

.NET 框架提供了一个语言无关的 CLR 来管理各种代码的执行过程,并为所有的 .NET 语言开发各种应用和服务提供了框架类库(framework class library, FCL)。FCL 包括基础类库(BCL)和用户接口库。.NET 框架包括以下组件,如图 1-1 所示。

(1) 公共语言运行库(CLR)。

- (2) 基础类库(BCL)。
- (3) 数据库访问组件(ADO.NET 和 XML)。
- (4) 基于 ASP.NET 编程框架的 Web 服务(Web serveice)和 Web 表单(Web form)。
- (5) Windows 桌面应用界面编程组件(WinForm)。

	无论是WinForms还是ASP.NET都可以使用ADO.NET与基础类库							
.NET框架类库								
ASP.NET								
Web表单 Web服	务 WinForms							
ADO.NET XML类	文件IO类							
基础框架类								
С	LR							

图 1-1 .NET 框架的组成

CLR 和 BCL 是.NET 框架的两个主要组成部分。

1. 公共语言运行库

公共语言运行库是.NET 框架的基础。CLR 是应用程序的执行引擎和功能齐全的类 库,该类库严格按照 CTS 规范实现。作为程序执行引擎,CLR 负责安全地载入和运行用 户程序代码。CLR 提供一个执行时的管理环境,提供内存管理、线程管理和远程处理以 及类型安全检查等核心服务。通常在 CLR 监控之下运行的代码,称为托管代码 (managed code)。

CLR 的两个组成部分如下。

(1) CTS(common type system,通用类型系统),定义了在微软中间语言 IL 中的数据类型。

CTS 不但实现了 COM 的变量兼容类型,而且定义了通过用户自定义类型的方式来进行类型扩展。任何以.NET 平台作为目标的语言必须建立数据类型与 CTS 类型之间的映射。所有.NET 语言共享这一类型系统,实现它们之间无缝的互操作。

(2) CLS(common language specification,公共语言规范),包括几种面向对象的编程 语言的通用功能。

.NET 通过定义 CLS 解决各种不同语言引发的互操作性问题。CLS 制定了一种以 .NET 平台为目标的语言所必须支持的最小特征以及该语言与其他.NET 语言之间实现 互操作性所需要的特征,以实现各种语言在同一平台下组件的相互操作。例如,CLS 并 不去关心一种语言用什么关键字实现继承,只关心该语言如何支持继承。

如果想要不同的语言在.NET CLR 上执行,就必须提供一个编译器将各种语言的程序编译成.NET CLR 所认识的元数据(metadata)以及中间语言,以符合通用类型系统

3

(CTS)的规定。然后通过即时编译器"翻译"平台专用语言,这样就使得各种不同的语言可以在同一平台上运行,如图 1-2 所示。



图 1-2 不同的语言在.NET CLR 上的执行过程

2. .NET 类库

.NET 框架也具有一套与公共语言运行库紧密集成的类库,它是一个综合性的面向 对象的可重用类型集合,使用该类库可以创建多种类型的应用程序,这些应用程序包括传 统的命令行或图形用户界面 (GUI) 应用程序,也包括基于 ASP.NET 所提供的 Web 表 单和 XML Web 服务应用程序。

1.1.3 C#编程语言简介

1. .NET 编程语言介绍

在公共语言运行库(CLR)环境下,开发人员可以选择任何支持.NET 框架的语言来进行 应用程序的开发,如 VB.NET、C[#]、J[#]以及一些第三方公司推出的语言。由于这些语言运 行在相同的公共语言运行库(CLR)执行环境下,可以有效地解决多语言之间的代码整合的 问题。

2. C # 程序设计语言

C[#](读作 C Sharp)是一种简单易用的新式编程语言,不仅面向对象,还具有类型安全 性。C[#]源于 C 语言系列,是一种面向对象的语言,还支持面向组件的编程。当代软件设计 越来越依赖采用自描述的独立功能包形式的软件组件,此类组件的关键特征包括为编程模 型提供属性、方法和事件;包含提供组件声明性信息的特性;包含自己的文档。C[#]能直接支 持这些概念,这使它成为一种非常自然的语言,可用于创建和使用软件组件。

除此之外,C#还具有以下功能。

(1) 垃圾回收机制可自动回收无法访问的未使用对象占用的内存。

(2) 异常处理。提供了一种结构化的可扩展方法来执行错误检测和恢复。

(3) 类型安全设计禁止读取未初始化的变量、为范围之外的数组编制索引或执行未 检查的类型转换。

4

(4) C # 采用统一的类型系统,所有 C # 类型(包括 int 和 double 等基础类型)均继承 自一个根——object 类型。因此,所有类型共用一组通用运算,任何类型的值都可以一致 地进行存储、传输和处理。此外,C # 还支持用户定义的引用类型和值类型,从而支持对 象动态分配以及轻量级结构的内嵌式存储。

1.1.4 Visual Studio .NET 集成环境

Microsoft Visual Studio(以下简称 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、集成开发环境等,是目前流行的 Windows 平台应用程序的集 成开发环境。用户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 框架、.NET Compact 框架和 Microsoft Silverlight 及 Windows Phone。

Visual Studio 系列产品共用一个集成开发环境,它由若干元素组成:菜单栏、标准工 具栏以及停靠或自动隐藏在左侧、右侧、底部和编辑器空间中的各种工具箱。可用的工具 箱、菜单和工具栏取决于所处理的项目或文件类型。

根据所应用的设置以及随后执行的任何自定义动作,IDE 中的工具箱及其他元素的 布置会有所不同。可以使用可视的菱形引导标记轻松移动和停靠窗口,或使用自动隐藏 功能临时隐藏窗口。

1. 起始页

Visual Studio 2010 起始页分为三个主要部分:显示"新建项目"和"打开项目"命令的命令部分;"最近的项目"列表;包含"入门"选项卡和"RSS 源"选项卡的选项卡式内容 区域。页面底部是设置起始页显示时间的选项,如图 1-3 所示。



图 1-3 Visual Studio 2010 的起始页

2. 项目

可以通过选择"文件"→"项目"命令或单击起始页的"新建项目"链接,创建一个项目, 如图 1-4 所示。

Ø	m	yprj - Micro	osoft Visua	al Studio(ඞ්	管理员)								
文件	ŧ(F)	编辑(E)	视图(V)	项目(P)	生成(B)	调	式(D)	团队(M)	工具(T)	测试(S)	分析(N)	窗口(W)	帮助(H)
	新建	≹(N)				•	睝	项目(P)		(Ctrl+Shift+N		
	打开	F(O)				۲	*⊕	网站(W)		9	Shift+Alt+N		
	添加	到源代码管	理				*	团队项目(T))				
	添加	0(D)				•	്വ	文件(F)		(Ctrl+N		
	关闭](C)						从现有代码创建项目(E)					
×	关闭]解决方案(T)				-					_	
•	保存	₹ myprj (S)			Ctrl+S								
	myp	orj 另存为(A	A)										
	全部	阝保存(L)			Ctrl+Shift+	S							
	导出	出模板(E)											
	源代	论码管理(R)				۲							
	页面	ī设置(U)											
	打印](P)			Ctrl+P								
	帐户	□设置(I)											
	最近	E使用过的文	:(牛(F)			Þ							
	最近	6使用的项目	和解决方案	(J)		+							
x	退出	1(X)			Alt+F4								
-	-					-	1						

图 1-4 新建一个项目

系统提供了已经安装的各类模板,用户根据不同的需求选择相应的项目模板即可,如 图 1-5 所示。

新建项目							Ŷ	23
▷ 最近		.NET Fr	amework 4.5.2 👻 排序依据: 默认	值 •	!! ' ! ≡		搜索已安装模板(Ctrl+E)	ρ-
▲ 已安装		C	Windows 窗体应用程序		Visual C#	*	类型: Visual C#	
▲ 模板 ▲ Visual C#	Î		WPF 应用程序		Visual C#		用于创建命令行应用程序的项目	
▷ Windows Web		C./	控制台应用程序		Visual C#			
Android Cloud			ASP.NET Web 应用程序		Visual C#			
Extensibility iOS	'	C#	共享项目		Visual C#			
Silverlight			类库(程序包)		Visual C#	Ľ		
Workflow		 ©\	控制台应用程序(程序包)		Visual C#			
▶ Visual Basic			类库		Visual C#			
Visual F# ♦ Visual C++			类库(可移植)		Visual C#			
SQL Server Pvthon	-		Silverlight 应用程序		Visual C#	Ŧ		
▶ 联机			单击此处以联机	并查找模板。				
名称(<u>N</u>):	firstprj							
位置(L):	c:\users\admini	-		浏览(<u>B</u>)				
解决方案(<u>S</u>):	创建新解决方案			•				
解决方案名称(<u>M</u>):	firstprj] 为解决方案创建目录(<u>D)</u>] 添加到源代码管理(<u>U</u>)			
							确定	消

图 1-5 选择项目模板

这里选择控制台应用程序。

3. 解决方案资源管理器

解决方案和项目包含一些项,这些项表示创建应用程序所需的引用、数据连接、文件 夹和文件。解决方案容器可包含多个项目,而项目容器通常包含多个项。

通过解决方案资源管理器,可以打开文件进行编辑,向项目中添加新文件以及查看解 决方案、项目和项目属性,如图 1-6 所示。



图 1-6 解决方案资源管理器

4. 代码编辑器和设计器

使用哪种编辑器和设计器取决于所创作的文件或文档的类型。文本编辑器是 IDE 中的基本字处理器,而代码编辑器是基本源代码编辑器。

代码编辑器和设计器通常有两个视图:图形设计视图和代码隐藏视图或源视图。设 计视图允许在用户界面或网页上指定控件和其他项的位置。可以从工具箱中拖动控件, 并将其置于设计视图上。

5. 生成和调试工具

Visual Studio 提供了一套可靠的生成和调试工具。使用生成工具,可选择需要生成的组件、排除不想生成的组件,并确定如何生成选定的项目以及在什么平台上生成这些

项目。

生成过程即是调试过程的开始。生成应用程序的过程可帮助用户检测编译时的错误。这些错误可能包含不正确的语法、拼错的关键字和输入不匹配。"输出"窗口将显示 这些错误类型。

在应用程序生成后,可以使用调试器检测和更正在运行时检测到的问题,如逻辑错误 和语法错误。处于中断模式时,可以使用"变量"窗口和"内存"窗口等工具来检查局部变 量和其他相关数据。

【例 1-1】 编写第一个 Hello 程序。

```
namespace prj
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("hello,word!");
        }
    }
}
```

1.1.5 编写代码环境

IDE 为编写各种程序代码提供了一个很方便的环境。

1. 代码编辑窗口

在 IDE 中,一个项目的代码可以通过以下两种方式体现。

1) 设计器窗口

对于设计像 WinForm 这样的用户界面项目,IDE 提供了一个图形化用户界面的设 计环境,大大提高了开发速度。利用此环境设计一个窗体或一个服务器端对象,设计器会 自动修改代码来反映所做的界面修改。

激活设计器窗口的方法:选择"项目"→"添加 Windows 窗体"命令,或选择对应的窗体对象,系统会自动打开各种窗体、控件或组件的设计器窗口,如图 1-7 所示。

2) 代码编辑窗口

激活代码编辑窗口有以下几种方法。

(1) 选择"视图"→"代码"命令(或按 F7 键)。

(2) 在设计器中,选中窗体或控件,右击,在弹出的快捷菜单中选择"查看代码"命令。

(3) 在解决方案资源管理器中,选中窗体文件 Form.cs,单击此小窗口上方的"查看代码"按钮。

(4) 双击某窗体或控件。



图 1-7 设计器窗口

代码编辑窗口(见图 1-6)顶部的两个下拉列表框可用于选择和浏览相应的代码。左 边的列表框包括了代码中的所有类;右边的列表框包括了左边列表中当前类的方法、数据 成员、所使用的控件以及相应的事件处理方法。选中一项后,光标会定位于对应方法的代 码的第一行。

每打开一个新文件编辑,IDE 就会在代码编辑窗口上添加一个新标签,可以通过各标 签来切换文件。

2. 代码折叠

在代码编辑窗口中,IDE 的另一个有用的功能就是"代码折叠"。此项功能通过树上的"+"和"一"图标实现对代码的展开和折叠,便于查看代码的层次结构。代码编辑窗口的左边有一条灰色细线,还有+和一的节点展开和折叠状态,如图 1-8 所示。

关闭代码折叠功能:选择"编辑"→"大纲显示"→"停止大纲显示"命令。

激活代码折叠功能:选择"编辑"→"大纲显示"→"启动自动大纲显示"命令。

自定义代码编辑器状态:选择"工具"→"选项"命令,在选项卡的左侧窗口中选择"文本编辑器"选项,然后自定义设置代码编辑器的各种状态。

3. 语法导航

代码编辑器环境中,有以下语法导航功能。

(1)当输入"."时,IDE 提供的语法导航会弹出被调用对象的所有方法列表,只须选择其中所需的方法,按 Enter 键便可在代码中显示该方法的名称,而不需要手工输入。

(2) 当输入"("时,IDE 会显示一个浮动窗口,它包含了该方法的输入参数类型和个

9

マ ご 快速启动 (Ctrl+Q) firstwin - Microsoft Visual Studio(管理员) ۵ х -文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 团队(M) 工具(T) 测试(S) 登录 М 分析(N) 窗口(W) 帮助(H) G - 〇 稻 - 🖆 💾 🥐 🤊 - 🤆 - Debug - Any CPU - ▶ 启动 - ♬ _ 旨 偱 🚆 Form1.cs* + × Form1.cs [设计]* 解决方案资源管理器 ▼ 4 × 器资源管理器 C# firstwin - 🔩 firstwin.Form1 ⊕ button1 Click(objec + 0 Z + 0 4 0 0 using System.Drawing; 搜索解决方案资源管理器(👂 • using System.Linq; using System. Text; ☑ 解决方案 "firstwin" (1 ▲ using System. Threading. Tasks; C# firstwin 正具箱 using System.Windows.Forms; Properties ■■引用 ⊡namespace firstwin 数据源 App.config | { public partial class Form1 : Form Form1.cs { T Form1.Desig public Form1() T Form1.resx ł At Form1 InitializeComponent(); 4 } 解决方... 团队资... 类视图 private void button1_Click(object sender, Ever **д х** 属性 ł 1 8≣ 2J ≫ } 100 % 4.1 輸出 Д x 显示输出来源(S): 诊断中心 行 22 字符 13 光结 列 13 Ins

图 1-8 代码编辑窗口

数,以确保方法调用不会出错。

(3)当在代码窗口中输入一个单词时,IDE 会以不同的颜色显示归类到相应的类别, 包括关键字、标识符和操作符等(可以选择"工具"→"选项"命令,在弹出的对话框中设置 各类别的颜色)。

(4) IDE 会在编译前检查代码。如输入了一行不完整的代码, IDE 会在该代码行下 画红色波浪线表示有错。

(5) IDE 环境中有上下文智能帮助功能。如不能确定某关键词的用法,可选中该单词然后按 F1 键,则 IDE 会打开相应的帮助窗口。

4. 对象浏览器

对象浏览器是 IDE 中最重要的工具之一,它显示了方案中所有定义的和引用的对象 或构件信息。

激活对象浏览器的方法:选择"视图"→"其他窗口"→"对象浏览器"命令,"对象浏览 器"窗口如图 1-9 所示。

"对象浏览器"窗口的左边显示了项目中的所有.NET 类库和命名空间,单击"+"图 10