



高等学校计算机规划教材

C#.NET实用教程

■ 李纯莲 刘玉宝 刘金凤 主编
■ 祝海英 孙宇 张淑艳 副主编

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校计算机规划教材

C#.NET 实用教程

李纯莲 刘玉宝 刘金凤 主 编
祝海英 孙 宇 张淑艳 副主编

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书针对高等院校培养应用型本科人才的发展目标而编写,介绍了.NET 基本知识及应用程序的开发技术。全书共 12 章,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。本书循序渐进,内容与实例相结合,理论与实践同步,逐步引领读者掌握.NET 应用程序开发技术。书中所附实验项目针对性强,紧密联系开发实际。以“C#语言基础+SQL Server 2005+目前流行的开发技术+实验指导+课程设计指导”,构成完整的 C#语言实际开发能力的培养体系,用于培养学生 C#语言实际开发能力,使学生初步具备.NET 应用程序的独立开发能力,以便与实际工作岗位无缝接轨。本书配有 PPT、习题答案等教学资源。

本书可作为普通高等学校计算机相关专业 C#.NET 等课程的教材,同时也适合程序开发人员参考之用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容
版权所有·侵权必究

图书在版编目(CIP)数据

C#.NET 实用教程 / 李纯莲等主编. —北京: 电子工业出版社, 2011.5
高等学校计算机规划教材
ISBN 978-7-121-13359-6

I. ①C… II. ①李… III. ①C#语言—程序设计—高等学校—教学参考资料 IV. ①TP312
中国版本图书馆 CIP 数据核字(2011)第 073092 号

策划编辑: 史鹏举

责任编辑: 史鹏举

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 21 字数: 621 千字

印 次: 2011 年 5 月第 1 次印刷

定 价: 36.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

.NET 技术是微软公司为适应未来软件的发展与应用而推出的开发平台。其技术全面、先进，对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技术，跟上新技术发展的步伐是非常必要的。

本书是在编者结合自己的教学经验和教学体会，整理和丰富教学讲义的基础上编写的。本书的特点在于：能够把.NET 平台中的 C#、ASP.NET 和 SQL Server 2005 有机结合起来，理论叙述严谨，应用能力的培养目标明确，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；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 12 章，参考学时为 48~72 学时，各学校可根据实际情况进行适当的取舍。

第 1 章 C#编程概述，主要介绍.NET 基础知识、C#语言基本情况，使用 Visual Studio 2008 创建 C#应用程序的方法，以及联机帮助文件的使用。

第 2 章 C#语言基础，主要介绍 C#语言的标识符、关键字、数据类型、常量和变量、运算符、控制语句、类的定义方法等基本知识。

第 3 章 Windows 基本控件的使用，主要介绍 Microsoft .NET 2008 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，主要介绍数据库基本概念及原理，SQL Server 2005 的安装与使用，T_SQL 语言的基本应用。

第 5 章系统建模技术，主要介绍系统建模的基本概念、UML 建模和 PowerDesigner 技术。

第 6 章使用 C#和 ADO.NET 技术访问数据库，主要介绍 ADO.NET 体系结构的组成及工作原理、数据提供程序的选择、SqlConnection、OleDbConnection、OracleConnection、数据的获取、DataReader，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，主要介绍数据绑定、数据源类型、使用 DataGrid 进行数据显示，以及一个完整的开发实例。

第 8 章 Web 应用技术，主要介绍 Web 应用模型、ASP.NET 基础知识，以及 Web 服务基本知识等。

第 9 章 Web 程序设计基础，主要介绍 Web 应用程序的创建、Web 数据绑定，以及数据绑定控件的基础知识。

第 10 章网上书店系统的设计，主要以一个网上书店开发实例，介绍 Web 应用程序的完整开发过程。

第 11 章 Windows Mobile 智能设备程序设计基础，主要介绍.NET Compact Framework 基础知识、Windows Mobile 智能设备程序概述、开发环境搭建、创建 Windows Mobile 智能设备应用程序，以及 SQL Mobile 本地数据访问。

第 12 章移动 Web 程序开发，主要介绍移动 Web 网站基本知识、移动 Web 网站创建与测试、使用 Visual Studio 2008 开发移动 Web 网站，以及移动 Web 窗体设计。

精选实验项目，紧密联系开发实践，能够切实提高读者的软件开发能力。

本书配有 PPT、习题答案等教学资源，需要者可登录电子工业出版社华信教育资源网 www.hxedu.com.cn，免费注册、下载。

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。以“C#语言基础+SQL Server 2005+目前流行的开发技术+实验指导+课程设计指导”，构成完整的 C#语言实际开发能力的培养体系，用于培养学生 C#语言实际开发能力，使学生初步具备 .NET 应用程序的独立开发能力，以便与实际工作岗位无缝接轨。读者可在此基础上查阅相关的书籍，以达到更好的学习效果。

本书共 62.1 万字，由李纯莲、刘玉宝、刘金凤任主编，祝海英、孙宇、张淑艳任副主编，参加编写的人员还有戴银飞、王薇、徐大伟、赵耀红、关丽红、祝虹媛、郑舒枫。其中，刘金凤参与编写第 9、10 章，祝虹媛参与编写第 5 章，郑舒枫参与编写第 4 章。本书由刘玉宝定稿。

在本书编写的过程中，得到了单位领导和同仁的热情帮助与支持，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

目 录

第 1 章 C#编程概述	(1)	2.8.2 方法	(50)
1.1 .NET 基础	(1)	2.9 异常处理	(55)
1.1.1 .NET 概述	(1)	2.9.1 校验和非校验语句	(55)
1.1.2 .NET 技术体系	(1)	2.9.2 异常处理语句	(56)
1.1.3 .NET 开发环境	(4)	2.9.3 引发异常	(59)
1.2 C#语言简介	(8)	习题	(61)
1.3 用 Visual Studio 2008 创建 C#		第 3 章 Windows 基本控件的使用	(62)
项目	(9)	3.1 窗体	(63)
1.3.1 创建项目	(9)	3.1.1 创建窗体	(63)
1.3.2 编写工程	(10)	3.1.2 窗体属性	(64)
1.3.3 编译工程	(13)	3.1.3 窗体的常用事件和方法	(64)
1.3.4 调试工程	(14)	3.1.4 将控件加入窗体	(65)
1.4 MSDN 联机帮助	(14)	3.1.5 创建第一个 Windows 应用	(67)
1.4.1 启动联机帮助文档	(14)	程序	(67)
1.4.2 查找类的使用信息	(14)	3.2 文本类控件	(71)
习题	(15)	3.2.1 TextBox 控件	(71)
第 2 章 C#语言基础	(16)	3.2.2 Label 控件	(73)
2.1 最简单的 C#程序	(16)	3.3 按钮类控件	(74)
2.2 C#中的标识符	(16)	3.3.1 Button 控件	(74)
2.3 C#中的关键字	(17)	3.3.2 RadioButton 控件	(75)
2.4 变量和常量	(17)	3.3.3 CheckBox 控件	(75)
2.4.1 变量	(17)	3.3.4 举例	(76)
2.4.2 常量	(20)	3.4 框架类控件	(77)
2.5 数据类型	(20)	3.4.1 Panel 控件	(77)
2.5.1 值类型	(21)	3.4.2 GroupBox 控件	(78)
2.5.2 引用类型	(24)	3.4.3 TabControl 控件	(78)
2.5.3 数据类型转换	(30)	3.5 列表框、视图类控件	(80)
2.6 语句、运算符和表达式	(31)	3.5.1 ListBox 控件	(80)
2.6.1 语句	(31)	3.5.2 ComboBox 控件	(82)
2.6.2 运算符	(32)	3.5.3 ListView 控件	(83)
2.6.3 表达式	(41)	3.5.4 TreeView 控件	(85)
2.7 控制语句	(42)	3.6 滚动类控件	(88)
2.7.1 选择语句	(42)	3.6.1 HScrollBar 和 VscrollBar	(88)
2.7.2 循环语句	(45)	控件	(88)
2.8 类	(49)	3.6.2 ProgressBar 控件	(89)
2.8.1 构造函数和析构函数	(49)	3.6.3 TrackBar 控件	(89)

3.7	工具栏和状态栏控件	(90)	6.2.1	ADO.NET 组件的总体 结构	(136)
3.7.1	ToolBar 控件	(90)	6.2.2	ADO.NET DataSet 结构	(136)
3.7.2	StatusBar 控件	(93)	6.2.3	.NET Framework 数据提供 程序	(138)
3.8	定时器控件	(94)	6.2.4	选择.NET Framework 数据提供 程序	(140)
3.9	菜单控件	(95)	6.3	ADO.NET 的数据对象	(140)
习题	(99)	6.4	数据库的连接	(141)
第 4 章	数据库系统设计基础	(100)	6.4.1	SqlConnection 的使用	(141)
4.1	基本概念	(100)	6.4.2	OleDbConnection 的使用	(142)
4.1.1	数据	(100)	6.4.3	OdbcConnection 的使用	(143)
4.1.2	数据库	(100)	6.5	数据的获取	(144)
4.1.3	数据库管理系统	(100)	6.5.1	创建 Command 对象	(144)
4.1.4	数据库系统	(101)	6.5.2	执行命令	(144)
4.2	数据库模型	(101)	6.5.3	参数化查询	(148)
4.2.1	层次模型	(102)	6.5.4	执行存储过程	(149)
4.2.2	网状模型	(102)	6.6	DataReader 的使用	(151)
4.2.3	关系模型	(102)	6.6.1	DataReader 简介	(151)
4.2.4	关系数据库的设计	(103)	6.6.2	使用 DataReader 读取 数据	(152)
4.3	SQL Server 2005 的使用	(103)	6.6.3	在 DataReader 中使用多个 结果集	(154)
4.3.1	安装 SQL Server 2005	(104)	6.7	DataSet 和 DataAdapter 的 使用	(155)
4.3.2	Transact-SQL 语言的使用	(104)	6.7.1	DataSet 简介	(155)
4.3.3	SQL Server 数据库管理	(108)	6.7.2	DataAdapter 简介	(156)
4.3.4	表的管理与使用	(112)	6.7.3	利用 DataSet 和 DataAdapter 访问 数据	(157)
4.3.5	视图及其应用	(116)	6.7.4	类型和无类型 DataSet	(158)
4.3.6	索引的创建与管理	(119)	习题	(159)
习题	(120)	第 7 章	使用 C# 开发 Windows 数据库应用 程序	(160)
第 5 章	系统建模技术	(122)	7.1	数据绑定	(160)
5.1	建模概念	(122)	7.1.1	简单数据绑定	(161)
5.2	UML 建模	(122)	7.1.2	复杂数据绑定	(161)
5.2.1	UML 概念	(122)	7.2	数据源的类型	(162)
5.2.2	UML 构成	(123)	7.2.1	数组作为数据源	(162)
5.2.3	UML 应用	(131)	7.2.2	数据表作为数据源	(163)
5.3	PowerDesigner 建模	(132)	7.2.3	数据集作为数据源	(163)
5.3.1	PowerDesigner 概念	(132)			
5.3.2	PowerDesigner 构成	(132)			
5.3.3	PowerDesigner 应用	(133)			
习题	(134)			
第 6 章	使用 C# 和 ADO.NET 操作 数据库	(135)			
6.1	ADO.NET 简介	(135)			
6.2	ADO.NET 的体系结构	(136)			

7.2.4	数据视图作为数据源	(163)	10.2.2	性能需求	(233)
7.3	使用 DataGrid 显示数据	(164)	10.3	系统框架	(233)
7.4	应用实例	(166)	10.3.1	系统流程分析	(233)
习题		(176)	10.3.2	系统功能模块的分析	(233)
第 8 章	Web 应用技术	(177)	10.4	系统配置	(235)
8.1	Web 应用模型	(177)	10.4.1	软件配置	(235)
8.1.1	CGI	(177)	10.4.2	硬件配置	(236)
8.1.2	ASP	(178)	10.4.3	网络配置	(237)
8.1.3	客户端动态技术	(179)	10.5	数据库分析	(238)
8.2	ASP.NET 基础	(180)	10.5.1	E-R 图分析	(238)
8.2.1	ASP.NET 和 ASP	(180)	10.5.2	表/字段分析	(239)
8.2.2	ASP.NET 内置对象	(181)	10.6	数据库创建	(241)
8.2.3	ASP.NET 特性	(182)	10.6.1	创建数据库	(241)
8.3	System.Web 命名空间	(184)	10.6.2	创建表	(241)
8.4	ASP.NET	(185)	10.6.3	创建存储过程	(242)
8.4.1	文件处理	(185)	10.7	数据库连接	(247)
8.4.2	域名查询	(188)	10.7.1	BMS 中 Config.web 的配置 文档	(247)
8.4.3	邮件处理	(189)	10.7.2	调用 ALianBook 中 Config.web 的代码	(248)
8.4.4	事件日志记录处理	(194)	10.8	主要界面及相关代码分析	(248)
8.5	Web 服务	(198)	10.8.1	网上书店首页	(248)
8.5.1	Web 服务基础	(198)	10.8.2	书目管理页面	(250)
8.5.2	Web 服务范例	(201)	10.8.3	添加书目信息页面	(252)
8.5.3	编写 Web 服务的 Windows Forms 客户端	(204)	10.8.4	书目信息修改页面	(255)
习题		(205)	10.8.5	销售统计页面	(258)
第 9 章	Web 程序设计基础	(206)	10.8.6	书目详细信息显示页面	(261)
9.1	新建 Web 应用程序	(206)	10.8.7	购物篮信息查询页面	(263)
9.2	数据绑定技术	(206)	10.8.8	书目高级搜索页面	(265)
9.3	Web 数据绑定控件	(207)	10.9	实例演示	(267)
9.3.1	列表控件	(207)	习题		(269)
9.3.2	网格视图 GridView	(213)	第 11 章	Windows Mobile 智能设备程序设计 基础	(270)
9.3.3	单个记录显示控件 FormView	(223)	11.1	.NET Compact Framework 简介	(270)
9.3.4	数据列表 DataList	(225)	11.1.1	.NET Framework 精简版 架构	(270)
9.3.5	循环控件 Repeater	(228)	11.1.2	公共语言运行库	(270)
习题		(230)	11.1.3	.NET Framework 精简版 类库	(270)
第 10 章	网上书店系统的设计	(231)			
10.1	概述	(231)			
10.2	需求分析	(232)			
10.2.1	功能需求	(232)			

11.2	Windows Mobile 智能设备程序概述	(271)	12.2	移动 Web 网站创建与测试	(294)
11.2.1	Windows Mobile 发展前景	(271)	12.2.1	创建 Web 移动站点	(294)
11.2.2	Windows Mobile 智能设备简介	(271)	12.2.2	测试 Web 移动站点	(294)
11.2.3	Windows Mobile 智能设备程序的特点	(272)	12.3	使用 Visual Studio 2005 开发移动 Web 网站	(296)
11.3	开发环境搭建	(273)	12.3.1	移动 Web 控件简介	(296)
11.3.1	系统环境要求与配置	(273)	12.3.2	使用移动 Web 控件构建页面	(297)
11.3.2	开发环境搭建指南	(273)	12.3.3	移动 Web 控件的事件处理	(297)
11.4	创建 Windows Mobile 智能设备应用程序	(273)	12.3.4	使用容器控件	(297)
11.4.1	创建项目	(273)	12.4	移动 Web 窗体设计	(299)
11.4.2	添加代码并运行	(274)	12.4.1	简单的移动 Web 窗体	(299)
11.5	SQL Mobile 本地数据访问	(275)	12.4.2	响应控件事件	(300)
11.5.1	SQL Mobile 简介	(275)	12.4.3	显示菜单	(301)
11.5.2	创建简单的 SQL Mobile 程序	(276)	12.4.4	显示文本	(302)
11.5.3	数据访问对象	(279)	12.4.5	允许用户打电话	(302)
11.5.4	SQL Mobile 数据库的管理	(283)	12.4.6	用户输入	(303)
习题		(288)	12.4.7	输入验证	(304)
第 12 章	移动 Web 程序开发	(289)	12.4.8	自动分页	(305)
12.1	移动 Web 网站简介	(289)	习题		(305)
12.1.1	移动 Web 网站布局	(289)	实验指导		(306)
12.1.2	移动 Web 网站文件类型	(290)	实验 1	C#语言基础	(306)
12.1.3	移动 Web 网站路径	(291)	实验 2	基本控件使用	(306)
12.1.4	移动 Web 网站中的共享代码文件夹	(291)	实验 3	数据库和表管理	(308)
12.1.5	移动 Web 应用程序生命周期	(291)	实验 4	数据查询	(311)
12.1.6	移动 Web 页面生命周期	(293)	实验 5	UML 建模	(312)
			实验 6	ADO.NET 数据库访问	(313)
			实验 7	Windows 数据库应用程序开发	(315)
			实验 8	Web 应用程序开发	(319)
			实验 9	移动 Web 程序开发	(325)

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. .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，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。用做元语言，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 用做一种应用之间无缝结合的手段，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 服务描述语言(WSDL)和 Web 服务发现协议(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 语言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资源集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。

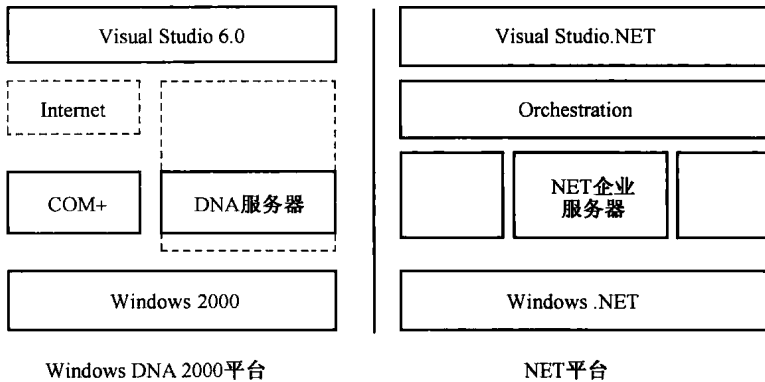


图 1-1 .NET 定制服务软件

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 服务的形式出现并在因特网上发布的。Web 服务是一种经过包装的可以在 Web 上发布的组件。.NET 通过 WSDL 语言来描述和发布这种 Web 服务信息，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、家庭网络设备及相关软件和平台。

(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——通过画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。

2. .NET 的特性

.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 平台均可以实现这点，将来甚至在非 Windows 操作系统上也可以实现这点。

● 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。但这种系统的弊端是灵活性差, 因为这种设计方式使得应用程序固定在服务器端。而因特网是整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。

3. .NET 组成

.NET 主要由 Windows .NET、.NET Framework、模块构建服务和 Visual Studio .NET 组成。这些组成部分的主要包括以下内容。

(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 服务, 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(CLR, Common Language Runtime)、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 框架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、社会和私人的日历)、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)、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 3 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 2008 以后，可以通过双击它的图标启动，启动后的初始画面如图 1-2 所示。



图 1-2 启动后的初始界面

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”菜单项，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 窗体应用程序”，输入项目名称 SampleApplication，单击“确定”按钮。新建项目后的界面如图 1-3 所示。

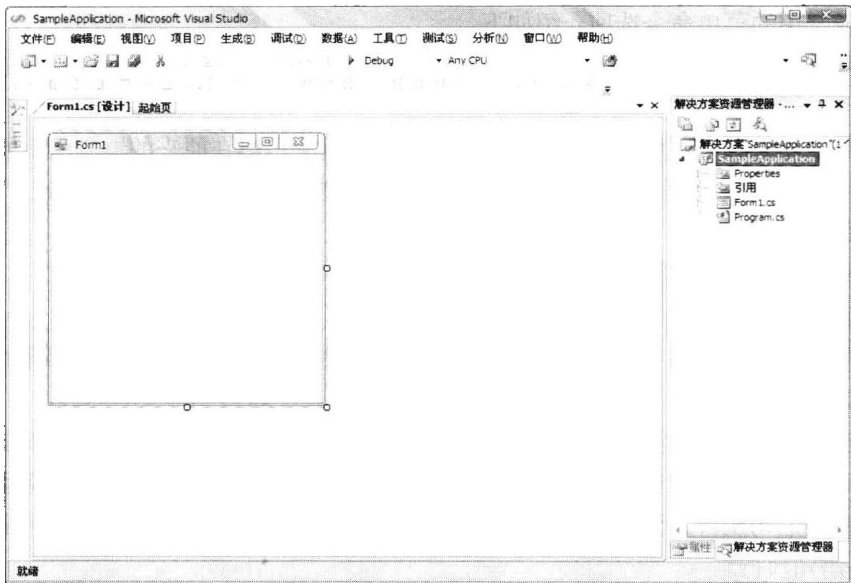


图 1-3 创建新项目后的界面

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，主要包括文件、编辑、视图、项目、生成、调试、数据、工具、测试、分析、窗口、帮助等菜单，如图 1-4 所示。

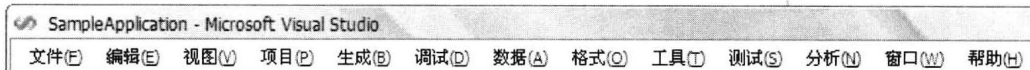


图 1-4 主菜单

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键可以代替鼠标操作。快捷键后面有的还提供了组合键，如 **Ctrl+S** 的意思是同时按下键盘上的 **Ctrl** 键和 **S** 键即可完成“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。

“打开项目/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。

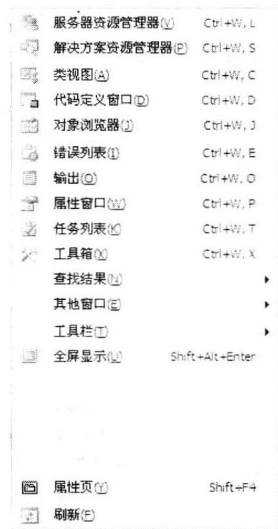


图 1-7 “视图”菜单

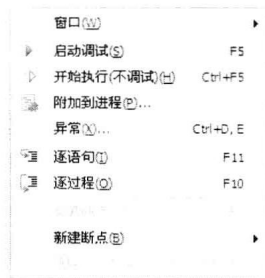


图 1-8 “调试”菜单

2. 工具箱介绍

选择“视图”→“工具箱”菜单项，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“Windows 窗体”中就有指针、标签、按钮等经常看到的控件，现在请大家试着加几个常用控件到 Form 上看看(具体的加法后面还会讲解到)，如图 1-10 所示。

3. 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”菜单项，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)。



图 1-9 工具箱

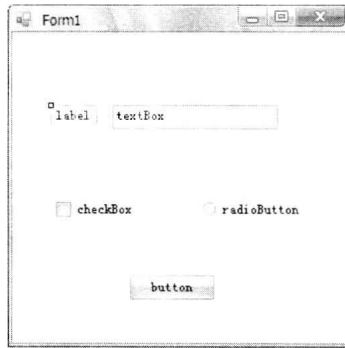


图 1-10 几个常见的 Windows 窗体控件

(2) 类视图

选择“视图”→“类视图”菜单项，就可以看到类视图，如图 1-12 所示。

通过类视图，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。

(3) 属性窗口

选择“视图”→“属性窗口”菜单项，就可以看到属性窗口，如图 1-13 所示。

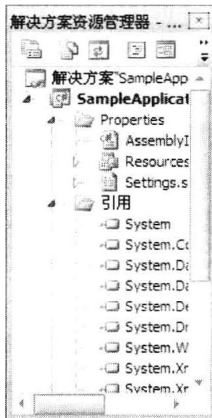


图 1-11 解决方案资源管理器

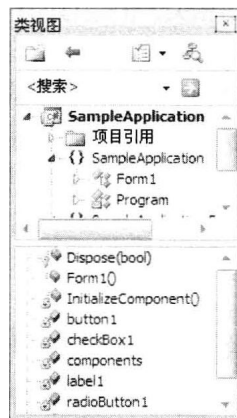


图 1-12 类视图

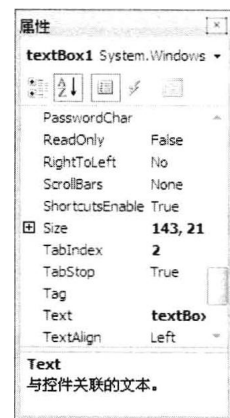


图 1-13 属性窗口

属性窗口可以列出当前所选中的控件的属性，如图 1-13 列出的就是一个 TextBox 控件的属性，一个控件会有很多个属性，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 ReadOnly 后面，可以设置这个 TextBox 控件的只读属性，如果设置为 True，则该文本框不可输入。

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++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 (VOS, Virtual Object System) 基础之上的，其对象模型是 .NET 基础构架的一部分，而不再是其本身的组成成分。下面将谈到，这样做的另一个好处是兼容性。

使用 C#编写类，程序员可以使用访问权限给类成员设置不同的访问等级。它除了支持 `private`、`public` 和 `protected` 三种访问权限，还增加了 `internal`。

C#只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。在后面的学习中将发现，C#中没有了全局函数，没有了全局变量，也没有了全局常数。所有的一切，都必须封装在一个类中。代码将具有更好的可读性，并减少了发生命名冲突的可能。

(3) 符合日渐成熟的 Web 编程标准

目前，大多数应用方案都与 Web 密切相关，但现有的开发工具都是在 Internet 和 Web 的雏形时期出现的，因此，它们一般不适合新的 Web 技术。C#语言开发建立在 Microsoft .NET 平台上，可以把组件转化成 Web 服务，从而被运行在任何平台上的任意应用程序所调用。另外，Web 服务框架让现有的 Web 服务就像本地的 C#对象，这使得程序员可以利用他们在面向对象编程方面的经验。

还有一些更精彩的特性使得 C#成为一种强大的编程工具。例如，XML 正在成为 Internet 上传递结构化数据的标准方法，这类数据通常很小。为了改善性能，C#允许 XML 数据直接映射成一个结构数据类型而不是一个类。这对处理较小数量的数据更有效。

(4) 完善的安全性及错误处理机制

语言的安全性与错误处理能力是衡量一种语言是否优秀的重要依据。任何人都会犯错误，即使是最熟练的程序员也不例外：忘记初始化变量，对于不属于自己管理范围的内存空间进行修改。这些错误常常产生难以预见的后果。一旦这样的软件投入使用，寻找与更正这些简单的错误的代价将是让人无法承受的。C#的先进设计思想可以消除软件开发中许多常见的错误，并提供了包括类型安全在内的完整的安全性能。为了减少开发中的错误，C#会帮助开发者通过更少的代码完成相同的功能，这不但减轻了编程人员的工作量，同时更有效地避免了错误的发生。

.NET 运行时库提供了代码访问的安全特性，它允许管理员和用户根据代码的 ID 来配置安全等级。在默认的情况下，从 Internet 和 Intranet 下载的代码都不允许访问任何本地的一些资源，而且异常将被触发，它将无情地被异常扔出去，若复制到本地硬盘上运行则一切正常。内存管理中的一些垃圾收集机制减轻了开发人员对内存管理的负担。.NET 平台提供的垃圾收集器 (GC, Garbage Collection) 将负责资源与对象撤销时的内存清理工作。