第 1 章　C# 和 Visual C# 开发环境

1.1　C＃语言简介

C＃是微软公司为配合.NET 战略推出的一门通用的面向对象的编程语言，主要用于开发运行在.NET 平台上的应用程序。C＃的语言体系都构建在.NET 框架上。

1.1.1　C＃的发展史

C＃读作 C Sharp。1998 年，Anders Hejlsberg(Delphi 和 Turbo Pascal 语言的设计者)及他的微软开发团队开始设计 C＃语言的第一个版本。2000 年 9 月，ECMA(国际信息和通信系统标准化组织)成立了一个任务组，着力为 C＃编程语言定义一个开发标准。据称，其设计目标是开发“一个简单、现代、通用、面向对象的编程语言”，于是出台了 ECMA-334 标准，这是一种令人满意的简洁的语言，它有类似 Java 的语法，但显然又借鉴了 C++ 和 C 的风格。设计 C＃语言是为了增强软件的健壮性，为此提供了数组越界检查和强类型检查，并且禁止使用未初始化的变量。C＃语言的正式发布是从 2002 年伴随着 Visual Studio 开发环境一起开始的，其一经推出，就受到众多程序员的青睐。

1.1.2　C＃的特点

C＃是从 C 和 C++ 派生出的一种简单、现代、面向对象和类型安全的编程语言，并且能够与.NET 框架完美结合，C＃具有以下突出的特点。

(1) 语法简洁，不允许直接操作内存，去掉了指针操作。

(2) 彻底的面向对象设计，C＃具有面向对象的语言所应有的一切特性：封装、继承和多态等。

(3) 与 Web 紧密结合，支持绝大多数的 Web 标准，例如，HTML、XML、SOAP 等。

(4) 强大的安全性机制，可以消除软件开发中的常见错误(如语法错误)，.NET 提供的垃圾回收器能够帮助开发者有效地管理内存资源。

(5) 因为 C＃遵循.NET 的公共语言规范(CLS)，从而保证能够与其他语言开发的组件兼容。

(6) C＃提供了完善的错误和异常处理机制，使程序在交付应用时能够更加健壮。

1.2 .NET 开发平台

1.2.1 .NET 概述

.NET 平台是由微软公司推出的全新的应用程序开发平台,可用来构建和运行新一代 Microsoft Windows 和 Web 应用程序。它建立在开放体系结构基础之上,集微软公司在软件领域的主要技术于一身。

.NET 平台的核心是.NET Framework,它为.NET 平台下应用程序的运行提供了基本框架,如果把 Windows 操作系统比作一栋摩天大楼,那么.NET Framework 就是摩天大楼中由钢筋和混凝土搭成的框架。

Visual Studio.NET 是.NET 平台的主要开发工具,由于.NET 平台是建立在开放体系结构基础之上的,因此应用程序开发人员也可以使用其他的开发工具。

1.2.2 .NET Framework 的结构

.NET Framework 以微软公司的 Windows 操作系统为基础,由不同的组件组成,能够与 Windows 的各种应用程序服务组件(如消息队列服务、COM+组件服务、Internet 信息服务、Windows 管理工具等)整合,以开发各种应用程序,如图 1-1 所示。

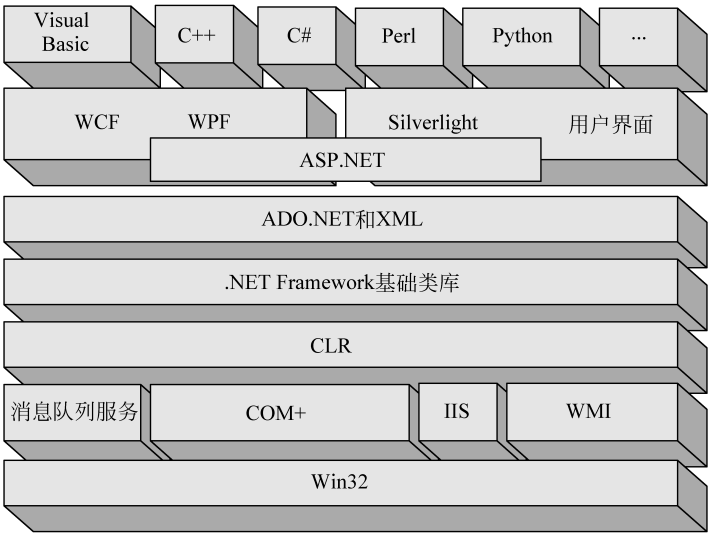

图 1-1 .NET Framework 的结构

在.NET Framework 的最顶层是程序设计语言,.NET Framework 支持诸如 Visual Basic(VB)、C#、C++、J#、Perl 等几十种高级程序设计语言。在 Visual Studio.NET 开发环境中,可直接使用 VB、C#、C++、F#、JScript 共 5 种语言开发应用程序。

.NET Framework 具有两个主要组件:公共语言运行库(Common Language Runtime,CLR)和.NET Framework 类库,除此之外还包括 ADO.NET、ASP.NET、XML Web 服务等。

CLR 是.NET Framework 的基础，是应用程序与操作系统之间的“中间人”，它为应用程序提供内存管理、线程管理和远程处理等核心服务。在.NET 平台上，应用程序无论使用何种语言编写，在编译时都会被语言编译器编译成 MSIL(微软公司中间语言)，在运行应用程序时 CLR 自动启用 JIT(Just In Time)编译器把 MSIL 再次编译成操作系统能够识别的本地机器语言代码(简称本地代码)，然后运行并返回结果，如图 1-2 所示。因此，CLR 是所有.NET 应用程序的托管环境。这种运行在.NET 之上的应用程序被称为托管应用程序，而传统的直接在操作系统基础上运行的应用程序则被称为非托管应用程序。

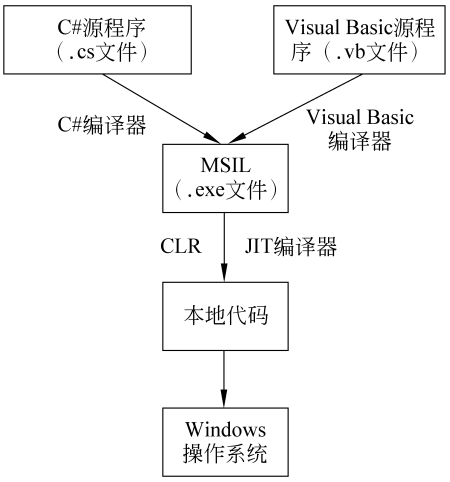

图 1-2　CLR 的工作机制

.NET Framework 类库是一个综合性的面向对象的可重用类型集合，利用它可以开发多种应用程序，包括传统的命令或图形用户界面(GUI)应用程序，也包括 Web 应用程序。

ADO.NET 是.NET Framework 提供的微软公司开发出的新一代的面向对象的数据处理技术，利用它可以简便、快捷地开发数据库应用程序。

ASP.NET 是.NET Framework 提供的全新的 Web 应用程序开发技术，利用它开发 Web 应用程序，如同开发 Windows 应用程序一样简单。

WCF (Windows Communication Foundation)、WPF (Windows Presentation Foundation)以及 Silverlight 等技术是微软公司推出的全新.NET 技术。WCF 可以理解为 Windows 通信接口，它整合了 TCP/IP、XML、SOAP 等技术，因此简化了 XML Web 服务的设计与实现。WPF 为用户界面、2D/3D 图形、文档和媒体提供了统一的描述和操作方法。Silverlight 为开发具有专业图形、音频和视频处理的 Web 应用程序提供了全新的解决方案。

1.2.3　.NET Framework 的优点

在.NET 平台诞生之前，虽然 Internet 已经出现，但很少有应用程序能够运行在各种不同类型的客户端上，也不能和其他应用程序进行无缝集成。这种局限性导致开发人员需花费大量的时间去改写应用程序，以保证它们能在各种客户端和平台上运行，而不是利用这些时间去设计新的应用程序。.NET Framework 的最大特点就在于它为应用程序开发人员提供了一个真正与平台无关的开发环境。使用.NET Framework 开发应用程序有以下优点。

1. 基于 Web 的标准

.NET Framework 完全支持现有的 Internet 技术，包括 HTML(超文本标记语言)、HTTP(超文本传输协议)、XML(可扩展标记语言)、SOAP(简单对象访问协议)、XSLT(可扩展样式表语言转换)、XPath(XML 路径语言)和其他 Web 标准。

2. 使用统一的应用程序模型

任何与.NET 兼容的语言都可以使用.NET Framework 类库。.NET Framework 为

Windows 应用程序、Web 应用程序和 XML Web 服务提供了统一的应用程序模型，因此，同一段代码可被这些应用程序无障碍地使用。

3. 便于开发人员使用

在.NET Framework 中，代码被组织在不同的命名空间和类中，而命名空间采用树形结构，方便开发人员引用。当开发人员企图调用.NET Framework 类库中的类时，只需将该类属性命名空间添加到引用解决方案中即可。

4. 可扩展类

.NET Framework 提供了通用类型系统，它根据面向对象的思想把一个命名空间或类中代码的实现细节隐藏，开发人员可以通过继承访问类库中的类，也可以扩展类库中的类，甚至构建自己的类库。

1.3 Visual C#开发环境

当计算机安装了 Visual Studio 2013 后，用户只需选择“开始”→“所有程序”下的 Visual Studio 2013 命令即可启动它。

刚启动的 Visual Studio 2013 的窗口由菜单栏、工具栏、工具箱、起始页、解决方案资源管理器等组成，如图 1-3 所示。其中，菜单栏列出了 Visual Studio 2013 的所有操作命令；工具栏列出了常见的操作命令；解决方案资源管理器用于显示将要创建的应用程序项目的文件夹结构以及文件列表；工具箱用于显示在设计应用程序操作界面时所要使用的可视化控件；起始页为常用的一些操作。

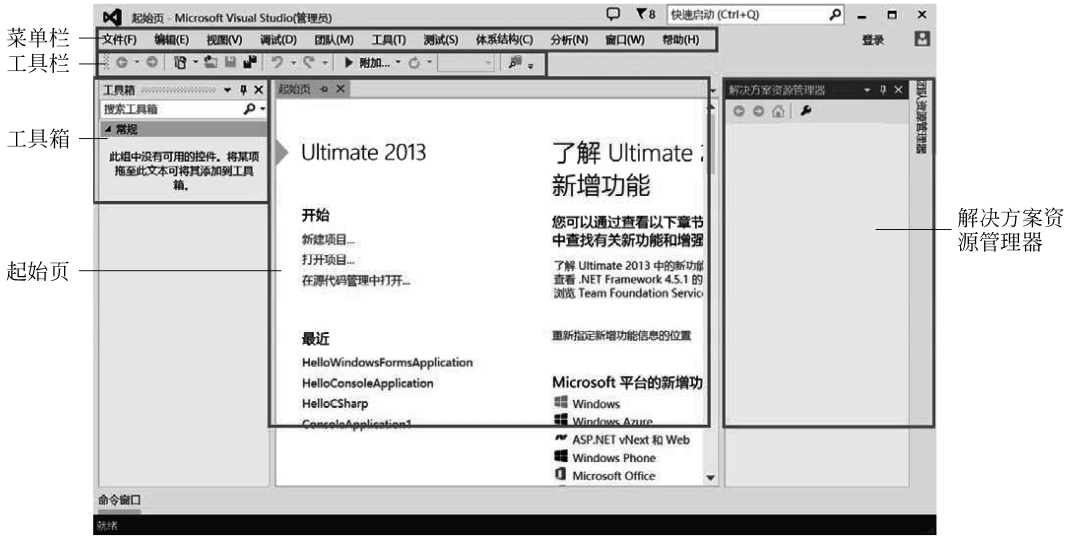

图 1-3　Visual Studio 2013 窗口

Visual Studio 2013 通过解决方案和项目来管理一个正在开发的软件项目。在 Visual Studio 2013 中，一个解决方案代表一个正在开发的庞杂的软件系统，一个项目可能只是正在开发的软件系统中的一个子系统。因此，一个解决方案可以把多个项目组织起来，而一个

项目可以把一个子系统中的所有文件管理起来。Visual Studio 2013 中支持多种文件类型及它们相关的扩展类型，表 1-1 列出了一些常用的文件类型。

表 1-1　Visual Studio 2013 中常用的文件类型

扩展名	名　称	描　　述
.sln	Visual Studio 2013 解决方案文件	.sln 文件为解决方案资源管理器提供显示管理文件的图形接口所需的信息。打开.sln 文件，能快速地打开整个项目的所有文件
.csproj	Visual C#项目文件	一个特殊的 XML 文档，主要用来控制项目的生成
.cs	Visual C#源代码文件	表示 C#源程序文件、Windows 窗体文件、Windows 用户控件文件、类文件、接口文件等
.resx	资源文件	包括一个 Windows 窗体、Web 窗体等文件的资源信息
.aspx	Web 窗体文件	表示 Web 窗体，由 HTML 标记、Web Server 控件、脚本组成
.asmx	XML Web 服务文件	表示 Web 服务，它链接一个特定的.cs 文件，在整个.cs 文件中包含了供 Internet 调用的方法函数代码

1.3.1　标题栏

标题栏是位于 Visual Studio 2013(以下简称 VS 2013)窗口顶部的水平条，它显示的是应用程序的名称。默认情况下，用户创建一个项目后，标题栏显示如下信息：

```
HelloCSharp-Microsoft Visual Studio
```

其中，HelloCSharp 表示解决方案名称。随着程序状态的变化，标题栏中的信息页随之改变。例如，当程序处于运行状态时，标题栏中显示如下信息：

```
HelloCSharp(正在运行)-Microsoft Visual Studio
```

1.3.2　菜单栏

菜单栏是 VS 2013 开发环境的重要组成部分，开发者要完成的主要功能都可以通过菜单或者与菜单对应的工具栏按钮和快捷键实现。在不同的状态下，菜单栏中的菜单项格式是不一样的，比如，控制台应用程序的菜单栏效果如图 1-4 所示，Windows 窗体应用程序的菜单栏效果如图 1-5 所示。

图 1-4　控制台应用程序的菜单栏

图 1-5　Windows 窗体应用程序的菜单栏

下面以 Windows 窗体应用程序的菜单栏为例，介绍 VS 2013 开发环境中常用的菜单。

1. “文件”菜单

“文件”菜单用于对文件进行操作，比如新建项目、网站、打开项目、网站以及保存、退出

等,如图 1-6 所示。

“文件”菜单主要的菜单项及其功能如表 1-2 所示。

表 1-2 “文件”菜单主要的菜单项及其功能

菜 单 项	功 能
新建	包括新建项目、网站、文件等内容
打开	包括打开项目、解决方案、网站、文件等内容
关闭解决方案	关闭打开的解决方案
保存选定项	保存当前项目
将选定项另存为	将当前的项目另存为其他名称
全部保存	保存当前打开的所有项目
导出模板	将项目导出为可用作其他项目的基础模板
最近的文件	通过最近打开过的文件名打开相应的文件
最近使用的项目和解决方案	通过最近打开过的解决方案或者项目名打开相应的解决方案或项目
退出	退出 VS 2013 开发环境

2.“视图”菜单

“视图”菜单主要用于显示或隐藏各个功能窗口或对话框。如果不小心关闭了某个窗口,可通过“视图”菜单中的菜单项打开。“视图”菜单如图 1-7 所示。

图 1-6 “文件”菜单

图 1-7 “视图”菜单

“视图”菜单主要的菜单项及其功能如表 1-3 所示。

表 1-3　“视图”菜单主要的菜单项及其功能

菜 单 项	功　　能
解决方案资源管理器	打开解决方案资源管理器窗口
服务器资源管理器	打开服务器资源管理器窗口
类视图	打开类视图窗口
起始页	打开起始页
工具箱	打开工具箱窗口
其他窗口	打开命令窗口、Web 浏览器、历史记录等窗口
工具栏	打开或关闭各种快捷工具栏
全屏显示	使 VS 2013 开发环境全屏显示

3. “项目”菜单

“项目”菜单主要用来向程序中添加或移除各种元素，例如添加 Windows 窗体、用户控件、组件、类、引用等。“项目”菜单如图 1-8 所示。

图 1-8　“项目”菜单

“项目”菜单主要的菜单项及其功能如表 1-4 所示。

表 1-4　“项目”菜单主要的菜单项及其功能

菜 单 项	功　　能
添加 Windows 窗体	向当前项目中添加新的 Windows 窗体
添加类	向当前项目中添加类文件
添加新项	向当前项目中添加新项(包括类、Windows 窗体、用户控件等)
添加现有项	向当前项目中添加已经有的项
添加引用	向当前项目中添加 DLL 引用
添加服务引用	向当前项目中添加服务引用(如 Web 服务引用)
设为启动项目	将当前项目设置为启动项目
HelloCSharp 属性	打开项目的属性页

4.“格式”菜单

“格式”菜单主要用来对窗体上的各个控件进行统一布局,还可以用来调整选定对象的格式。“格式”菜单如图 1-9 所示。

图 1-9 “格式”菜单

“格式”菜单主要的菜单项及其功能如表 1-5 所示。

表 1-5 “格式”菜单主要的菜单项及其功能

菜 单 项	功 能
对齐	调整所有选中控件的对齐方式
使大小相同	使所有选中的控件大小相同
水平间距	调整所有选中控件的水平间距
垂直间距	调整所有选中控件的垂直间距
在窗体中居中	使选中的控件在窗体中居中显示
顺序	使选中的控件按照前后顺序放置
锁定控件	锁定选中的控件,不能调整位置

5.“调试”菜单

“调试”菜单主要用于选择不同的调试程序的方法,比如启动调试、开始执行(不调试)、逐语句、逐过程、新建断点等。“调试”菜单如图 1-10 所示。

图 1-10 “调试”菜单

“调试”菜单主要的菜单项及其功能如表 1-6 所示。

表 1-6　“调试”菜单主要的菜单项及其功能如

菜 单 项	功　　能
启动调试	以调试模式运行程序
开始执行(不调试)	不调试程序,直接运行
逐语句	逐句地执行程序
逐过程	逐个过程地执行程序(这里的过程以独立的语句为准)
新建断点	设置断点
删除所有断点	清除所有已经设置的断点

6.“工具”菜单

“工具”菜单主要用来选择在开发程序时的一些工具,比如连接到数据库、连接到服务器、选择工具箱项、导入和导出设置、自定义、选项等。“工具”菜单如图 1-11 所示。

图 1-11　“工具”菜单

“工具”菜单主要的菜单项及其功能如表 1-7 所示。

表 1-7　“工具”菜单主要的菜单项及其功能

菜 单 项	功　　能
连接到数据库	新建数据库连接
导入和导出设置	重新对开发环境的默认设置进行设置
选项	打开“选项”对话框,以便对 VS 2013 开发环境进行设置,比如代码的字体、字体大小、代码行号的显示等

7.“生成”菜单

“生成”菜单主要用于生成可以运行的可执行文件,生成之后的程序可以脱离开发环境独立运行(需要.NET Framework 框架)。“生成”菜单如图 1-12 所示。

图 1-12 “生成”菜单

“生成”菜单主要的菜单项及其功能如表 1-8 所示。

表 1-8 “生成”菜单主要的菜单项及其功能

菜 单 项	功 能	菜 单 项	功 能
生成解决方案	生成当前解决方案	清理 HelloCSharp	清理已生成的项目
清理解决方案	清理已生成的解决方案	发布 HelloCSharp	对当前项目进行发布
生成 HelloCSharp	生成当前项目		

8.“帮助”菜单

“帮助”菜单主要帮助用户学习和掌握 C# 方面相关内容,例如,用户可以通过内容、索引、搜索、MSDN 论坛等方式寻求帮助。“帮助”菜单如图 1-13 所示。

图 1-13 “帮助”菜单

“帮助”菜单主要的菜单项及其功能如表 1-9 所示。

表 1-9 “帮助”菜单主要的菜单项及其功能

菜 单 项	功 能
查看帮助	打开本地安装的帮助文档
添加和移除帮助内容	安装和卸载本地的帮助文档