

天津滨海迅腾科技集团有限公司

《企业级卓越人才培养解决方案》指定用书



Windows

程序设计基础.net

基于.NET平台

王新强 主编

南开大学出版社

Windows 程序设计基础

——基于.NET 平台

王新强 主编

南开大学出版社
天 津

图书在版编目(CIP)数据

Windows 程序设计基础：基于.NET 平台 / 王新强主
编. —天津：南开大学出版社，2016.7

ISBN 978-7-310-05124-3

I. ①W… II. ①王… III. ①Windows 操作系统—程序
设计 IV. ①TP316.7

中国版本图书馆 CIP 数据核字(2016)第 125234 号

版权所有 侵权必究

南开大学出版社出版发行

出版人：孙克强

地址：天津市南开区卫津路 94 号 邮政编码：300071

营销部电话：(022)23508339 23500755

营销部传真：(022)23508542 邮购部电话：(022)23502200

*

北京楠海印刷厂印刷

全国各地新华书店经销

2016 年 7 月第 1 版 2016 年 7 月第 1 次印刷

260×185 毫米 16 开本 15 印张 376 千字

定价：49.00 元

如遇图书印装质量问题，请与本社营销部联系调换，电话：(022)23507125

企业级卓越互联网应用型人才培养解决方案

一、企业概况

天津滨海迅腾科技集团是以 IT 产业为主导的高科技企业集团,总部设立在北方经济中心——天津,子公司和分支机构遍布全国近 20 个省市,集团旗下的迅腾国际、迅腾科技、迅腾网络、迅腾生物、迅腾日化分属于 IT 教育、软件研发、互联网服务、生物科技、快速消费品五大产业模块,形成了以科技为源动力的现代科技服务产业链。集团先后荣获“全国双爱双评先进单位”“天津市五一劳动奖状”“天津市政府授予 AAA 级和谐企业”“天津市文明单位”“高新技术企业”“骨干科技企业”等近百项殊荣。集团多年中自主研发天津市科技成果 2 项,自主研发计算机类专业教材 36 种,具备自主知识产权的开发项目包括“进销存管理系统”“中小企业信息化平台”“公检法信息化平台”“CRM 营销管理系统”“OA 办公系统”“酒店管理系统”等数十余项。2008 年起成为国家工业和信息化部人才交流中心“全国信息化工程师”项目联合认证单位。

二、项目概况

迅腾科技集团“企业级卓越互联网应用型人才培养解决方案”是针对我国高等职业教育量身定制的应用型人才培养解决方案,由迅腾科技集团历经十余年研究与实践研发的科研成果,该解决方案集三十余本互联网应用技术教材、人才培养方案、课程标准、企业项目案例、考评体系、认证体系、教学管理体系、就业管理体系等于一体。采用校企融合、产学研融合、师资融合的模式在高校内建立校企共建互联网学院、软件学院、工程师培养基地的方式,开展“卓越工程师培养计划”,开设互联网应用技术领域系列“卓越工程师班”,“将企业人才需求标准引进课堂,将企业工作流程引进课堂,将企业研发项目引进课堂,将企业考评体系引进课堂,将企业一线工程师请进课堂,将企业管理体系引进课堂,将企业岗位化训练项目引进课堂,将准职业人培养体系引进课堂”,实现互联网应用型卓越人才培养目标,旨在提升高校人才培养水平,充分发挥校企双方特长,致力于互联网行业应用型人才培养。迅腾科技集团“企业级卓越互联网应用型人才培养解决方案”已在全国近二十所高校开始实施,目前已形成企业、高校、学生三方共赢格局。未来五年将努力实现在 100 所高校实施“每年培养 5~10 万互联网应用技术型人才”发展目标,为互联网行业发展做好人才支撑。

前 言

首先感谢您选择了企业级卓越互联网应用型人才培养解决方案，选择了本教材。本教材是企业级卓越互联网应用型人才培养解决方案的载体之一，面向行业应用与产业发展需求，系统传授软件开发全过程的理论和技能，并注重 IT 管理知识的传授和案例教学。本书主要讲解数据库基础知识以及使用 C#语言编写应用程序。这本书是提高读者企业编程能力的一条捷径。

本书的特点是由浅入深，从最基本的数据库开始讲起，逐步深入到面向对象、Windows 程序设计、数据库高级等编程方法。在介绍语法时，本书并没有像一些语法书那样教条死板地讲定义，而是利用示例代码生动地让读者在实践中体会一个个知识点。

本书第 1 章介绍了 Windows 开发的基础知识，让初学者了解事件编程，开始编写第一个窗体应用程序。第 2 章介绍了 C#语言面向对象的三大特性，以及 C#语言的基本数据类型、运算符、表达式和控制流程，让读者对 C#语言有更深一步的认识。第 3 章介绍了 WinForm 窗体控件，主要包括对窗体、控件和属性等概念的理解，如何实现窗体间的跳转，以及 MessageBox 对象的应用。第 4 章主要介绍了常用控件，包括 Label 和 TextBox 的使用，CheckBox、RadioButton、ComboBox 和 ListBox 的使用，NumericUpDown 和 pictureBox 的使用等。第 5 章主要介绍了 Panel、GroupBox 容器控件和 MenuStrip 等菜单控件。第 6 章介绍了 Connection、Command、DataAdapter、DataReader、DataSet 对象，用 DataGridView 显示数据，用 ComboBox 显示数据，使用 DataReader 读取数据并显示。第 7 章介绍了如何向数据库中增加、删除和修改数据，以及 C#的异常处理方法。

由于作者水平有限，加之计算机技术博大精深，书中难免有不当和疏漏之处，在内容选取和叙述上也难免有不当之处。欢迎广大读者对本书提出批评和建议，我们的邮箱是：
develop_etc@126.com。

天津滨海迅腾科技集团有限公司课程研发部
2016 年 5 月

目 录

理论部分

第 1 章 Windows 开发基础知识	3
1.1 Windows 开发基本知识	3
1.2 Windows 窗体特性	5
1.3 Windows 窗体应用程序模型	6
1.4 集成开发环境 (VS.NET 2012)	7
1.5 第一个 Windows 窗体应用程序示例	12
1.6 认识 Windows 窗体应用程序文件夹结构	14
1.7 使用命名空间	18
1.8 小结	18
1.9 英语角	19
1.10 作业	19
1.11 思考题	19
第 2 章 C#语言	20
2.1 C#的特性	20
2.2 Console 类	23
2.3 数据类型	28
2.4 变量	30
2.5 表达式	32
2.6 流程控制	36
2.7 小结	42
2.8 英语角	42
2.9 作业	42
2.10 思考题	42
第 3 章 WinForm 基础	44
3.1 基于事件的编程	44
3.2 Windows 窗体控件	45

3.3	按钮控件.....	49
3.4	实现窗体间的跳转.....	51
3.5	MessageBox 对象的应用.....	51
3.6	小结.....	60
3.7	英语角.....	60
3.8	作业.....	60
3.9	思考题.....	61
第 4 章	Windows 窗体常用控件.....	62
4.1	Label (标签控件) 和 LinkLabel (超链接标签控件).....	63
4.2	TextBox (文本框控件).....	66
4.3	CheckBox (复选框控件).....	73
4.4	RadioButton (单选框控件).....	74
4.5	ComboBox (组合框控件).....	76
4.6	ListBox (项目列表控件).....	78
4.7	NumericUpDown (数字显示框控件).....	80
4.8	PictureBox (图片框控件).....	82
4.9	小结.....	86
4.10	英语角.....	86
4.11	作业.....	87
4.12	思考题.....	87
第 5 章	C#Windows 容器控件和菜单控件.....	88
5.1	GroupBox.....	88
5.2	Panel.....	91
5.3	TabControl.....	93
5.4	TabPage.....	95
5.5	StatusStrip.....	96
5.6	MenuStrip.....	98
5.7	上下文菜单.....	101
5.8	ToolStrip.....	102
5.9	小结.....	105
5.10	英语角.....	106
5.11	作业.....	106
5.12	思考题.....	107
第 6 章	ADO.NET 简单应用 (1).....	108
6.1	ADO.NET 概述.....	108
6.2	Connection 对象.....	110

6.3	Command 对象	113
6.4	DataGridView 控件	115
6.5	DataSet、DataAdapter 对象	116
6.6	DataReader 对象	125
6.7	趣谈 ADO.NET 对象模型	129
6.8	小结	131
6.9	英语角	131
6.10	作业	131
6.11	思考题	132
第 7 章	ADO.NET 简单应用 (2)	133
7.1	异常处理	133
7.2	对数据库数据的添加操作	138
7.3	对数据库数据的删除操作	143
7.4	对数据库数据的修改操作	147
7.5	数据库增删改操作小结	153
7.6	小结	153
7.7	英语角	154
7.8	作业	155
7.9	思考题	155
上机部分		
第 1 章	Windows 开发基础知识	159
1.1	指导 (60 分钟)	159
1.2	练习 (60 分钟)	163
1.3	作业	163
第 2 章	C#语言	164
2.1	指导 (60 分钟)	164
2.2	练习 (60 分钟)	173
2.3	作业	174
第 3 章	WinForm 基础	175
3.1	指导 (60 分钟)	175
3.2	练习 (60 分钟)	181
3.3	作业	181
第 4 章	Windows 窗体常用控件	182
4.1	指导 (60 分钟)	182

4.2	练习 (60 分钟)	189
4.3	作业	192
第 5 章	C#Windows 容器和菜单控件	193
5.1	指导 (60 分钟)	193
5.2	练习 (60 分钟)	204
5.3	作业	204
第 6 章	ADO.NET 的简单应用 (1)	205
6.1	指导 (60 分钟)	205
6.2	练习 (60 分钟)	213
6.3	作业	219
第 7 章	ADO.NET 简单应用 (2)	220
7.1	指导 (60 分钟)	220
7.2	练习 (60 分钟)	226
7.3	作业	229

理论部分

第 1 章 Windows 开发基础知识

学习目标

- ◇ 了解 Windows 应用程序模型。
- ◇ 理解 Windows 窗体 (Form)。
- ◇ 掌握集成开发环境 (VS.NET 2012)。

1.1 Windows 开发基本知识

在前面介绍的一些编程中，我们只有在控制台命令下进行输入一些简单的数据，然后应用程序把结果在控制台上输出，这个和我们现在使用的应用程序有很大的区别，现在的应用程序只要在屏幕上点击一些按钮，在一些框中输入内容，就好像 MSN、Office 等应用程序，我们把这些有图形用户界面 (GUI) 的应用程序称为 Windows 应用程序。

我们要进行 Windows 应用程序编程，首先要了解一些 Windows 应用程序的基本知识。下面我们就来介绍一些 Windows 应用程序的基本知识。

Windows 应用程序一般拥有图形用户界面，这种界面也称作“视觉化界面”或“图形窗口环境”。图形用户界面概念的出现改变了原有的应用程序的开发和运行方式，它把个人电脑上的各种技术都融合在了一起。

所有图形用户界面都在点矩阵对应的视频显示器上处理图形。图形提供了使用屏幕的最佳方式、传递信息的视觉化的环境，以及能够体现所见即所得 (What you see is what you get) 的图形视频显示，还可以为书面文件准备好格式化文本输出内容。

在 DOS 时代，视频显示器仅用于回应用户通过键盘输入的文本。现在，在 Windows 图形用户界面中，视频显示器自身成为用户输入的一个来源。视频显示器以图形和输入设备 (例如按钮和滚动条) 的形式显示多种图形对象。用户可以使用键盘 (或者更直接地使用鼠标等指向设备) 直接在屏幕上操纵应用程序，拖动应用程序的一些图形、按下鼠标按钮以及滚动滚动条。这个就好比我们在 DOS 时代的游戏一样，在 DOS 时代的游戏大多都是通过键盘来控制人物的行走，然后，游戏在显示器上通过文字描述来通知玩家发生什么事情了，用户再看屏幕上的文字来进行操作。但目前在 Windows 下的游戏就不是这样的了，我们通过图形知道我们在地图的什么地方，然后通过鼠标控制人物行走，碰到什么情况游戏都是通过图形来通知游戏玩家的。

这样，用户与程序的交流变得更为亲密。这不再是一种从键盘到程序，再到视频显示器的单向信息流动，用户已经能够与显示器上的对象直接交互作用了。

Windows 应用程序的用户不再需要花费长时间学习如何使用电脑或掌握新程序了。图形用户界面使 Windows 应用程序的操作变得非常的简单，因为所有应用程序都有基本相同的外观和感觉。程序占据一个窗口——屏幕上的一块矩形区域。每个窗口由一个标题列标识。大多数程序功能由程序的菜单开始。用户可使用滚动条观察那些无法在一个屏幕中装下的信息。某些菜单项目触发对话框，用户可在其中输入额外的信息。几乎在每个大的 Windows 程序中都有一个用于打开文件的特殊对话框。该对话框在所有这些 Windows 应用程序中看起来都一样（或接近相同），而且几乎总是从同一菜单选项中启动。

一旦我们了解一个 Windows 应用程序的使用方法，我们就非常容易学习其他的 Windows 应用程序。菜单和对话框允许用户实验一个新程序并探究它的功能。大多数 Windows 应用程序同时具有键盘界面和鼠标界面。虽然 Windows 应用程序的大多数功能可通过键盘控制，但使用鼠标要容易得多。

从程序开发的角度看，一致的用户界面来自于 Windows 建构菜单和对话框的内置程序。所有菜单都有同样的键盘和鼠标界面，因为这项工作是由 Windows 处理，而不是由应用程序处理。

那么接下来我们看看在 Windows 编程经常可以看到哪些术语。我们以一个普通的应用程序为例。下面是资源管理器系统界面，那么我们看看这个窗体由哪些 Windows 元素组成。图 1-1 为资源管理器的效果图。

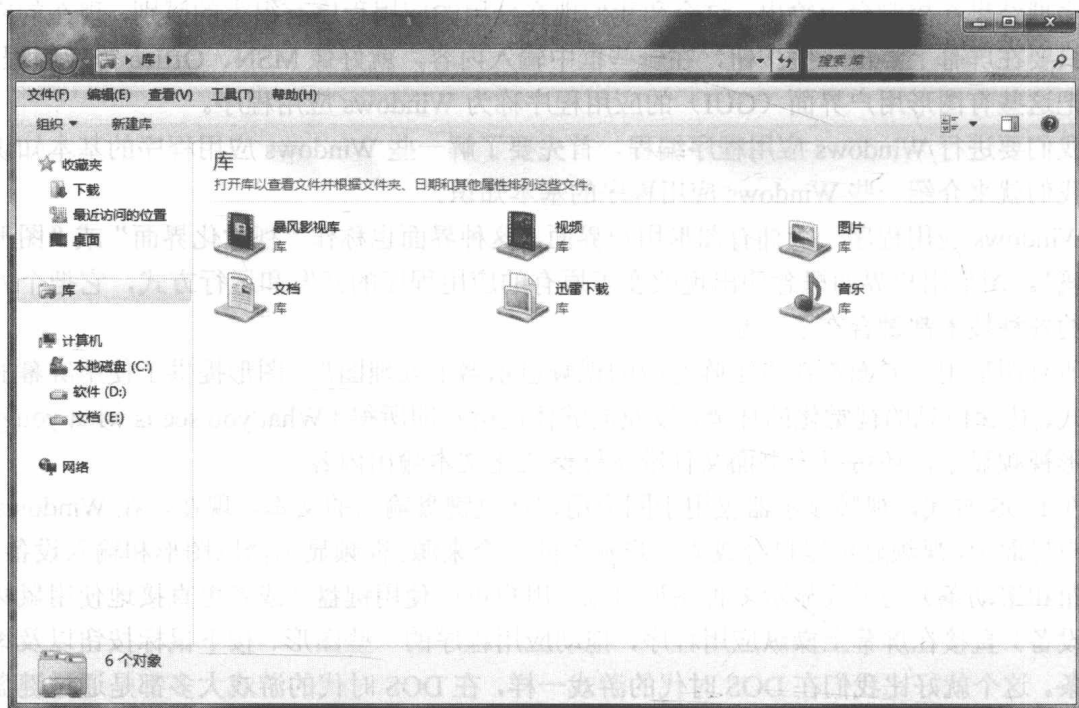


图 1-1 Windows 资源管理器

我们可以看到一个 Windows 应用程序首先要有一个窗口，内容都在窗体中，然后是标题，告知这个窗体的作用是什么，然后是菜单，就是这个窗体的一些操作，还有就是标签用

来显示内容的，文本输入框是让用户输入内容的。所以，在 Windows 应用程序中看得最多的是窗口（Windows）。

1.2 Windows 窗体特性

我们采用微软的 .NET Framework 平台开发 Windows 应用程序。 .NET Framework 是微软用于支持生成和运行下一代应用程序 XML Web Services 的内部 Windows 组件。 .NET Framework 旨在实现下列目标：提供一个一致的编程环境（而无论代码是在本地存储和执行，还是在本地执行，还是在 Internet 上分布，或者是在远程执行的）；提供一个将软件部署和版本控制冲突最小化的代码执行环境；提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的执行环境；提供一个可消除脚本环境或解释环境的性能问题的代码执行环境；使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致；按照工业标准生成所有通信，以确保基于 .NET Framework 的代码可与任何其他代码集成。公共语言运行库是 .NET Framework 开发平台的一个核心，同时也是 .NET Framework 开发平台的基础。

Windows 窗体就是利用公共语言运行库生成 Windows 客户端应用程序的框架。可用公共语言运行库支持的任何语言编写 Windows 窗体应用程序。Windows 窗体一般有以下好处：

简单而且功能强大：Windows 窗体是用于开发 Windows 应用程序的编程模型，它融合了像搭积木方式创建房子一样的简单性和公共语言运行库的强大功能和灵活性。

控件的结构：Windows 窗体提供用于控件和控件容器的结构，该结构基于控件和容器类的具体实现。这显著减少了控件和容器间的交互问题。在开发的时候，只要把一些控件，如按钮、菜单、文本输入框等拖入窗体中，不需要任何编码，就能实现在窗体上显示这些控件。也就是说，这些控件是预先提供好的，我们在编程的时候，只要把这些控件组织在一起。

安全性：Windows 窗体充分利用公共语言运行库的安全功能。这意味着 Windows 窗体可用于实现所有内容，从在浏览器中运行的不受信任的控件到安装在用户硬盘的完全受信任的应用程序，范围十分广泛。

丰富的图形：Windows 窗体是图形设备接口（GDI）的一个实现，图形设备接口支持 Alpha 混合效果、纹理画笔、高级转换和多格式文本支持等。这样，我们可以在窗体上画各种系统没有提供的图形。

灵活的控件：Windows 窗体提供一组丰富的控件，其中包含 Windows 提供的所有控件。这些控件还提供新功能，如用于按钮、单选按钮和复选按钮的“平面”样式。有关控件的完整列表，请参考 1.3.2。

数据识别功能：Windows 窗体可以通过 ADO.NET 数据模式直接方便地访问数据库。

设计时支持：Windows 窗体为控件用户和控件实施者提供全面的设计时支持，提供直接在设计时修改控件属性来改变控件的显示效果。对象是指具有数据、行为和标识的编程结构。例如，我们可以把一面墙看作为一个具体对象，那么墙的颜色就是这面墙的一个属性（即对象属性）。对象是指具有数据和行为的一个实体。这里的数据一般称为对象的属性，行为称为对象的方法。

1.3 Windows 窗体应用程序模型

Windows 窗体的应用程序编程模型主要由窗体、控件和事件组成。在这里我们将讲述 Windows 窗体应用程序模型的以下方面：窗体、控件、事件、属性和生存期。

1.3.1 窗体

窗体是 Windows 本身及 Windows 环境下的应用程序的基本界面单位。在 .NET Framework 平台下，Windows 窗体是一个 Form 的对象。在 Windows 窗体中，Form 对象是应用程序中显示的任何窗体的表示形式。

1.3.2 控件

我们在应用程序中经常看到一些按钮、文本输入框、下拉选择框、单选按钮和复选框等等，这些在 .NET Framework 开发平台上统称为控件。我们在开发应用程序时只需从工具箱中把它们拖入到窗体上就可以了。如果我们要让用户输入的话，我们可以把文本输入框拖入到窗体中；如果我们要让用户从多个选项选择一个，我们可以把下拉选择框拖入到窗体中即可。

我们可以通过设置这些控件的属性来达到不同的显示效果。

1.3.3 属性

控件的属性就是控件的特性，也称为对象属性，是用来表示控件对象特征的数据，如按钮控件有背景色，前景色，字体大小等属性。我们在开发应用程序时需要在属性窗口设置控件的属性。

控件的属性决定了控件的外观和操作。可以用两种方法来设置。一种是通过窗口设置，另外一种是在程序事件过程中通过程序代码设置。

1.3.4 事件

Windows 窗体编程模型基于事件，即 Windows 程序是由事件驱动的。换句话说，事件驱动的程序是由事件的发生来控制。这种设计思想是一种非顺序方式处理事件，从而避开了顺序的过程处理方法。如当用户在应用程序上用鼠标点击按钮时，就发生了按钮点击的事件，此时我们只要在按钮点击事件处理程序中加入相应的处理就实现了我们想要的功能。

事件的含义很广泛，最常见的是鼠标事件，包括鼠标移动、鼠标左右键的单击双击、鼠标位于窗口的位置等。应用程序启动后，程序等候事件的发生，当某事件发生时就进行相应的处理、执行特定的功能。事件也可以由用户自己来定义。我们在本教材中只讲述一些简单事件的应用。

1.3.5 生存期

“生存期”是指对象可供使用的时间周期。比如，当我们要使用一个窗体的时候，首先

要创建该窗体，这个时候，该窗体开始了它的生命周期，当我们不再使用该窗体，关闭该窗体的时候，该窗体就被释放，窗体的生命周期就结束了。我们只有在窗体的生命周期中才能使用该窗体，否则会出现错误。控件也是如此。

1.4 集成开发环境（VS.NET 2012）

Visual Studio 2012 是一套完整的开发集，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C#和 Visual J#全都使用相同的集成开发环境（IDE），利用此集成开发环境可以共享工具且有助于创建混合语言解决方案。在本书中，我们仅使用 C#编程语言。首先，我们打开 Visual Studio 2012 集成开发环境，看看其界面和为开发人员提供了哪些功能，效果如图 1-2 所示。

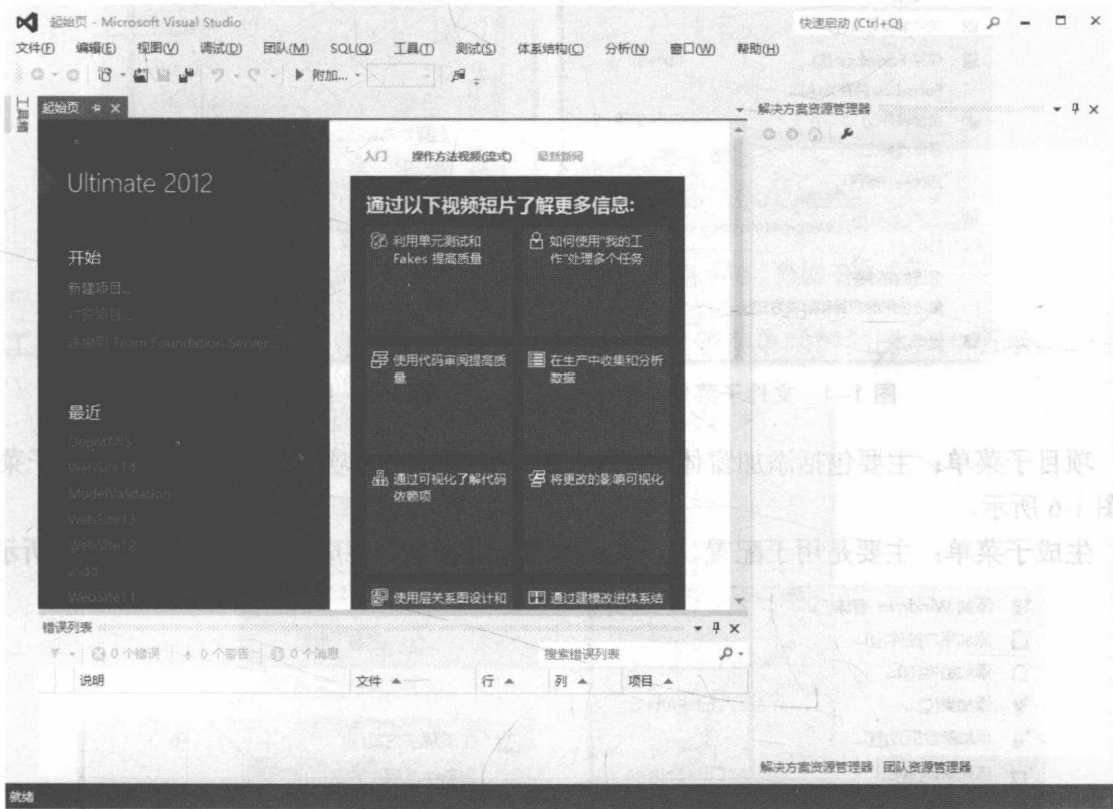


图 1-2 Visual Studio 2012 开发环境

Visual Studio 产品家族共享一个集成开发环境（IDE）。IDE 由若干元素组成：菜单工具栏、标准工具栏，停靠或自动隐藏在左侧、右侧、底部以及编辑器空间的各种工具窗口。在任何给定时间，可用的工具窗口、菜单和工具栏取决于所处理的项目或文件类型。

1.4.1 菜单栏

主菜单主要包括文件、编辑、视图、调试、团队、SQL、工具、测试、体系结构、分析、窗口和帮助等子菜单。图 1-3 为菜单栏的效果图。



图 1-3 菜单栏

下面我们就着重介绍几个主要的子菜单和一些常用的工具：

文件子菜单：主要包括新建、打开、保存和关闭解决方案、导出模版、源代码管理以及最近文件、退出等功能。其中添加功能主要是向方案或项目中添加新的项目或现有项目。文件子菜单的效果如图 1-4 所示。

编辑子菜单：主要是文本内容的剪切、撤销、查找和替换等功能。编辑子菜单功能如图 1-5 所示。

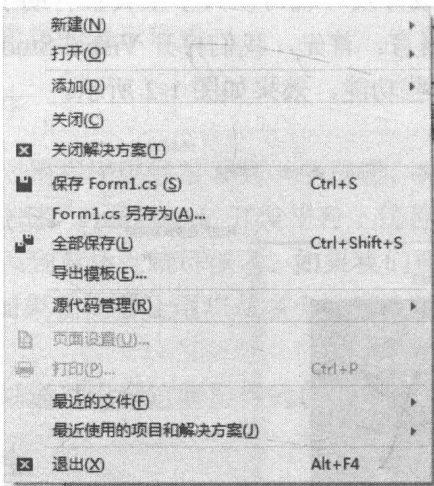


图 1-4 文件子菜单

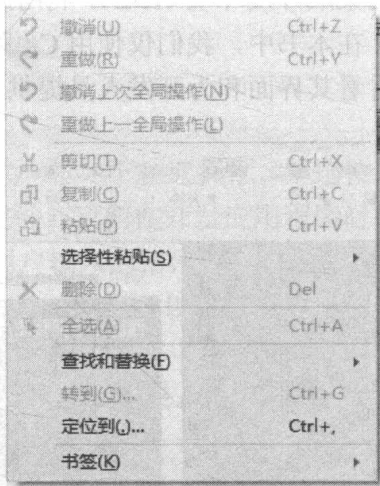


图 1-5 编辑子菜单

项目子菜单：主要包括添加窗体、控件、类、引用和设置项目属性等功能。项目子菜单如图 1-6 所示。

生成子菜单：主要是用于配置、生成或清理应用程序。生成子菜单功能如图 1-7 所示。

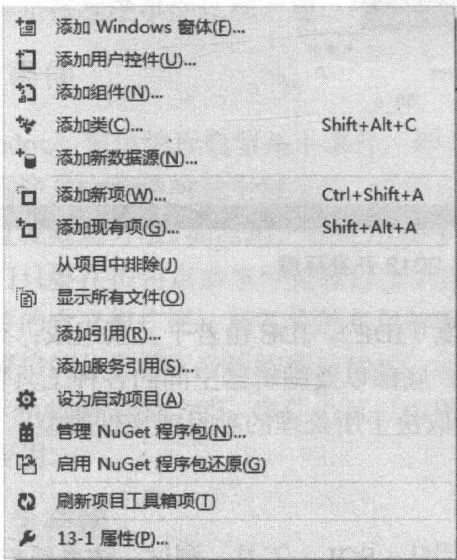


图 1-6 项目子菜单

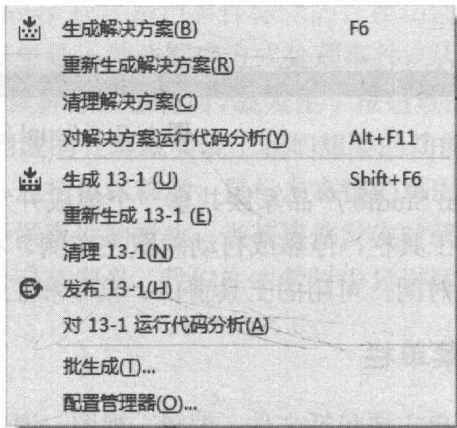


图 1-7 生成子菜单