



21世纪高等学校电子信息类专业规划教材

C#

实用开发教程

李纯莲 刘玉宝 主 编
祝海英 孙 宇 副主编



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

21 世纪高等学校电子信息类规划教材

C# 实用开发教程

李纯莲 刘玉宝 主 编
祝海英 孙 宇 副主编

清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

本书是面向高等院校培养应用型本科人才的发展目标而编写的。介绍了.NET 基本知识及其应用程序的开发技术。全书共分为 12 章, 主要内容包括: C# 编程概述、C# 语言基础、Windows 基本控件的使用、数据库系统设计基础、系统建模技术、使用 C# 和 ADO.NET 操作数据库、使用 C# 开发 Windows 数据库应用程序、Web 应用技术、Web 程序设计基础、网上书店系统的设计、Windows Mobile 智能设备程序设计基础、移动 Web 程序开发。本书本着循序渐进的原则, 内容与实例相结合, 使理论与实践同步, 逐步引领读者掌握.NET 应用程序开发技术。

本书内容丰富, 结构合理, 实用性强。力求理论叙述严谨, 应用能力培养目标明确, 使读者学完本门课程后, 能够具备.NET 应用程序的独立开发能力。

本书可作为各大学、专科院校计算机及信息专业的本、专科学生的教科书, 也可作为高职高专学校在校生的教科书, 同时也适合程序开发人员参考之用。

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 13501256678 13801310933

图书在版编目 (CIP) 数据

C#实用开发教程/李纯莲, 刘玉宝主编. —北京: 清华大学出版社, 北京交通大学出版社, 2008.10

(21 世纪高等学校电子信息类专业规划教材)

ISBN 978-7-81123-418-3

I. C… II. ①李… ②刘… III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 157987 号

责任编辑: 郭东青

出版发行: 清华大学出版社 邮编: 100084 电话: 010-62776969

北京交通大学出版社 邮编: 100044 电话: 010-51686414

印刷者: 北京交大印刷厂

经 销: 全国新华书店

开 本: 185×260 印张: 25.25 字数: 630 千字

版 次: 2008 年 11 月第 1 版 2008 年 11 月第 1 次印刷

书 号: ISBN 978-7-81123-418-3/TP·443

印 数: 1~4 000 册 定价: 38.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

前 言

.NET 技术是微软公司为了适应未来软件的发展与应用而推出来的开发平台。其技术全面而先进,必将对今后的软件开发与应用产生巨大的影响,因此及时了解与掌握这门新技术,跟上新技术发展的步伐是非常必要的。

本书是组织了多年讲授数据库原理与数据库应用技术的一线教师,结合自己的教学经验和教学体会,整理和丰富了教学讲义而编写的。本书的特点在于能够把 .NET 平台中的 C#、ASP.NET 和 SQL Server 2000 有机结合起来,理论叙述严谨,培养目标明确,使读者在学习过程中做到理论与实践相结合,并且掌握使用最新的软件开发工具开发数据库应用程序的技能;而且,叙述力求简单明了、深入浅出,在数据库技术应用相关章节尽量避免冗长的理论叙述,而侧重于技术的应用与程序开发能力的培养,使读者阅读后能具备基本的 .NET 应用程序的开发能力,快速适应实际工作。

全书共分为 12 章,理论教学参考学时为 56 学时。使用者可以根据实际情况进行适当的取舍。

第 1 章 C# 编程概述,简单介绍了 .NET 发展历史、技术体系、开发环境、C# 语言的特点以及 MSDN 联机帮助的使用方法,并对基于 Visual Studio 2005 创建 C# 工程的具体步骤作了较为详尽的描述。

第 2 章 C# 语言基础,主要介绍了 C# 简单程序的特征、标识符、关键字、常量和变量、运算符、控制语句及自定义类等基础知识。

第 3 章 Windows 基本控件的使用,主要介绍了 Microsoft Visual Studio 2005 基本控件的使用以及运用它们开发 Windows 基本应用程序的方法。

第 4 章数据库系统设计基础,主要介绍了数据库基本概念及原理,SQL Server 2000 的安装与使用,Transact-SQL 语言的基本应用。

第 5 章系统建模技术,主要介绍了 UML 建模的概念、构成及应用,并以 PowerDesigner 为例进行了简要的介绍。

第 6 章使用 C# 和 ADO.NET 操作数据库,主要介绍了 ADO.NET 的体系结构的组成及工作原理,数据提供程序的选择,SqlConnection 的使用,OleDbConnection 的使用,OracleConnection 的使用,数据的获取,DataReader 的使用以及 DataSet 和 DataAdapter 的使用等。

第 7 章使用 C# 开发 Windows 数据库应用程序,主要介绍了简单数据绑定,复杂数据绑定,数据源的类型,DataGrid 的使用以及 Windows 应用程序开发实例等。

第 8 章 Web 应用技术,主要介绍了 Web 应用模型、System.Web 命名空间及 ASP.NET、Web Service 的相关知识。

第 9 章 Web 程序设计基础,主要介绍了新建 Web 应用程序的方法、数据绑定技术原理及 Web 数据绑定控件的使用。

第 10 章网上书店系统的设计,按照面向对象的软件工程的观点详细介绍了网上书店

系统的分析、设计与实现的全过程。

第11章 Windows Mobile 智能设备程序设计基础，首先介绍了 .NET Compact Framework 精简版架构和 Windows Mobile 智能设备程序的基本知识，然后详细介绍了开发环境的搭建，最后介绍了 Windows Mobile 智能设备应用程序的创建及 SQL Mobile 数据访问方法。

第12章移动 Web 程序开发，主要介绍了移动 Web 网站的特点、站点的创建与测试、Web 移动控件的使用及 Web 移动窗体的设计。

全书在内容编排上，本着循序渐进的原则，结构合理，内容翔实，实例丰富，语言通俗易懂，实用性强，具有较高的参考价值和实用价值。建议读者在此基础上查阅相关的书籍，以达到更好的学习效果。

本书由李纯莲、刘玉宝担任主编，祝海英、孙宇担任副主编。基中第1、2、8章由孙宇编写，第3、4章由祝海英编写，第5章由徐志伟编写，第6、7章由刘玉宝编写，第9、10、11、12章由李纯莲编写，林营、高鹏、王任之、肖骞、辛梁艺、徐攀等在本书的编辑、排版、校对等方面做了大量的工作，最后由李纯莲统一定稿。

在本书编写的过程中得到了单位的领导和同仁的热情帮助和支持，在此表示衷心的感谢！

本书的编写参考了广大同行的著作和成果，在此对他们表示衷心的感谢！

计算机技术日新月异，软件开发技术的发展更是十分迅速，由于时间仓促，加之作者的水平有限，书中难免有疏漏和不足之处，恳请同行专家和广大读者批评指正。

编者

2008年9月

目 录

第 1 章 C# 编程概述	1
1.1 .NET 基础	1
1.2 C# 语言简介	9
1.3 用 Visual Studio 2005 创建 C# 项目	11
1.4 MSDN 联机帮助	17
习题	20
第 2 章 C# 语言基础	21
2.1 最简单的 C# 程序	21
2.2 C# 中的标识符	22
2.3 C# 中的关键字	22
2.4 数据类型	23
2.5 常量和变量	30
2.6 运算符	30
2.7 控制语句	31
2.8 自定义类	38
习题	42
第 3 章 Windows 基本控件的使用	43
3.1 窗体	44
3.2 文本类控件	57
3.3 按钮类控件	61
3.4 框架类控件	67
3.5 列表框、视图类控件	70
3.6 滚动类控件	82
3.7 工具栏和状态栏控件	85
3.8 定时器控件	91
3.9 菜单控件	94
习题	99
第 4 章 数据库系统设计基础	100
4.1 基本概念	100
4.2 数据库模型	102
4.3 SQL Server 2000 的使用	104
习题	141
第 5 章 系统建模技术	143

5.1 建模概念	143
5.2 UML 建模	143
5.3 PowerDesigner 建模	156
习题	161
第 6 章 使用 C# 和 ADO . NET 操作数据库	162
6.1 ADO . NET 简介	162
6.2 ADO . NET 的体系结构	163
6.3 ADO . NET 数据对象介绍	168
6.4 数据库的连接	170
6.5 数据的获取	173
6.6 DataReader 的使用	184
6.7 DataSet 和 DataAdapter 的使用	189
习题	194
第 7 章 使用 C# 开发 Windows 数据库应用程序	196
7.1 数据绑定	196
7.2 数据源的类型	199
7.3 使用 DataGrid 显示数据	201
7.4 应用实例	203
习题	219
第 8 章 Web 应用技术	220
8.1 Web 应用模型	220
8.2 ASP . NET 基础	223
8.3 System. Web 命名空间	228
8.4 ASP . NET	229
8.5 We Service	245
习题	258
第 9 章 Web 程序设计基础	259
9.1 新建 Web 应用程序	259
9.2 数据绑定技术	260
9.3 Web 数据绑定控件	261
习题	292
第 10 章 网上书店系统的设计	293
10.1 概述	293
10.2 需求分析	294
10.3 系统框架	296
10.4 系统配置	298
10.5 数据库分析	301
10.6 数据库创建	306

10.7	数据库连接.....	313
10.8	主要界面及相关代码分析.....	315
10.9	实例演示.....	341
	习题.....	345
第 11 章	Windows Mobile 智能设备程序设计基础	346
11.1	.NET Compact Framework 简介	346
11.2	Windows Mobile 智能设备程序概述	347
11.3	开发环境搭建.....	349
11.4	创建 Windows Mobile 智能设备应用程序	350
11.5	SQL Mobile 本地数据访问	352
	习题.....	370
第 12 章	移动 Web 程序开发.....	371
12.1	移动 Web 网站简介	371
12.2	移动 Web 网站创建与测试	377
12.3	ASP.NET 移动控件	381
12.4	移动 Web 窗体设计	385
	习题.....	394

第 1 章 C#编程概述

2000 年 6 月 22 日, 微软公司正式对外宣布了其 .NET 战略, 并确定每年为这个新的战略直接投入 40 亿美元的研发费用。 .NET 这个曾经模糊不定的庞大体系迅速成为业界瞩目的焦点。2000 年 9 月, 微软公司在旧金山发布了 Enterprise 2000。 同月, 微软原总裁兼首席执行官鲍尔默来到中国就“下一个互联网”主题进行演说, 在中国掀起了一股“.NET 旋风”。2000 年 11 月, 微软在 COMDEX 计算机大展上发表了 Visual Studio .NET 软件, 并展示了其 .NET 发展战略的框架体系和开发工具的相关特性, 全面加速了微软以 .NET 技术进军市场的步伐。

随着 .NET 的推出, Microsoft 强力推出了一种新型的编程语言 C# (读作 C Sharp)。C# 像 VB 一样简单, 又像 C++ 一样强大, 还有些像 Java, 但 C# 绝不是 Java 的克隆, C# 是微软在研究了 C, C++, Java, Modula2 及 SmallTalk 等大量语言的基础上推出的。与 Java 比较, C# 最大的不同之处在于它更接近于 C++, 同时 C# 也吸收了大量的新概念。例如, C# 是面向组件的语言, C# 能做到与 XML 协议最大程度的融合。同时, 在编译方式上 C# 与 Java 也很不一样。C# 的推出与 .NET 是密切相关的。

1.1 .NET 基础

1.1.1 .NET 概述

随着网络经济的到来, 微软公司希望帮助用户在任何时候、任何地方、利用任何工具都可以获得网络上的信息, 并享受网络通信所带来的快乐。 .NET 战略就是为实现这样的目标而设立的。

Microsoft .NET 平台的基本思想是将侧重点从连接到互联网的单一网站或设备上, 转移到计算机、设备和服务群组上, 使其通力合作, 提供更广泛更丰富的解决方案, 从而使用户能够控制信息的传送方式、时间和内容。计算机、设备和服务将能够相辅相成, 从而提供丰富的服务, 而不是像孤岛那样, 由用户提供唯一的集成。企业可以提供一种方式, 允许用户将它们的产品和服务无缝地嵌入自己的电子框架中。这种思路将扩展 20 世纪 80 年代由 PC 赋予的个人权限。

1.1.2 .NET 技术体系

1.1.2.1 .NET 技术特征

.NET 具有 4 个重要特点, 分别为: 基于 XML 的共同语言, 定制服务软件, 融合多平台和设备及新一代的人机界面。这 4 个特点基本上覆盖了 .NET 的技术特征。

1. 基于 XML 的共同语言

XML 是从 SGML 语言演化而来的一种标记性语言。用作元语言, 它可以定义不同种类的应用的数据交换语言。在 .NET 体系结构中, XML 用作一种应用间无缝结合的手段,

用于多种应用之间的数据采集与合并，用于不同应用之间的互操作和协同工作。具体而言，.NET 通过 XML 语言定义了简单对象访问协议(SOAP)、Web Service 描述语言(WSDL) 和 Web Service 发现协议(DISCO)。SOAP 协议提供了在无中心分布环境中使用 XML 交换结构化有类型数据的简单轻便的机制。WSDL 协议定义了服务描述文档的结构，如类型、消息、端口类型、端口和服务本身。DISCO 协议定义了如何从资源或者资源集合中提取服务描述文档和相关服务发现算法等。

2. 定制服务软件

微软原首席执行官鲍尔默在谈到软件服务时说，今天的软件产品仅仅是一张光盘，用户购买软件、亲自安装、管理和维护。但是软件服务是来自因特网的服务，它替用户安装、更新和跟踪这些软件，并让它们和用户一起在不同的机器间漫游。它为用户存储自己的信息和参考资料。图 1-1 显示了 .NET 定制服务软件的技术特征。

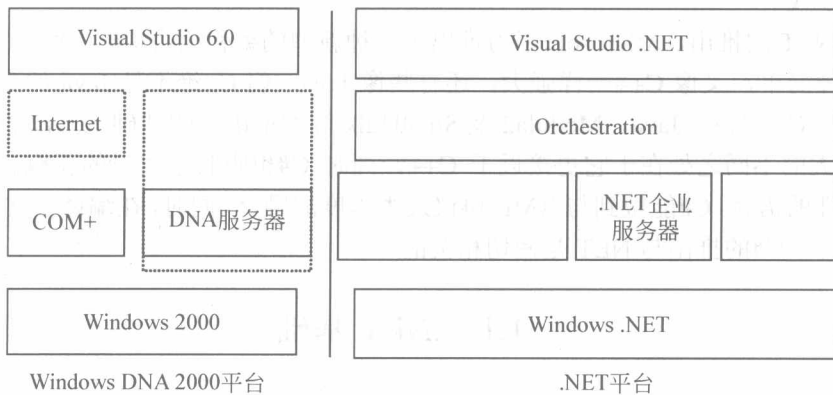


图 1-1 .NET 定制服务软件的技术特征

Orchestration 可视化编程工具产生基于 XML 的 XLANG 代码，它和 BizTalk 服务器、.NET Framework 及 Visual Studio .NET 都曾 是 Windows DNA 2000 战略的重要部分。

伴随着 ASP 产业的兴起，软件正逐步从产品形式向服务形式转化，这是整个 IT 行业的大势所趋。在 .NET 中，最终的软件应用是以 Web Service 的形式出现并在因特网上发布的。Web Service 是一种经过包装的可以在 Web 上发布的组件。.NET 通过 WSDL 协议来描述和发布这种 Web Service 信息，通过 DISCO 协议来查找相关的服务，通过 SOAP 协议进行相关的简单对象传递和调用。

微软的 .NET 战略意味着：微软公司及在微软平台上的开发者将会制造服务，而不是制造软件。

3. 融合多平台和设备

随着因特网逐渐成为一个信息和数据的中心，各种设备和服务已经或正在接入和融合因特网，成为其中的一部分。.NET 谋求与各种因特网接入设备和平台的一体化，主要关注无线设备和家庭网络设备及相关的软件和平台。

4. 新一代的人机界面

.NET 的新一代人机界面特征主要体现在“智能与互动”两个方面，包括通过自然语音、视觉、手写等多种模式的输入和表现方法，基于 XML 的可编辑符合信息构架——通过发布个性化的信息代理服务及使机器能够更好地进行自动处理的智能标记等技术。

1.1.2.2 .NET 特性

.NET 包括很多特性,这些特性包括一致的编程模式、运行于多个平台、支持多语言的综合、自动资源管理、一致的出错处理方式、安全性和 XML 语言的引入等。

➤ 一致的编程模式。在.NET 环境中,所有的应用程序都采用通用的面向对象的编程模式,而 Windows 环境则既有 DLL 函数也有 COM 对象。

➤ 简化的编程模式。在.NET 环境下,由于 CLR 的作用,在进行编程时不再需要掌握 GUIDs, Iunkown 和 AddRef 等令人头疼的 COM 知识了。

➤ 运行于多个平台。对于任何操作平台,只要支持.NET 运行时均可以运行.NET 应用程序。现在所有的 Windows 平台均可以实现这一点,将来甚至在非 Windows 操作系统上也可以实现这一点。

➤ 支持多语言的综合。按照 COM 的原理,代码重用是建立在二进制代码的级别上的。在.NET 环境下,代码重用可以建立在源码级别上,也就是说,别人用 C#语言写的某个类可以直接在 C++这样的语言中使用。.NET 有这样的巨大威力在于它为所有支持.NET 编程的语言提供了一整套通用类型系统。

➤ 自动资源管理。对于所有开发人员而言,最头疼的就是内存的处理问题。在.NET 环境下,这个问题得到彻底解决,自动资源管理功能已经纳入 CLR 之中。同时,由于增加了资源回收功能,在一定程度上安全性也得到了保障,诸如内存溢出、攻击等将得到有效控制。

➤ 一致的出错处理方式。相信所有的 Windows SDK 程序员都对 Windows 环境下混乱的错误处理方式感到厌烦,如 Win32 错误代码、异常情况处理和 HRESULT 等。在.NET 环境下,所有的程序都采用统一的错误处理方式(产生异常)。

➤ 安全性。如前所述,.NET 的出现是为了迎合下一代因特网环境下的企业级计算,一般的访问控制已经不能满足要求,所以在安全性方面,.NET 相对于 Windows 等其他系统而言,有了更深入的改进,如从装载一个类开始,就进行确认检查;在访问代码和相应资源时,实施代码访问安全措施。.NET 还提供了一整套机制来判断角色和确认身份信息,并且能做到跨进程和跨机器,从而确保所需的代码在远端不会受到破坏。.NET 的安全性也深深嵌入到 CLR 结构中,以确保应用程序本身安全。这些安全机制是对现有操作系统安全机制的一种本质上的扩展,从而使.NET 在安全性方面得到进一步加强。

➤ XML 和 SOAP 的引入。回忆一下过去的分布式应用程序的设计,通常设计两层应用程序,在此基础上出现了诸如 CORBA, IIOP, RMI 和 DCOM 这样的协议。人们已经熟悉了这样的分布式系统。但是这种系统的弊端就是灵活性差,因为这种设计方式使得应用程序固定在服务器端。而因特网是个松散连接和分布非常广的世界。原有的 Client/Server 结构已经过时,因此提出了全新的编程模式,而 XML 和 SOAP 能使这种模式很好地工作。在.NET 中,XML 和 SOAP 已经深深融入其中并成为非常重要的组成部分。

1.1.2.3 .NET 组成

.NET 主要由 Windows .NET、.NET Framework、模块构建服务和 Visual Studio .NET 组成。主要包括以下内容。

1. Windows .NET

Windows .NET 是融入了.NET 技术的 Windows,它紧密地整合了.NET 的一系列核心构造模块,为数字媒体的应用、协同工作提供支持,是微软公司的下一代 Windows 桌面平台。

2. .NET Framework

.NET Framework 的设计目的是便于开发商更容易地建立网络应用程序和 Web Service, 它的关键特色是提供了一个多语言组件开发和执行环境。从层次结构来看, .NET Framework 又包括了 3 个主要组成部分: 公共语言运行时 (Common Language Runtime, CLR)、服务框架 (Service Framework)、上层的两类应用表单——面向 Web 的网络应用程序表单 (Web Form 或 Web Service) 和 Windows 应用程序表单 (WinForm)。

公共语言运行时, 负责管理内存分配, 启动和终止线程和进程, 强化安全系数, 同时还调整组件涉及的任何其他组件、附件配置。在公共语言运行时层次之上是服务框架, 它为开发人员提供了一套能够被任何现代编程语言调用的、统一的、面向对象的、异步的、有层次结构的可扩展类库, 包括集合、输入/输出、字符串、网络、线程、全球化、安全加密、数据库访问和调试有关服务等。在服务框架层次之上是两种应用类型的表单, 一类是传统的 Windows 应用程序表单, 另一类是基于 ASP.NET 的 Web 网络应用程序表单。其中 ASP.NET 以一组控件和体系结构的方式提供了一个 Web 应用模型, 由 .NET Framework 提供的类库构建而成, 通过它可以简化 Web 应用的实现过程。

3. 模块构建服务

模块构建服务 (Building Block Services) 是 .NET 平台中的核心网络服务集合。它主要包括以下几个组成部分: 因特网 XML 通信 (使 Web 站点变成灵活的服务来交换和处理数据)、因特网 XML 数据空间 (在 Web 上提供安全的和可编程的 XML 存储空间)、因特网动态更新 (为快速开发和动态配置应用提供服务)、因特网日程安排 (集成工作、社会 and 私人的日历)、因特网身份认证 (提供从密码到生理数据等多级身份认证手段) 及因特网目录服务和因特网即时信息传递等服务。

4. Orchestration

Orchestration 是基于 XML 的面向应用的软件集成和自动化处理技术。它的目标是不受时间、组织、应用及个人的限制, 最大程度和最好地把集成技术和自动化技术结合起来, 以便商业事务能够交互并动态、可靠地进行下去。Orchestration 有几个基本要求: 处理与执行过程分离, 即整个处理并不一定非要执行的细节及途径绑定; 动态处理, 即随着整个处理过程不能对参与的平台、应用及协议等提出限制。NET 的 BizTalk Orchestration 是上述技术的一个实现, 它包括一个可视化的设计环境、一套捆绑的工具和一个 Orchestration 引擎, 用于业务流程的处理、管理和调试。

5. Visual Studio .NET

Visual Studio .NET 是基于 XML 编程工具和环境的, 它便于快速开发, 符合 .NET 体系软件服务, 使其在独立设备、企业数据中心和因特网之间传送更加容易。

1.1.3 .NET 开发环境

安装完 Visual Studio 2005 以后, 可以通过双击它的图标来启动它, 启动后的初始画面如图 1-2 所示。

下面通过创建一个新项目来介绍开发环境。选择【文件】|【新建】|【项目】菜单, 在弹出的【新建项目】对话框中选择【Visual C#项目】, 在【模板】中选择【Windows 应用程序】, 单击【确定】按钮。新建一个项目后的界面如图 1-3 所示。



图 1-2 启动后的初始画面

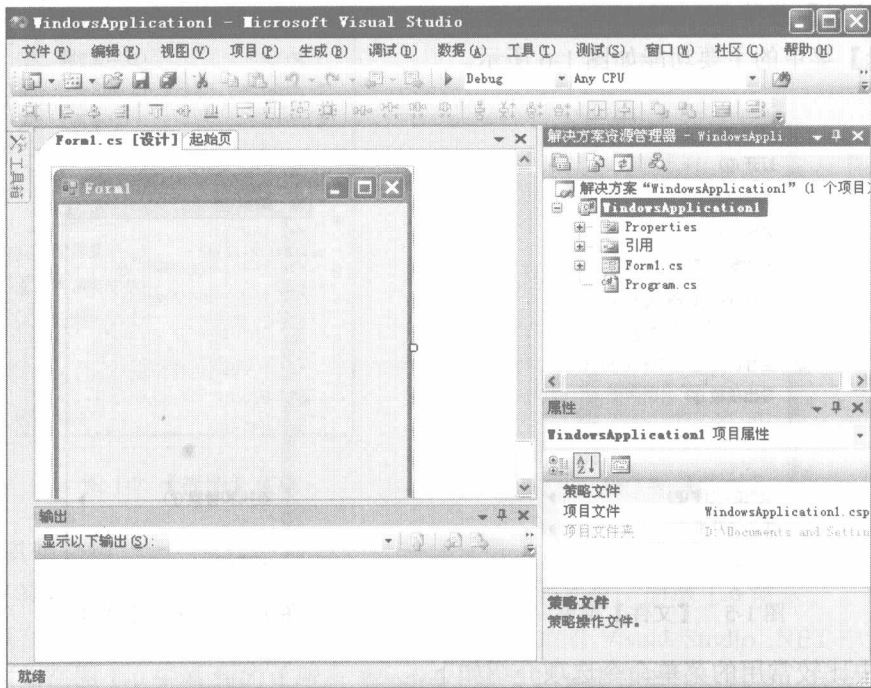


图 1-3 创建新项目后的界面

这个界面的顶部有主菜单，几乎所有的功能都可以用主菜单来完成，它主要包括文件、编辑、视图、项目、生成、调试、数据、格式、工具、测试、窗口、社区、帮助等菜单，如图 1-4 所示。

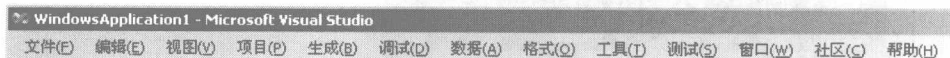


图 1-4 主菜单

1. 几个常用的菜单

1) 【文件】菜单

它的主要功能如图 1-5 所示，其中菜单后括号中的字母是快捷键，在菜单已打开时按键盘上相应的快捷字母键可以代替鼠标操作。快捷键后面有的还提供了组合键，比如 Ctrl+S 的意思是同时按下键盘上的 Ctrl 键和 S 键也可以完成“保存”的功能（无须打开菜单）。

对其中比较常用的菜单命令选项介绍如下。

【新建】菜单可以新建文件和项目。

【打开】菜单主要用来打开保存在硬盘上的项目、文件等资源。

【关闭】菜单用来关闭当前的窗口。

【打开解决方案】菜单可以打开后缀名为 .solution、.cs、.project 的文件，它既可以打开解决方案，也可以打开工程项目。

【关闭解决方案】菜单可以关闭当前解决方案中的所有文件。

【退出】菜单用于退出整个开发环境。

2) 【编辑】菜单

【编辑】菜单的主要功能如图 1-6 所示。

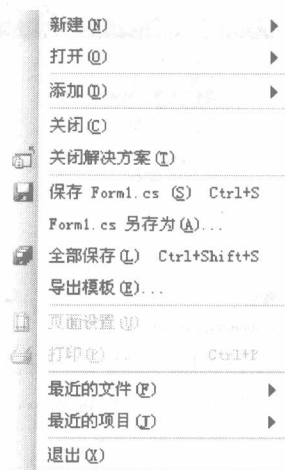


图 1-5 【文件】菜单

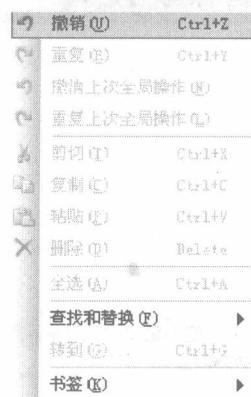


图 1-6 【编辑】菜单

对其中比较常用的菜单命令选项介绍如下。

【剪切】菜单可以把当前选中的内容剪下来，然后放到另一个地方，原来地方的内容就不存在了。

【复制】菜单功能和【剪切】菜单的内容相似，只不过它可以在原来的地方保留原来的内容。

【粘贴】菜单的作用就是把剪切或复制的内容放到需要的地方。它必须在完成【剪切】和【复制】的功能之后进行。

【删除】菜单可完成删除当前选中内容的功能。

3) 【视图】菜单

【视图】菜单的主要功能如图 1-7 所示。其中有一些很重要的视图将在后面详细介绍。

4) 【调试】菜单

【调试】菜单主要功能如图 1-8 所示。其中【窗口】菜单用于打开断点、输出和显示即时窗口，【启动调试】可使程序进入调试状态，【开始执行】菜单的功能是不调试程序而直接运行。【逐语句】菜单可使程序逐条语句执行，而【逐过程】菜单是指调试按过程（或函数）进行，即不进入过程（或函数）的内部逐条语句执行。

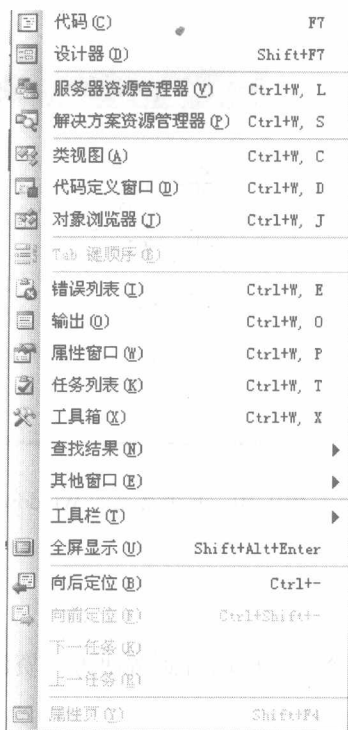


图 1-7 【视图】菜单

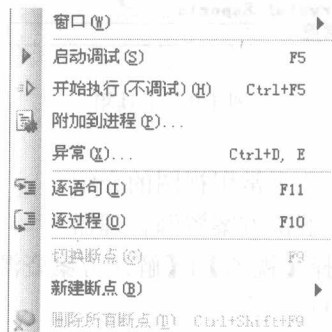


图 1-8 【调试】菜单

2. 工具箱介绍

选择【视图】|【工具箱】菜单命令，就可以看到工具箱，如图 1-9 所示。

工具箱在 .NET 的编程中有很重要的作用，它里面放着 Visual Studio .NET 中一些常用的控件，比如在图 1-9 中看到的【所有 Windows 窗体】中就有指针、标签、按钮等一些我们经常看到的控件，现在请大家试着加几个常用控件到 Form 1 上看看（具体的加法后面还会讲解到），如图 1-10 所示。

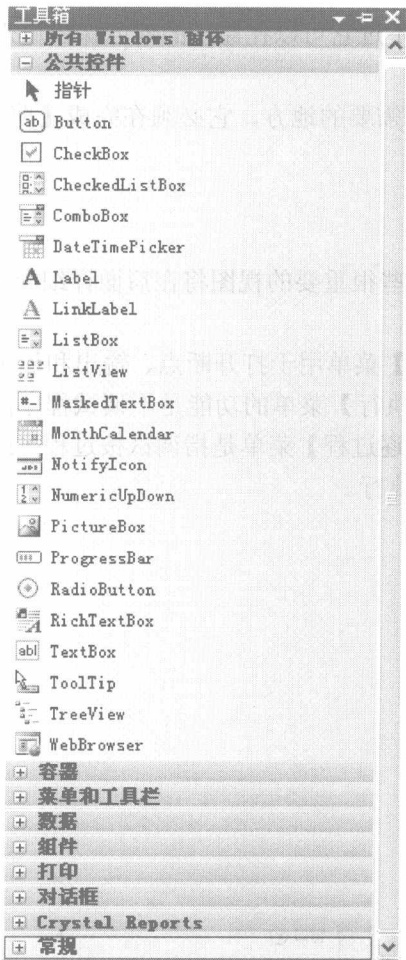


图 1-9 工具箱

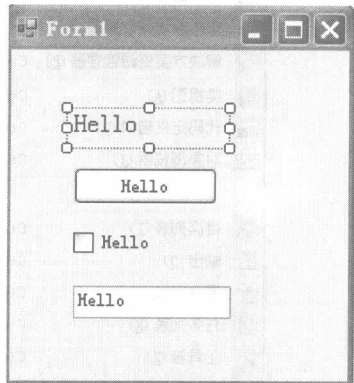


图 1-10 几个常见的 Windows 窗体控件

3. 几个常用视图的介绍

1) 解决方案资源管理器

选择【视图】|【解决方案资源管理器】菜单命令，就可以看到解决方案资源管理器，如图 1-11 所示。

通常一个工程是由很多文件组成的，解决方案资源管理器里面会显示出正在进行的工程和工程里面引用的资源。比如图 1-11 中的【引用】下面就列出了整个工程所引用的类（关于类的概念会在后面的章节中介绍）。

2) 类视图

选择【视图】|【类视图】菜单命令，就可以看到类视图，如图 1-12 所示。

通过类视图这个工具，可以非常方便地查看到整个工程引用了哪些类，还可以引用哪些类，也可以通过它来非常方便地查看类成员属性和方法的定义。（关于类的一些概念会在后面的章节中详细介绍）

3) 属性窗口

选择【视图】|【属性】菜单命令，就可以看到属性窗口，如图 1-13 所示。

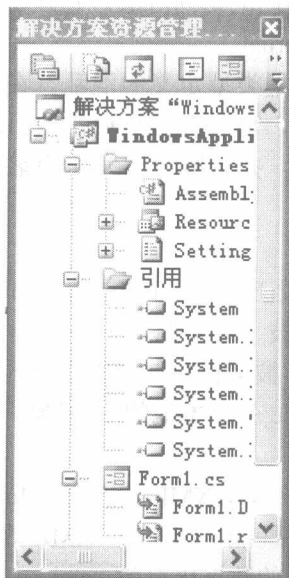


图 1-11 解决方案资源管理器

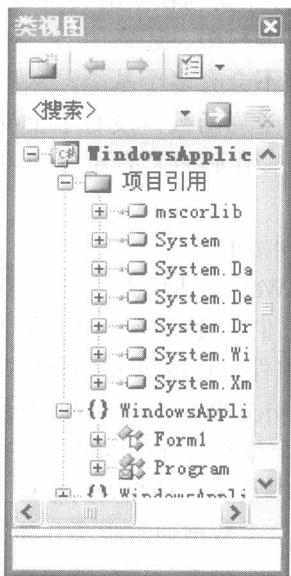


图 1-12 类视图

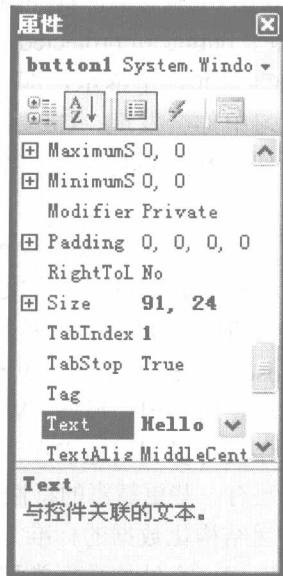


图 1-13 Form 窗口的属性

属性窗口可以列出当前所选中的控件的属性，比如图 1-13 列出的就是一个 Form 窗口的属性，一个控件会有很多个属性，比如名称、外观、颜色、字体等，可以在这个窗口中设置这些属性的值。例如，在 BackColor 后面，就可以设置这个 Form 窗体的背景颜色。

默认情况下这些视图窗口有些是打开的，有些是关闭的。使用者可以根据自己的需要，从视图菜单中打开或关闭视图窗口。

1.2 C#语言简介

C#语言自 C/C++演变而来。它在带来对应用程序的快速开发能力的同时，并没有丢失 C/C++的优点。如果是 C/C++的程序员，学习起来将会更容易。C#简化了 C++ 在类、命名空间、方法重载和异常处理等领域的工作。摒弃了 C++的复杂性，使它易使用、更少出错。对 C#的易用有贡献的是减少了 C++的一些特性，不再有宏、模板和多重继承，而增加了一些更方便的新功能，如类型安全、版本控制、垃圾收集等。所有的这些功能目标都是瞄准了开发面向组件的软件。C#的一些突出特点表示如下。

1. 简单

越来越激烈的竞争要求开发人员尽量缩短开发周期，而这正是 C#的设计思想。这种语言的设计目的是帮助开发人员减少代码量及代码出错的机会。

2. 面向对象

C#支持所有关键的面向对象的概念，如封装、继承和多态性。完整的 C#类模型是建立在 .NET 虚拟对象系统 (Virtual Object System, VOS) 基础之上的，其对象模型是 .NET