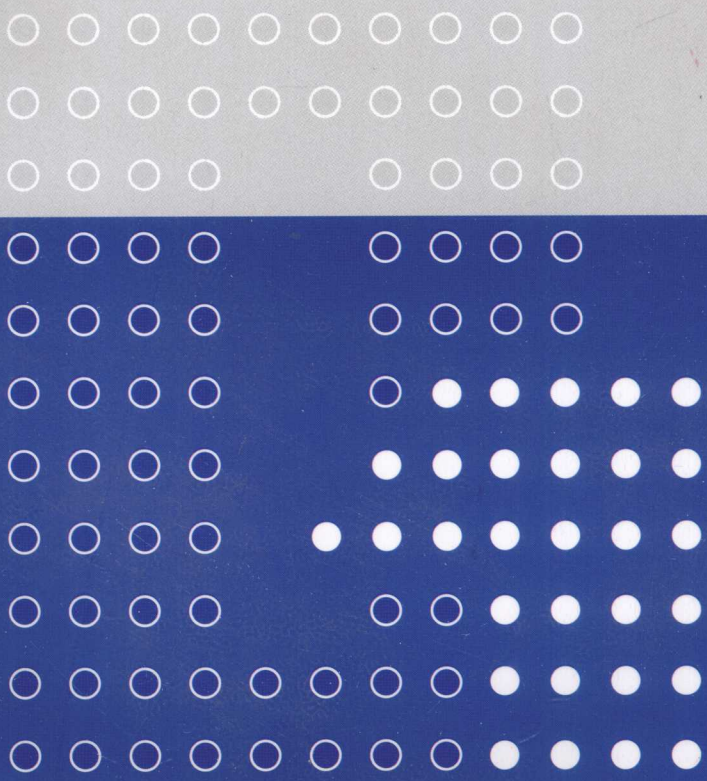


计算机系列教材

2

.NET 框架程序设计



张宏鸣 王美丽 朱珊珊 蔚继承 编著

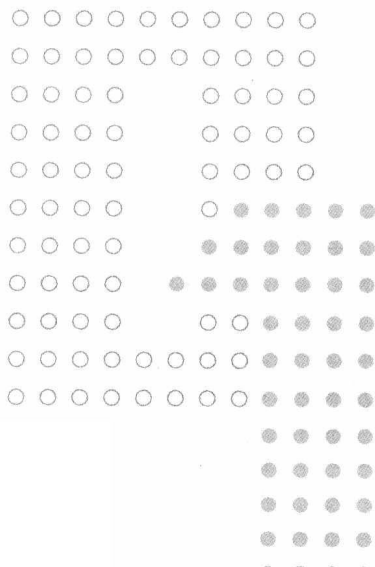
清华大学出版社



计算机系列教材

张宏鸣 王美丽 朱珊珊 蔚继承 编著

.NET 框架程序设计



清华大学出版社
北京

内 容 简 介

微软公司的 .NET 技术涉及桌面、平板和移动端等多个平台,尤其随着 Windows 10 的发布,.NET 开发环境可以跨多种平台,再加上 .NET 框架的开源特性,.NET 的应用必将越来越广泛。本书适合有一定编程基础的读者学习。本书没有详细介绍 C# 语法,而是将其中较为重要的方面进行了介绍。

为了介绍基于 .NET 的开发方法,作者根据多年的教学经验,将 .NET 能够完成的工作分为 10 章进行介绍。第 1 章主要是对 .NET 技术的简介;第 2 章介绍 C# 语言所具有的与 C++ 和 Java 不同的特性;第 3 章介绍基于 Windows 操作系统的窗体程序设计方法及过程;第 4 章在第 3 章的基础上,介绍微软最新的界面设计方法 WPF;第 5 章介绍文件的相关操作;第 6 章介绍多线程的使用;第 7 章介绍网络通信部分——套接字的使用方法;第 8 章介绍网页设计方法;第 9 章介绍数据库的增、删、改等操作,以及 XML 和 LINQ 的使用方法;第 10 章介绍 .NET 框架下的 Web 服务调用方法。各部分均结合作者多年的教学、科研工作中总结的开发实例。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

.NET 框架程序设计/张宏鸣等编著. --北京:清华大学出版社,2016
计算机系列教材
ISBN 978-7-302-44637-8

I. ①N… II. ①张… III. ①互联网络—程序设计—高等学校—教材 IV. ①TP393.4

中国版本图书馆 CIP 数据核字(2016)第 209389 号

责任编辑:张 民
封面设计:常雪影
责任校对:焦丽丽
责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:22.25 字 数:512 千字

版 次:2016 年 10 月第 1 版 印 次:2016 年 10 月第 1 次印刷

印 数:1~2000

定 价:45.00 元

产品编号:066942-01

微软公司吸收了多种语言开发的精华,将最新的开发工具、开发方法都集成到 Visual Studio 中。随着互联网产业不断发展,微软的平台也向 PC 以外的平台扩展,例如平板、手机、物联网,涉及人们生活中的方方面面,而这一切开发工具都集成在了 IDE 开发环境中。伴随着 Windows 10 的发布,微软对 .NET 平台有了更加开放的心态,包括:

(1) .NET Innovation: 带来了平台推动与创新,其中包括语言、编译器、基类库、应用模型、运行时以及工具。

(2) Open Source: 将工程进程引入一个开放环境,在社区的帮助支持下开发,繁殖一个富有生命力的生态系统。许多 .NET 2015 的部件是开源的,并由 .NET 基金会指导工作。 .NET 基金会支持的所有工程的 repos 和 activity 如下:

GitHub: <http://dotnet.github.io/>

ASP.NET 5: <https://github.com/aspnet/Home>

.NET Core CLR: <https://github.com/dotnet/coreclr>

.NET Core Framework: <https://github.com/dotnet/corefx>

Roslyn: <https://github.com/dotnet/roslyn>

(3) Cross platform: 拓展 .NET 到 Linux 和 Mac 平台,因为消费者逐渐地在多样的环境中工作。

尽管 Java 是目前面向对象开发的主流语言,但是以 C# 为基础的 .NET 开发环境也受到了开发人员的好评。目前市面上介绍 C# 的书很多,但将微软公司最新的成果集成进来的教材却很少,本书是针对学习面向对象程序设计之后,想了解或者学习 .NET 开发技术的学生、工作人群设计的。本书主要内容包括 C# 面向对象开发技术的特征,并将多线程、套接字、WPF、ASP.NET、WCF 等微软的最新技术也融合进去,基于 VS 2013 展示了不同方法的设计过程,特别适合有一定面向对象语言基础的读者学习使用。

本书作者从事 .NET 软件开发工作已经 10 年了,从事科研教学工作时大都采用 .NET 技术,当然也见证了 Visual Studio .NET 2003, Visual Studio 2005, Visual Studio Team System 2008, Visual Studio 2010 Ultimate, Visual Studio 2013 一系列版本的变化。 Visual Studio .NET 2003 极力推广的 Web Services 虽然没有很流行,但是它开发 ASP.NET 的编译模型很快得到公司的技术部认可,完全重写了原来的 ASP 系统,本书作者用这个工具解决了开发工作中很多棘手的问题。

本书适合作为高等学校计算机类专业面向对象程序设计相关课程的教材,也可作为

具有一定面向对象编写经验的技术人员的参考书。

本书第 1、2 章由蔚继承编写；第 3、4、7、8 章由张宏鸣编写，第 5、6 章由王美丽编写，第 9、10 章由朱珊珊编写。在本书编写的过程中得到了于尚尚、李瑶、王猛、宋泽鲁、熊志国、卢鹏浩、郭晓毅等同学的大力协助，在此表示感谢！

在本书的编写过程中，陕西微软创新中心有限公司、西咸新区管委会在技术、软件、内容方面给予了很大支持，在此表示感谢！

由于编者能力有限，错误和不当之处在所难免，欢迎大家提出宝贵意见。

张宏鸣

2016 年 5 月于杨凌

F O R E W O R D

第 1 章 .NET 概述	/1
1.1 .NET Framework	/1
1.1.1 .NET 发展历史	/2
1.1.2 .NET 框架与公共语言运行库	/3
1.2 Visual Studio 2013 IDE 简介	/4
1.2.1 Visual Studio 开发环境	/4
1.2.2 创建项目	/5
1.2.3 运行和调试	/7
习题	/8
第 2 章 .NET 特性	/9
2.1 继承与接口	/9
2.1.1 继承	/9
2.1.2 接口	/11
2.2 属性与索引器	/12
2.2.1 属性	/12
2.2.2 索引器	/13
2.3 委托与事件	/15
2.3.1 委托	/15
2.3.2 异步委托	/16
2.3.3 事件	/17
2.4 字符串与正则表达式	/18
2.4.1 字符串	/18
2.4.2 正则表达式	/19
2.5 集合	/21
2.6 泛型	/22
2.6.1 定义泛型集合	/22
2.6.2 定义泛型类	/22
2.6.3 泛型方法	/23

2.7 异常 /23

 2.7.1 捕获异常 /24

 2.7.2 finally 块 /25

 2.7.3 抛出异常 /25

习题 /26

第 3 章 Windows 窗体设计 /27

3.1 Windows 窗体与控件 /27

 3.1.1 Windows 窗体应用程序 /27

 3.1.2 控件 /29

3.2 常用控件 /31

 3.2.1 Button, Label 和 LinkLabel /31

 3.2.2 TextBox 和 RichTextBox /34

 3.2.3 CheckBox 和 RadioButton /37

 3.2.4 ListBox 和 CheckedListBox /38

 3.2.5 ListView /41

 3.2.6 TreeView /46

 3.2.7 ComboBox /50

3.3 容器 /52

 3.3.1 Panel /52

 3.3.2 SplitContainer /54

 3.3.3 TabControl /56

 3.3.4 FlowLayoutPanel 和
 TableLayoutPanel /59

3.4 菜单和工具栏 /62

 3.4.1 MenuStrip /62

 3.4.2 ToolStrip /65

 3.4.3 ContextMenuStrip 和 StatusStrip /67

3.5 对话框 /69

 3.5.1 OpenFileDialog 和 SaveFileDialog /69

 3.5.2 ColorDialog 和 FontDialog /73

 3.5.3 FolderBrowserDialog /76

- 3.6 其他控件 /78
 - 3.6.1 ProgressBar 和 Timer /78
 - 3.6.2 NotifyIcon 和 ToolTip /80
 - 3.6.3 ImageList 和 PictureBox /82
 - 3.6.4 MonthCalendar, DateTimePicker 和
NumberricUpDown /85
- 3.7 多文档界面简介 /91
 - 3.7.1 MDI 概述 /91
 - 3.7.2 创建多文档界面 /91
 - 3.7.3 使用多文档界面 /92
- 习题 /94

第 4 章 WPF /95

- 4.1 WPF 概述 /95
 - 4.1.1 什么是 WPF /95
 - 4.1.2 WPF 的体系结构 /96
 - 4.1.3 第一个 WPF 应用程序 /98
- 4.2 XAML /100
 - 4.2.1 XAML 概述 /100
 - 4.2.2 XAML 基础 /101
 - 4.2.3 XAML 属性和事件 /103
- 4.3 依赖属性和路由事件 /108
 - 4.3.1 依赖属性 /108
 - 4.3.2 路由事件 /113
 - 4.3.3 在 WPF 中使用事件 /118
- 4.4 布局 /123
 - 4.4.1 WPF 布局概述 /123
 - 4.4.2 StackPanel 布局 /125
 - 4.4.3 WarpPanel 布局和
DockPanel 布局 /128
 - 4.4.4 Grid 布局 /130
 - 4.4.5 Canvas 布局 /136

4.5	控件	/137
4.5.1	控件概述	/137
4.5.2	内容控件	/139
4.5.3	文本控件	/146
4.5.4	列表控件	/150
4.5.5	范围控件	/153
4.5.6	日期控件	/155
4.5.7	菜单和工具栏	/157
	习题	/162

第 5 章 文件与流 /163

5.1	文件与流操作概述	/163
5.2	File 和 FileInfo 类	/164
5.3	Directory 和 DirectoryInfo 类	/166
5.4	文件的读写	/170
5.4.1	FileStream 类	/170
5.4.2	文本文件的读写	/172
5.4.3	二进制文件的读写	/176
5.5	驱动器的操作	/179
	习题	/181

第 6 章 多线程 /182

6.1	线程的概述	/182
6.1.1	线程与进程	/182
6.1.2	.NET 中多线程的特点	/182
6.2	线程的基本操作	/183
6.2.1	线程的创建和启动	/184
6.2.2	线程的挂起和中断	/185
6.2.3	线程的终止	/186
6.3	线程的优先级与调度	/186
6.4	线程的状态和状态转换	/189
6.5	线程同步	/191

6.5.1	lock 语句与线程安全	/192
6.5.2	监视器与线程安全	/194
6.5.3	等待句柄与同步事件	/196
6.5.4	Mutex 类与线程安全	/199
6.5.5	Interlocked 类与线程安全	/202
6.5.6	ReaderWriterLock 类与线程安全	/202
6.6	线程池	/206
	习题	/208
第 7 章	套接字编程	/209
7.1	套接字编程基础	/209
7.1.1	.NET 的网络组件	/209
7.1.2	网络编程中的常用类	/210
7.2	套接字编程	/212
7.2.1	套接字简介	/212
7.2.2	.NET 中的 Socket 类	/212
7.2.3	基于 Socket 类的编程实例	/214
7.3	TCP 编程	/217
7.3.1	TcpClient 类	/218
7.3.2	TcpListener 类	/219
7.3.3	基于 TcpClient 和 TcpListener 的编程实例	/220
7.4	UDP 编程	/222
7.4.1	UdpClient 类	/222
7.4.2	基于 UdpClient 类的编程实例	/224
	习题	/225
第 8 章	ASP.NET 4.0	/226
8.1	ASP.NET 简介	/226
8.1.1	ASP.NET 的工作原理	/226
8.1.2	ASP.NET 运行环境	/227
8.2	ASP.NET 的开发模式	/231

8.2.1	Web Forms	/231
8.2.2	Web Pages	/231
8.2.3	MVC	/232
8.3	ASP.NET 常用控件	/233
8.3.1	Label 控件	/233
8.3.2	TextBox 控件	/234
8.3.3	Button 控件	/234
8.3.4	GridView 控件	/235
8.4	ASP.NET 与常见开源框架的结合	/239
8.4.1	FineUI 开源框架	/239
8.4.2	ECcharts 图表	/262
	习题	/264

第9章 数据库、XML 和 LINQ /265

9.1	ADO.NET 数据库编程	/265
9.1.1	ADO.NET 模型	/265
9.1.2	使用 ADO.NET 连接和 操作数据库	/269
9.1.3	DataGridView 控件	/282
9.2	XML 及其应用	/284
9.2.1	XML 概述	/284
9.2.2	文档结果与基本语法	/286
9.2.3	XML 操作	/289
9.3	LINQ 技术	/297
9.3.1	LINQ 概述	/297
9.3.2	LINQ 的查询简介	/298
9.3.3	LINQ 的基本查询操作	/301
9.3.4	LINQ to SQL	/306
9.3.5	LINQ to DataSet	/308
9.3.6	LINQ to XML	/312
	习题	/317

第 10 章 Windows 服务	/318
10.1 WCF	/318
10.1.1 WCF 简介	/318
10.1.2 在 Visual Studio 中开发 WCF	/321
10.2 WF	/329
10.2.1 WF 简介	/329
10.2.2 在 Visual Studio 中开发 WF	/330
习题	/339
参考文献	/341

第 1 章 .NET 概述

本章提示：本章介绍 .NET 框架的发展历程和框架概览，并学习如何使用 Visual Studio 创建和开发 .NET 项目。此外，对 C# 编程语言做了简要介绍。

1.1 .NET Framework

.NET Framework 是 2000 年微软公司推出的一个普及的开发平台，把 Internet 与万维网集成到软件开发、工程、发布与使用中。.NET 项目不强迫开发人员使用一种编程语言，.NET 体系结构适用于多种平台，不局限于 Microsoft Windows 系统。

该框架用于构建 Windows、Windows Store、Windows Phone、Windows Server 和 Windows Azure 的应用程序。.NET Framework 平台包括 Visual C#、Visual C++ 和 Visual Basic 编程语言、公共语言运行库(Common Language Runtime, CLR)和广泛的类库(Framework Class Library, FCL)。

.NET 战略将软件复用思想扩展到 Internet，使程序员可以集中考虑自己的专业，而不必实现每个应用程序的每个组件。公司可以购买 Web 服务，即基于 Web 的程序，将其加进自己的系统中，从而加速 Web 应用程序开发过程。

.NET 是新的组件技术革命，它和下一版本的 Visual Studio 一起提供新的应用程序框架(如 Web Services、ASP.NET、Windows Forms、Windows Phone 和 ADO.NET)。作为微软公司的下一代组件技术，.NET 设计范围从底层一直到简化组件开发和发布，同时提供空前规模的交互性能。

.NET 基于 CLR 环境，这个环境管理代码运行时的每个方面，即所有 .NET 组件，不论它是何种开发语言，都在同一个 Runtime 运行。CLR 就像一个温暖的毛毯包围着你的代码，它提供了内存管理、安全的运行环境、对象位置透明、并行管理和访问底层的操作系统服务。因为 CLR 管理项目行为的这些方面，所以在 CLR 上运行的代码叫“受管制代码”。CLR 提供了空前规模的语言交互性，允许组件重用，用 COM 技术是不可能实现这点的。COM 也提供了语言独立性，允许两个二进位的组件用两种不同的语言开发，比如 Visual Basic 和 C++，然后可以相互调用，但是 COM 的交互性仅在运行时存在。在开发时，.NET 可以在一种语言开发中得到用另一种语言开发的组件。.NET 可以做到这一点是因为 CLR 是基于严格类型的系统。为了符合 .NET 规范，语言中的所有结构，比如 class、struct 和原始类型，都必须编译成与 CLR 一致的类型。这种语言的交互性是以牺牲现有的语言和编辑器为代价的。现有的编辑器生成 CLR 不认识的代码，这些代码不遵从 CLR 类型系统，不能被 CLR 所管理。为了解决这个问题，Visual Studio .NET 包含了 4 种遵从 CLR 的语言：C#、Visual Basic .NET、JScript .NET 和受管制的 C++。第三方编辑器的出售者也将针对 CLR 开发出超过 20 种语言的编辑器，从 COBOL 到 Eiffel。

所有的 .NET 编程语言使用相同的基类、开发环境、CLR 设计限制和 CLR 类型。在 .NET 中编译代码分为两个过程。首先,高级语言被编译成一般的语言,即微软中间语言(IL),它和机器码有点相似。当运行时,第一次调用中间语言代码,.NET CLR 编译中间语言成为本地语言,然后作为本地代码执行它。一旦编译后,本地代码就被使用直到程序终结。另一种选择是,可以在安装时把它编译成本地语言。IL 是所有的 .NET 语言的共同部分,所以理论上,两种不同的语言等价构造体在编译成 IL 时应该是相同的。这样,所有的 .NET 语言在实现和开发难易度上是相同的。语言之间的不同在于美学和个人爱好。例如,为了使 C++ 顺从 CLR,你必须使用大量的编译指令和扩展,这叫 C++ 的管制扩展,其结果是很少可读的代码。与此类似,Visual Basic .NET 和以前的 Visual Studio 6.0 也是仅有一点相似,这就要求你忘记 Visual Basic 6 的一些习惯。只有 C#,它是新的 .NET 语言,没有遗留问题。C# 源自 C++,它组合了 C++ 功能的强大性和 Visual Basic 6 的易用性,提供了可读的、遵从 CLR 的类似 C++ 的语言。实际上,C# 更像标准的 C++ 语言。C# 也是纯粹的面向对象的语言,可以用来写可理解的和可重复使用的面向对象的代码。本书将在例题中使用 C#,当然你也可以用 Visual Basic .NET 和 C++ 来做同样的事情。.NET 语言的其他特征包括基于异常的公共错误处理,把每个实体看作一个对象,实体包括原始类型,其结果得到一个完整的编程模式。CLR 有丰富的、预定义的异常处理类,既可以照现在的样子使用它们,也可以为了特定的需求继承并扩展它们。一种语言抛出的异常可以在另一种语言中捕捉并处理。

.NET 使得开发二进位的组件更加容易了。不需要一个像 ATL 一样的框架,简单地声明一个类,它就可以变成一个组件。

面向组件开发的最重要的原则是分离接口和实现。COM 强制把接口定义从实现它们的类中分离出来。.NET 并不强制类的公共方法成为接口的一部分,但是强制同一接口的不同实现是多态的。接口定义是程序的一部分。不需要 IDL 文件,C# 保留字“接口”允许定义一个完全虚拟的、没有实现的、不能被客户端实例化的类型,与 C++ 中纯粹的虚拟或者抽象类相似。接口方法不必返回 HRESULT 或者其他错误处理类型。在出错的情况下,实现方法应该抛出一个异常。

1.1.1 .NET 发展历史

.NET 的第一版(1.0 版)于 2002 年发布,赢得了人们的喝彩。.NET 2.0 于 2005 年发布,是该架构的一个主要版本。2.0 版本增加了很多新特性,如 C# 和运行库中对泛型的支持、新类和接口。.NET 3.0 以 2.0 运行库为基础,引入了创建 UI 的新方式(WPF 和 XAML,基于矢量的图形)和一个新的通信技术(WCF)。.NET 引入 LINQ,这是可用于所有数据源的查询语法。.NET Framework 4 是该产品的另一个重要的版本,也引入了运行库的一个新版本,提供了动态语言集成和大量用于并行编程的新库。

.NET Framework 4.5 是一个针对 .NET Framework 4 的高度兼容的更新。通过将 .NET Framework 4.5 与 C#, Visual Basic 或 F# 编程语言结合使用,可以编写 Windows 应用程序。.NET Framework 4.5 包括针对 C#, Visual Basic 和 F# 的重大语

言和框架改进(以便更轻松地编写异步代码)、同步代码中的控制流混合、可响应 UI 和 Web 应用程序可扩展性。.NET Framework 4.5 添加了针对其他功能区域(如 ASP.NET, Managed Extensibility Framework, Windows Communication Foundation, Windows Workflow Foundation 和 Windows Identity Foundation)的大量改进。.NET Framework 4.5 提供了更高的性能、可靠性和安全性。图 1-1 展示了 .NET 的发展史。

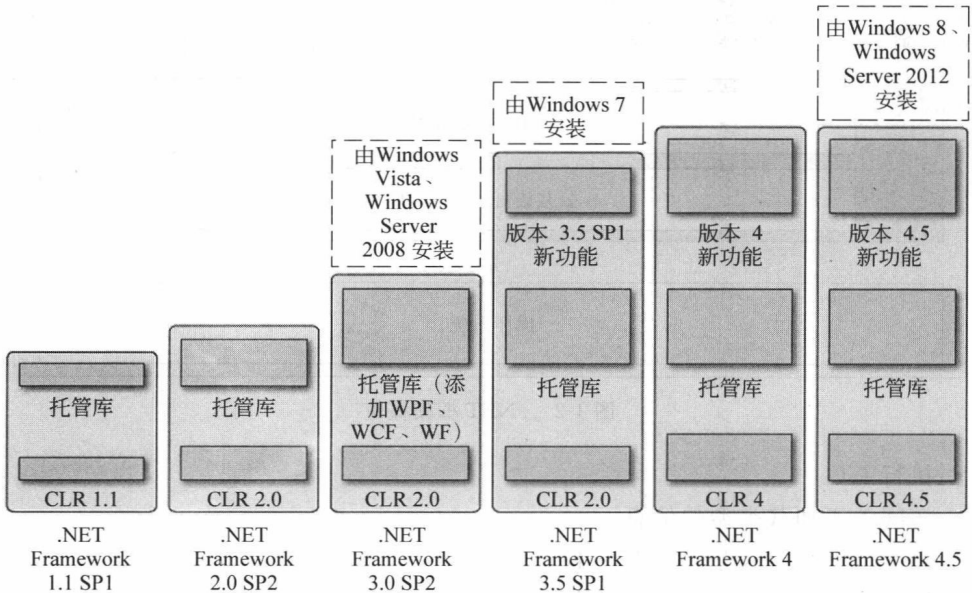


图 1-1 .NET 发展史

1.1.2 .NET 框架与公共语言运行库

可以将 .NET Framework 看作一个中间层,它介于操作系统和用户的应用程序之间。通过 .NET,用户可以忽略操作系统的差异,即平台无关性(目前只停留在理论阶段,.NET 的完整实现只限于 Windows 平台),这得益于 .NET 将代码编译为中间语言后再去执行。

.NET 框架整体结构如图 1-2 所示,中间的深色区域即 .NET 框架。

.NET 框架的底层是公共语言运行库(CLR),负责将源码编译为微软中间语言(IL)。

CLR 的上一层是 .NET Framework 类库,它封装了可加快和优化开发过程并提供对系统功能的访问的类、接口和值类型。为了便于语言之间进行交互操作,大多数 .NET Framework 类型都符合公共语言规范(CLS),因而可在编译器符合(CLS)的任何编程语言中使用。.NET Framework 类型是生成 .NET 应用程序、组件和控件的基础。.NET Framework 包括的类型可执行下列功能:

- 表示基础数据类型和异常。
- 封装数据结构。

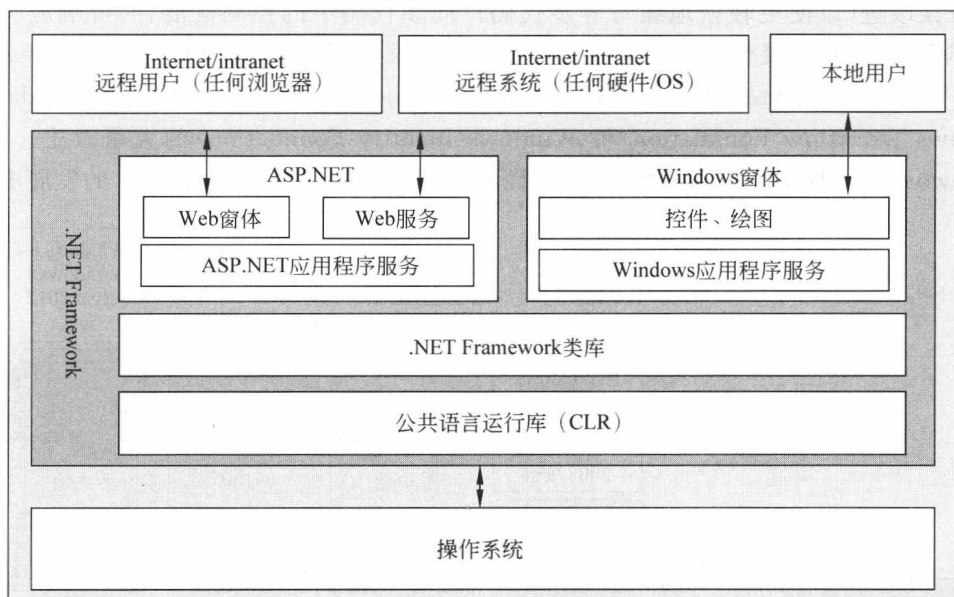


图 1-2 .NET 框架概览

- 执行 I/O。
- 访问关于加载类型的信息。
- 调用 .NET Framework 安全检查。
- 提供数据访问、多客户端 GUI 和服务端控制的客户端 GUI。

.NET Framework 提供了一组丰富的接口以及抽象类和具体(非抽象)类。可以按原样使用这些具体的类,或者在多数情况下从这些类中派生类。若要使用接口的功能,既可以创建实现接口的类,也可以从某个实现接口的 .NET Framework 类中派生类。

最上层是 ASP.NET 和 Windows 窗体,这是用户应用程序的两大类别。ASP.NET 能使用 Web 窗体设计开发动态网页,提供服务。Windows 窗体主要是桌面应用程序,可以使用控件设计出高效的用户界面。

公共语言运行库(CLR)是 .NET Framework 的核心之一,它能将源代码(C# 或其他语言)编译为平台专用的代码,这个过程需要使用 IL 做中介。IL 是一种低级语言,能够快速转换为本地机器码,这样设计的目的是实现平台无关性以及提高性能和语言的互操作性。CLR 的核心功能包括内存管理、程序集加载、安全性、异常处理和线程同步,可供面向 CLR 的所有语言使用,并保证应用和底层操作系统之间必要的分离。

1.2 Visual Studio 2013 IDE 简介

1.2.1 Visual Studio 开发环境

Visual Studio 是微软公司推出的一套强大的开发工具,它集编写、编译、调试、运行

于一体,能开发 ASP.NET、WPF、WCF 以及 Windows 窗体应用程序,编程语言涵盖 C#、Visual Basic、C++ 以及 F#,它还集成了设计视图编辑器用于设计各种窗体样式。Visual Studio 窗口如图 1-3 所示。

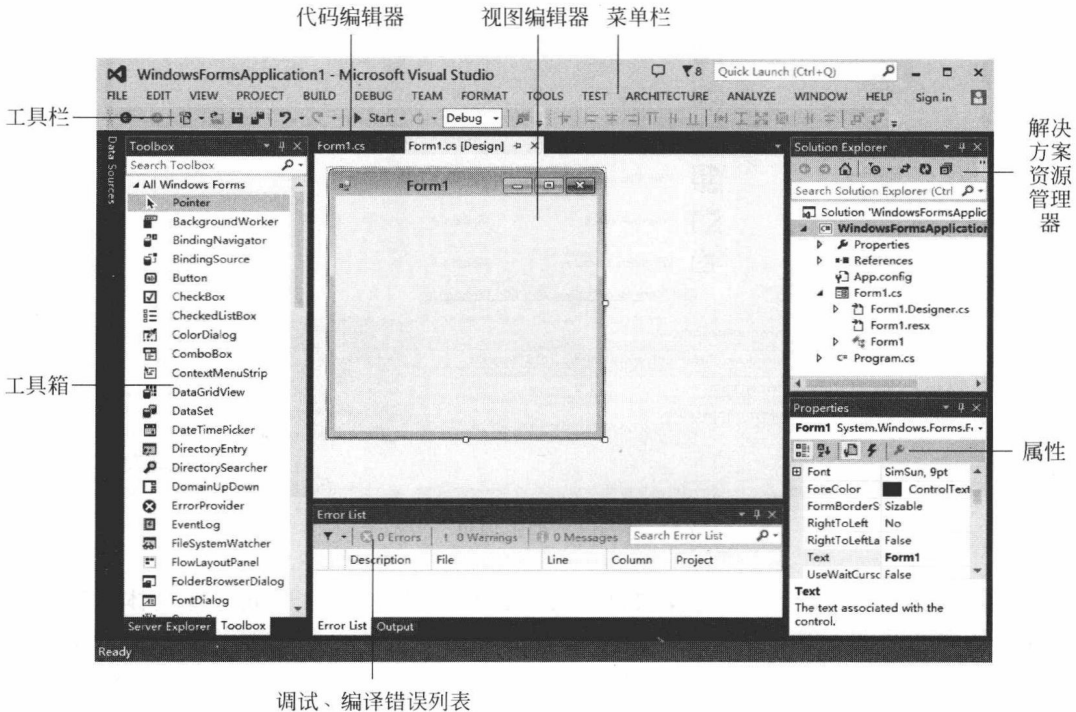


图 1-3 Visual Studio 窗口

图 1-3 中指示了 Visual Studio 中各个窗口的名称,下面介绍各个窗口的作用:

- 菜单栏: 包含了项目创建、文件操作以及调试运行等功能项。
- 工具栏: 各类工具的集合,可选择显示某些工具栏。
- 视图编辑器: 用于设计窗体,保证“所见即所得”。
- 代码编辑器: 编写代码,并提供代码颜色标记。
- 解决方案资源管理器: 项目的窗体、代码、引用等都由其管理,显示为树形列表。
- 属性: 方便设置窗体的各类属性。
- 工具箱: 控件的集合,用于拖放控件。
- 调试、编译错误列表: 调试的中间结果以及编译的错误都将在此详细列出。

1.2.2 创建项目

Visual Studio 的项目种类繁多,可以根据自己的需要创建合适的项目。在菜单栏中依次选择 FILE→NEW→Project,或使用快捷键 Ctrl+Shift+N 便能打开新建项目窗口,如图 1-4 所示。