



高等院校计算机类课程“十二五”规划教材

C# 程序设计

C# CHENGXU SHEJI

主 编◎陈 锐 李 佳

副主编◎李绍华 方 洁 雷 军



可下载教学资源

<http://www.hfutpress.com.cn>
<http://blog.csdn.net/crcr>
nwuchenrui@126.com



高等院校计算机类课程“十二五”规划教材

C# 程序设计

主 编 陈 锐 李 佳
副主编 李绍华 方 洁 雷 军
参 编 夏敏捷 赵 娟 李爱菊

合肥工业大学出版社

内容简介

C#是一种安全的、稳定的、简单易学的面向对象编程语言。它继承了C和C++的强大功能,同时去掉了它们的一些复杂特性,成为目前使用最为广泛的开发语言,是大中专院校计算机专业的必修课程。本书主要内容包括:.NET 开发环境介绍、C#编程基础、面向对象编程基础、面向对象高级编程、目录与文件管理、类库与控件库、ADO .NET 数据库开发、Web 应用程序设计基础、ASP .NET Web 服务器控件和 ASP .NET Web 的内置对象等。

本书内容丰富,结构清晰,语言通俗,实例众多,所有程序都能够直接运行。本书可作为高等院校计算机及相关专业的教材,也可作为初级、中级程序员的参考用书。

图书在版编目(CIP)数据

C# 程序设计/陈锐,李佳主编. —合肥:合肥工业大学出版社,2012.8

ISBN 978-7-5650-0871-9

I. ①C… II. ①陈…②李… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 185760 号

C# 程序设计

陈 锐 李 佳 主 编

责任编辑 汤礼广 马小利

出 版 合肥工业大学出版社

版 次 2012 年 8 月第 1 版

地 址 合肥市屯溪路 193 号

印 次 2012 年 9 月第 1 次印刷

邮 编 230009

开 本 787 毫米×1092 毫米 1/16

电 话 总 编 室:0551-2903038

印 张 26.75

市场营销部:0551-2903198

字 数 605 千字

网 址 www.hfutpress.com.cn

印 刷 合肥学苑印务有限公司

E-mail hfutpress@163.com

发 行 全国新华书店

ISBN 978-7-5650-0871-9

定价: 54.00 元

如果有影响阅读的印装质量问题,请与出版社市场营销部联系调换。

前言

微软推出的 C# 是 .NET 中最方便和最高效的程序设计语言，在继承 C++ 和 Java 等语言优势的基础上，不仅具有封装、继承和多态等特性，而且还增加了不少创新元素，适用于桌面系统、Web 应用、数据库应用、网络应用等程序的研发。C# 语言已成为目前主流的开发工具之一。

随着 Visual Studio .NET 平台的不断升级，C# 的功能变得更强，应用开发变得更加方便。为此，我们以当前最新的 Visual Studio .NET 2008 作为平台，结合近年来教学和应用开发的实践经验，编写了本书。

本书首先介绍 C# 的特点并大致介绍 .NET 开发平台 Visual Studio .NET 2008，通过简单实例了解控制台应用程序、Windows 应用程序、ASP .NET Web 应用程序的开发过程；然后介绍 Visual Studio .NET 2008 的程序结构和调试技巧；在此基础上，系统地介绍 C# 的编程基础、面向对象编程基础、面向对象编程高级编程、窗体和控件、目录和文件管理、图形图像和多媒体编程；最后介绍 ADO .NET 和数据操作、水晶报表、Web 应用程序开发、ASP .NET Web 服务器控件、Web 应用程序开发实例。一般来说，通过对本教程的学习，特别是通过上机操作，学生可在较短的时间内，便基本能够在 Visual Studio .NET 2008 环境下用 C# 解决一些小型应用问题。本书作者多年来一直从事 C# 程序的开发、数据结构与算法、自然语言理解方面的研究工作，具有较为丰富的计算机理论基础与开发实战经验，可以说本书是作者教学和实践经验的结晶，因此，它非常适合计算机及相关专业的学生作为教材与自学参考书。

本书的特点

1. 内容丰富，讲解详细

为了方便学生学习，本书不仅介绍了 C# 语言的开发环境，而且还介绍了 C# 语言的编程基础、常用控件、图形图像编程、ADO .NET 数据访问技术、类库和控件库设计、Web 开发等内容。因此本书内容丰富，几乎覆盖了 C# 语言的全部知识点。对每个知识点，作者都结合具体的实例进行讲解，这样可使学生快速掌握 C# 语言的基础理论和初步运用。

2. 结构清晰，逻辑性强

本书按章、节和小节将学习内容划分为一个个大小不同的知识点，讲解时，循序渐进，由浅入深，最后通过实例强化知识点。因此，本书结构框架清晰，条理清晰，呈现较强的逻辑性。

3. 语言通俗，叙述简单

本书针对知识点的讲解，尽量使用比较通俗的语言。平实、通俗，让人容易亲近是本书最大的特色。

4. 配有习题，巩固知识

在每一章的最后，还配有一定数量的实践题目，学生在学习每一章的内容之后，

可以通过这些习题试着编写 C# 语言，以巩固所学的知识。

本书的内容

第 1 章：介绍了 Visual C# .NET 2008 开发环境及 C# 语言的程序结构，使学生对 C# 语言有一个初步的了解。

第 2 章：主要介绍 C# 语言的数据类型、常量与变量、运算符与表达式及 C# 的流程控制语句，使学生快速入门，能使用 Visual Studio 开发简单的应用程序。

第 3 章：主要介绍常用的数据类型：数组、字符串、集合、堆栈等，使学生初步理解 C# 中的集合类，并能熟练使用常见的类编写较为复杂的程序。

第 4 章：主要介绍面向对象编程的一些基础知识，如类、方法、属性等，使学生掌握面向对象编程方法。

第 5 章：主要介绍面向对象编程的一些比较高级的知识，如继承、多态、接口、序列化、泛型等。

第 6 章：主要介绍常用控件，帮助学生掌握商业化应用软件开发的基本技术。

第 7 章：主要介绍系统环境相关类，使学生理解目录管理方式，学会对文件进行读写操作。

第 8 章：主要介绍类库与控件库的基本知识，使学生能自己设计用户控件库和调用控件库，开发出高效的程序。

第 9 章：主要介绍 ADO .NET 数据访问方法，使学生熟练掌握数据库编程技术。

第 10 章：主要介绍 Web 页面的设计基础、Web 应用程序中的常用对象及 ASP .NET 页面中的数据传递方式，使学生熟悉 HTML、CSS 和 JavaScript 的使用方法，熟练掌握 Web 应用开发技术。

第 11 章：主要介绍 ASP .NET Web 服务器控件，使学生掌握控件属性的设置方法、事件的订阅及控件的使用方法。

第 12 章：主要介绍 ASP .NET Web 的内置对象，即 Request 对象、Response 对象、Application 对象、Session 对象和 Server 对象，使学生掌握在页面中如何使用对象及页面之间的切换、数据传递。

第 13 章：通过一个具体的案例来介绍 Web 应用程序开发，使学生掌握 Web 应用程序开发过程，深入理解三层架构的概念及三层架构的搭建过程，熟练掌握 Web 应用程序的前台和后台功能模块设计。

本书由陈锐（国家高级程序员）、李佳（重庆工商职业学院）担任主编，李绍华（大连外国语学院）、方洁（郑州轻工业学院）、雷军（焦作市教育局）担任副主编，参编的老师有夏敏捷（中原工学院）、赵娟（南阳理工学院）、李爱菊（北京联合大学）等。全书由陈锐负责统稿。

因时间仓促，错误和疏漏之处在所难免，恳请广大读者批评指正。

在使用本书的过程中，若有疑惑，或想获取本书的例题代码，请从 <http://blog.csdn.net/crcr> 或 <http://www.hfutpress.com.cn> 下载，或通过电子邮件 nwuchenrui@126.com 进行联系。

作 者

目 录

第 1 章 C# 语言概述	(1)
1.1 .NET 框架概述	(1)
1.2 Visual Studio .NET 2008 集成开发环境	(8)
1.3 Visual C# .NET 应用程序结构	(13)
1.4 C# 应用程序的一般结构	(19)
1.5 断点设置与程序调试	(23)
1.6 命名建议	(25)
第 2 章 数据类型与流程控制	(27)
2.1 数据类型	(27)
2.2 不同数据类型之间的转换	(32)
2.3 常量与变量	(35)
2.4 运算符与表达式	(37)
2.5 控制台应用程序与格式化输出	(43)
2.6 C# 流程控制语句	(45)
第 3 章 常用数据类型的用法	(61)
3.1 数组	(61)
3.2 字符串	(76)
3.3 集合	(85)
3.4 日期与时间处理	(94)
3.5 数学运算	(97)
3.6 随机数	(97)
第 4 章 面向对象编程基础	(101)
4.1 类	(101)
4.2 结构类型	(107)
4.3 方法	(109)
4.4 属性与索引器	(116)

第 5 章 面向对象高级编程	(124)
5.1 类的继承	(124)
5.2 多态	(134)
5.3 接口	(140)
5.4 委托与事件	(144)
5.5 反射	(151)
5.6 序列化与反序列化	(154)
5.7 .NET 泛型编程	(159)
第 6 章 Windows 窗体应用程序	(166)
6.1 窗体应用程序	(166)
6.2 控件的共有操作	(171)
6.3 公共控件	(173)
6.4 容器控件	(187)
6.5 菜单与工具栏控件	(190)
6.6 对话框控件	(193)
6.7 视图操作类控件	(196)
第 7 章 目录与文件管理	(204)
7.1 目录管理	(204)
7.2 文件管理	(208)
7.3 文件的读写	(210)
7.4 系统环境相关类	(222)
第 8 章 类库与控件库设计	(230)
8.1 特性 (Attribute)	(230)
8.2 类库设计	(235)
8.3 用户控件	(241)
8.4 控件库设计	(248)
第 9 章 ADO .NET 数据访问模型	(257)
9.1 ADO .NET 简介	(257)
9.2 SqlConnection 连接对象	(260)
9.3 SqlCommand 命令对象	(264)
9.4 SqlDataReader 数据读取对象	(273)
9.5 DataTable 数据表对象	(275)

9.6	DataSet 数据集对象	(278)
9.7	SqlDataAdapter 数据适配器对象	(282)
9.8	SqlParameter 参数对象	(285)
9.9	存储过程	(288)
第 10 章	网页制作基础	(298)
10.1	HTML	(298)
10.2	CSS	(305)
10.3	JavaScript	(312)
第 11 章	ASP.NET Web 服务器控件	(320)
11.1	服务器控件	(320)
11.2	标准控件	(323)
11.3	验证控件	(338)
11.4	导航控件	(347)
11.5	数据操作控件	(351)
第 12 章	ASP.NET 内置对象	(372)
12.1	Request	(372)
12.2	Reponse	(374)
12.3	Application	(375)
12.4	Session	(377)
12.5	Server	(380)
12.6	网页间的跳转	(381)
12.7	网页间的数据传递	(383)
第 13 章	Web 应用程序开发实例	(387)
13.1	系统分析与总体规划	(387)
13.2	系统框架设计	(389)
13.3	功能模块设计	(402)
参考文献		(419)

第 1 章

C# 语言概述

■ 本章导读

C# 语言和 .NET 框架简化了软件开发的复杂度。利用 C# 语言和基于 .NET 框架 Visual Studio 2008 开发平台，程序员可以很方便地开发出各种应用程序。本章主要介绍了 C# 语言的基本理论知识，并对 C#、C++、Java 三种语言进行了对比，还介绍了 .NET 框架的概念以及开发 .NET 应用程序的运行环境 Visual Studio 2008。

■ 学习目标

- (1) 了解 .NET 框架；
- (2) 了解 C# 语言与其他语言的异同；
- (3) 初步了解 C# 语言的程序结构；
- (4) 掌握 Visual Studio 2008 的程序调试技术。

1.1 .NET 框架概述

1.1.1 .NET 简介

.NET 是微软为适应 Internet 高速发展的需要，而隆重推出的新的开发平台。2003 年，Microsoft 公司发布了 Visual Studio .NET 2003，提供了在 Windows 操作系统下开发各种基于 .NET 框架 1.1 的全新应用程序开发平台。2005 年底，Microsoft 公司又发布了基于 .NET 框架 2.0 的 Visual Studio .NET 2005 开发平台，植入了适用于大型团队开发的各种功能。2007 年底，Microsoft 公司再次发布了基于 .NET 框架 3.5 的 Visual Studio .NET 2008 开发平台，引入了 250 多个新特性，整合了对象、关系型数据、XML 的访问方式，语言更加简洁。

.NET 是一种面向网络、支持各种用户终端的开发平台。利用 Visual Studio .NET 2008，用户可以非常轻松地创建具有自动伸缩能力的可靠应用程序和组件。

Visual Studio .NET 2008 集成开发环境包含 Visual Basic .NET、Visual C++

.NET、Visual C# .NET 和 Visual J# .NET 四种编程语言，并允许它们共享工具，从而有助于创建混合语言解决方案。

Visual C# .NET 是微软公司针对 .NET 平台推出的一门新语言，作为 .NET 平台的第一语言，也是微软公司推出的下一代主程序开发语言。Visual C# .NET（简称 C# 语言）几乎集中了所有关于软件开发和软件工程研究的最新成果，如面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理等。它在设计、开发程序界面时和以前的某些程序开发语言有所不同，既有 Visual Basic 快速开发的特点，又不乏 C++ 语言强大的功能。所以，C# 将来很有可能成为最主要的软件开发语言。

Visual C# .NET 提出了很多新的功能、概念和观点。掌握 Visual C# .NET，不仅要掌握语法，还需要理解并运用这些新的功能、概念和观点，另外也需要掌握 .NET 框架，理解 CLR（Common Language Runtime，公共语言运行时）。

.NET 框架（Framework）是一组用于帮助开发应用程序的类库集。Visual Studio .NET 开发平台需要此类库集的支持。.NET 框架如图 1-1 所示。

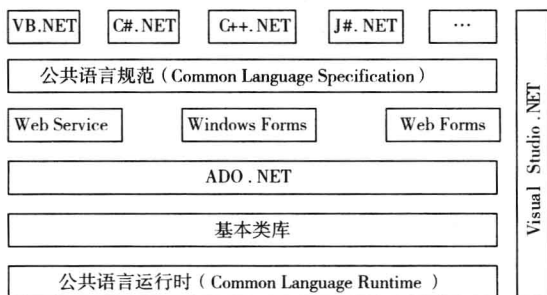


图 1-1 .NET 框架的整体结构

1.1.2 公共语言运行时（CLR）

公共语言运行时（Common Language Runtime，CLR）是 .NET 框架的基础，它负责在运行时管理代码的执行，并提供一些核心服务，如编译、内存管理、线程管理、代码执行、强制实施类型安全以及代码安全性验证。由于公共语言运行时提供代码执行的托管环境，从而提高了开发人员的工作效率并有助于开发可靠的应用程序。



说明：

以运行库为目标的代码称为托管代码；不以运行库为目标的代码称为非托管代码。

凡是使用符合公共语言规范的程序语言开发程序，均可以在任何安装有 CLR 的操作系统中执行。CLR 可以大幅度简化应用程序的开发，同时，由于代码在托管模式下运行，可以有效避免内存泄漏。使用 .NET 提供的编译器可以直接将源程序编译为 .EXE 或者 .DLL 文件。需要注意的是，此时编译出来的程序代码并不是 CPU 能直接

执行的机器代码，而是一种中间语言（Intermediate Language，IL）代码。在代码被执行时，CLR 的 Class Loader 会将需要的 IL 代码装入内存，然后再通过即时（Just - In - Time）编译方式将其临时编译成所用平台的 CPU 可直接执行的机器代码。

那么，这种二次编译是否会影响程序的运行速度，答案是肯定的，二次编译的确影响了程序的运行速度，但它却为实现跨平台带来了可能。其实，这种编译过程犹如 Java 中的 JVM（Java 虚拟机），正是 JVM 才使得 Java 能够开发出跨平台的应用程序。二次编译是 CLR 在 .NET 框架下自动实现 IL 文件到二进制文件的转变，并不需要人员的参与，所以它并不会给程序的执行带来麻烦。



说明：

在应用程序开发过程中，内存管理是一件麻烦的事情。不科学的内存管理会使应用程序不断消耗系统资源，最终导致操作系统崩溃。公共语言运行时的垃圾回收器为应用程序管理内存的分配和释放。对开发人员而言，这就意味着在开发托管应用程序时，不必编写执行内存管理任务的代码。自动内存管理可以解决常见问题，如忘记释放对象并导致内存泄漏，或尝试访问已释放对象的内存。

1.1.3 类库

类库（Class Library）提供了一组标准的系统服务，为 Web 应用程序和 Web 服务提供了基本模块。类库提供了与 Microsoft 基础类（MFC）相同的函数。与一般的 DLL 和 API 不同，这个类库是以面向对象的方式提供的。利用命名空间和在它们中定义的类，可以访问平台的任何特性。如果希望自定义类的行为，可以从已有的类中派生出自己的类。具体来说，要引用这些类，只要用 Use 语句（在 Java 中时 Import）将对应的命名空间链入程序即可。.NET 类库的概念框架如图 1 - 2 所示。

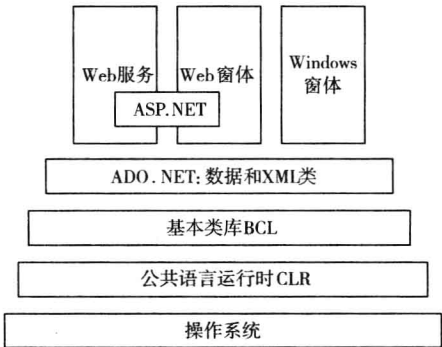


图 1 - 2 .NET 类库的概念框架

在 .NET 中，按照应用领域的不同，类库可以分为 4 类：

1. 基本类库 (Base Class Library, BCL)

BCL 中提供了输入/输出、字符串操作、安全性管理、网络通信、线程管理、文本管理及其他函数等标准功能，如图 1-3 所示。

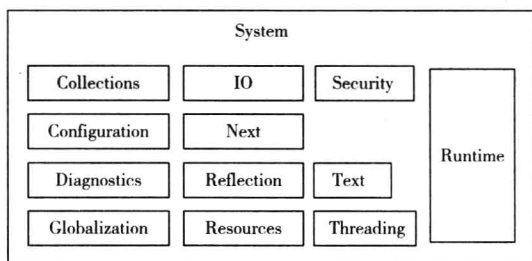


图 1-3 基本类库

每一个小模块表示一个命名空间，如 System.IO 包含输入、输出服务。

2. ADO.NET: 数据和 XML 类

ADO.NET 是下一代 ActiveX Data Objects (ADO) 技术。ADO.NET 提供了易于使用的类集，以便访问数据，同时，Microsoft 希望统一 XML 文档中的数据。因此，ADO.NET 中也提供了对 XML 的支持。ADO.NET: 数据和 XML 类中包含两个命名空间，即 System.Data 和 System.XML，如图 1-4 所示。

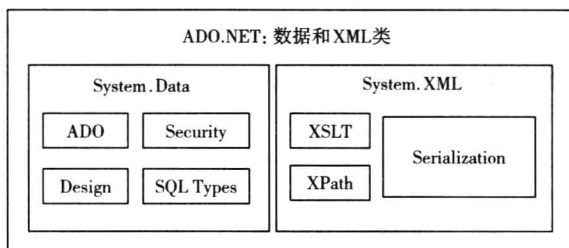


图 1-4 ADO.NET: 数据和 XML 类

ADO.NET 为 Internet 和企业开发提供了所需要的各种功能，完全支持 Web 标准及其应用，而且使用简单，扩充方便。类库分配到不同的命名空间下，其中 System 命名空间是 .NET Framework 中基本类型的根命名空间。

3. ASP.NET: Web 服务和 Web 窗体

ASP.NET 是建立在 CLR 基础上的编程框架，用来建立强大的 Web 应用程序。其中，Web 窗体为建立动态 Web 用户界面提供了简单而有效的方法，Web 服务为以 Web 作为基础的分布式应用程序提供了模块，如图 1-5 所示。

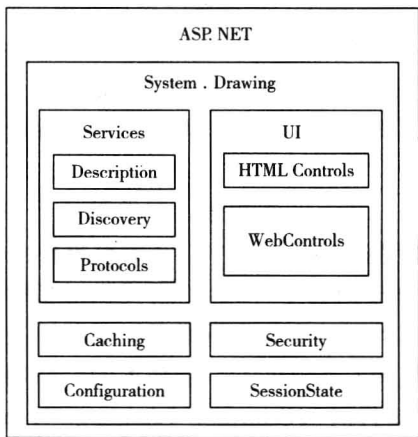


图 1-5 ASP.NET: Web 服务和 Web 窗体

4. Windows 窗体类

Windows 窗体支持一组类，通过这些类可以开发基于 Windows 的 GUI 应用程序。此外，Windows 窗体类还为 .NET 框架下的所有编程语言提供了一个公共的、一致的开发界面。Windows 窗体类包括两个命名空间：`System.WinForms` 和 `System.Drawing`，如图 1-6 所示。

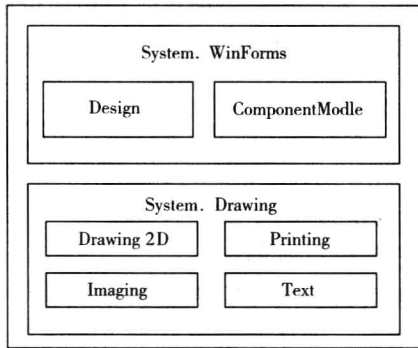


图 1-6 Windows 窗体类

.NET 开发框架在公共语言运行时的基础上，为开发者提供了完善的基本类库、数据库访问技术 ADO .NET、网络开发技术 ASP .NET，以便开发者使用多种语言及 Visual Studio 开发工具快速构建下一代的网络应用程序。随着互联网技术的发展，以后将会有越来越多的开发者采用这些技术开发出丰富、多样的下一代互联网应用产品。

1.1.4 C# 简介及其与 .NET 的关系

C 语言和 C++ 语言一直以来都是最有生命力的编程语言，这两种语言不仅功能强大，还具有高度的灵活性以及完整的底层控制能力；缺点是开发周期较长，学习和掌握这两种语言比较困难。而许多开发效率高的语言，如 Visual Basic，在功能方面又具

有局限性。于是，在选择开发语言时，许多程序设计人员面临着两难的抉择。

在这种情况下，Microsoft 公司发布了称之为 C#（C Sharp）的编程语言。C# 是为 .NET 平台量身定做的开发语言，采用面向对象的思想，支持 .NET 最丰富的基本类库资源。C# 吸收了 C++、Visual Basic、Delphi、Java 等语言的优点，体现了当今最新程序设计技术的功能。

C# 语言与 C 语言和 C++ 语言非常相似，它继承了 C 语言的语法风格，同时又继承了 C++ 语言的面向对象特性。不同的是，C# 的对象模型已经面向 Internet 进行了重新设计，使用的是 .NET 框架的类库，C# 不再提供对指针类型的支持，使得程序不能随便访问内存地址空间；C# 不再支持多重继承，避免了类层次结构中由于多重继承带来的严重后果。.NET 框架为 C# 提供了一个强大的、易用的、逻辑结构一致的程序设计环境。同时，公共语言运行时（Common Language Runtime）为 C# 程序语言提供了一个托管的运行时环境，使程序比以往更加稳定、安全。

1. C# 语言的特点

(1) 简洁的语法。C# 语言与 Java 语言相比，使用了统一的操作符，淘汰了 C++ 语言中混乱的表示符号和伪关键字，使用最简单、最常见的形式进行描述。

(2) 精心的面向对象程序设计。C# 语言是完全按照面向对象的思想来设计的，因此它具有面向对象所应有的一切特性，如封装、继承、多态等。在类的继承方面，C# 语言只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。在 C# 语言中，每种类型都是一个对象，不存在全局函数、全局变量的概念，所有常量、变量、属性、方法、索引、事件等都必须封装在类中，从而使代码具有更好的可读性，也避免了发生命名冲突的可能。

(3) 与 Web 的紧密结合。在 C# 语言中，对于复杂的 Web 编程和其他网络编程看起来更像是对本地对象进行操作，从而简化了大规模、深层次的分布式开发。用 C# 语言构建的 Web 组件能够方便地作为 Web 服务（Web Service），并且可以通过 Internet 被运行在任何操作系统上的任何语言所调用。

(4) 完整的安全性和错误处理。安全性和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，即使是最熟练的开发人员也不例外。例如，忘记变量的初始化，对不属于自己管理范围的内存空间进行修改，忘记回收不用的大对象空间等，这些错误常常导致难以预见的结果。C# 的先进设计思想有助于开发人员清除软件开发中的许多常见错误。C# 帮助开发者用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，而且有效地避免了错误发生。

(5) 版本控制。C# 语言内置了版本控制功能，如对方法重载和对接口的处理方式以及对特性（Attribute）的支持等，从而有助于开发和升级复杂的软件。

(6) 灵活性和兼容性。在托管状态下，C# 语言不能使用指针，而是使用委托（Delegate）来模拟指针的功能。如果确实需要在类或者类的方法中使用指针，只需要声明这些代码为非托管的代码即可。另外，虽然 C# 语言不支持类的多继承，但是却可以通过接口来实现多继承。兼容性是指 C# 语言允许与 Win32 API 进行交互操作，允许 C# 语言组件与其他语言组件间的交互操作等。

2. C# 语言与 C++ 语言的比较

C# 语言对 C++ 语言进行了多处改进，主要区别如下所示。

(1) 编译目标：C++ 代码直接编译为本地可执行代码，而 C# 默认编译为中间语言 (IL) 代码，执行时再通过 Just-In-Time 将需要的模块临时编译成本地代码。

(2) 内存管理：C++ 需要显式删除动态分配给堆的内存，而 C# 采用垃圾回收机制自动在合适的时机回收不再使用的内存。

(3) 指针：C++ 中大量地使用指针，而 C# 使用对类实例的引用。如果确实想在 C# 中使用指针，必须声明该内容是非安全的。一般情况下，C# 中没有必要使用指针。

(4) 字符串处理：在 C# 中，字符串是作为一种基本数据类型来处理的，因此比 C++ 中对字符串的处理要简单得多。

(5) 库：C++ 依赖于以继承和模板为基础的标准库，C# 则依赖于 .NET 基库。

(6) 继承：C++ 允许类的多继承，而 C# 只允许类的单继承。

在后面的学习中会发现，C# 与 C++ 相比还有很多不同和改进之处，包括一些细节上的差别，这里就不一一列举了。

3. C# 语言与 Java 语言的比较

从语法上讲，C# 与 Java 非常相似，只是在细节上有一些差别。实际上，C# 与 Java 的差别不是在语言本身，而是在内部功能实现上以及性能上的不同。

Java 程序需要一个运行环境 JRE (Java Runtime Environment) 来执行代码，但是，JRE 只限在 Java 语言中使用。C# 也需要一个运行环境 CLR (Common Language Runtime) 来执行代码，但是，CLR 提供了对支持 .NET 框架所有语言的支持。

Java 源代码可以被编译成字节代码的一种中间状态，然后，由已提供的虚拟机来执行这些字节代码。C# 代码也可以被编译成一种中间状态，称为中间语言 (IL)。但是，IL 代码则被传输到由 CLR 管理的执行进程上，然后通过 CLR 的 JIT 编译器编译成本地代码执行。

C# 与 Java 相比也有很多不同之处，如 C# 的文件名不受文件中类名的限制，而在 Java 中则有此限制。另外，C# 还提供了一些在 Java 中没有的功能，如运算符重载、装箱和拆箱、结构及方法隐藏等。

4. C# 语言与 VB.NET 语言的比较

与 C# 语言一样，VB.NET 也是基于 .NET Framework 和 CLR 的高级语言。但是，C# 有一些 VB.NET 不具备的独有特性，如 C# 可以使用非托管代码、移位操作符、内嵌的文档 (XML) 和运算符重载等。在发展前景上，由于 C# 一开始就是完全按面向对象的思路来设计的，而且它使用的全部是 .NET 框架定义的语法格式，不存在考虑与 .NET 之前版本兼容性的问题，因此给人的感觉是结构清晰、语法简洁。另外，C# 作为一种高级语言标准，其基本内容实现形式是公开的，因此更容易被多种平台 (如 Linux 等) 接受和广泛应用。当然，VB.NET 也有自身的优点，如以前学过 VB 的人既可以使用其原来的语法和函数，又可以使用 .NET 的语法和函数。

1.2 Visual Studio .NET 2008 集成开发环境

Visual Studio .NET 是 Microsoft 集成开发环境 (IDE) 的一个集大成者，它是一套完整的开发工具集，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++、C# 全部使用相同的集成开发环境 (IDE)，利用此 IDE 可以共享工具并且有助于创建混合语言解决方案。

目前，Microsoft 公司发布的 Visual Studio .NET 有 4 个版本：Visual Studio .NET 2002、Visual Studio .NET 2003、Visual Studio .NET 2005 和最新发布的 Visual Studio .NET 2008。

最新发布的 Visual Studio .NET 2008 是对 Visual Studio .NET 2005 全面升级的版本。Visual Studio .NET 2008 整合了对象、关系型数据、XML 的访问方式，语言更加简洁。使用 Visual Studio .NET 2008 可以高效地开发出 Windows 应用程序。Visual Studio .NET 2008 集成开发环境与以前版本的 Visual Studio 集成开发环境非常类似，如图 1-7 所示。

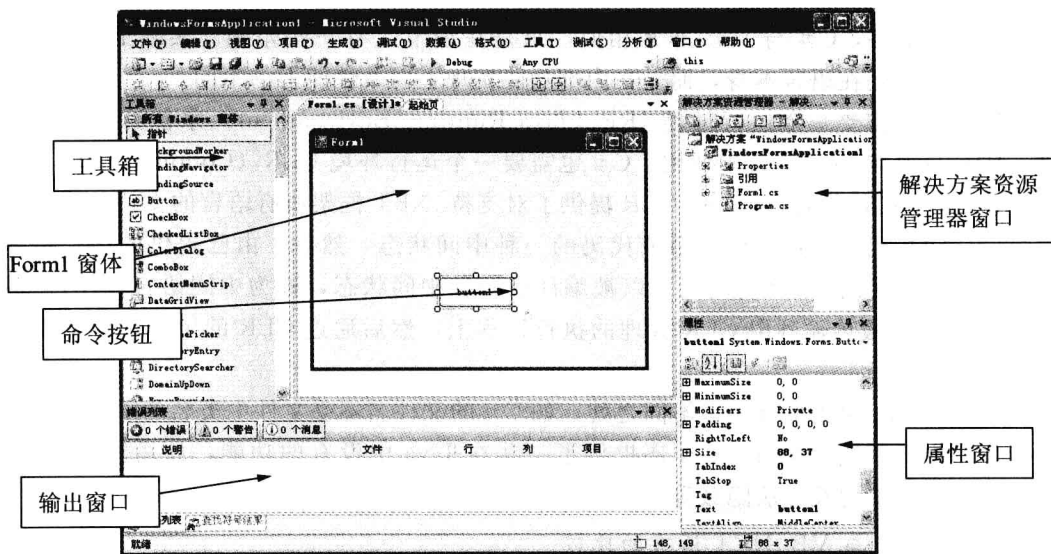


图 1-7 Visual Studio .NET 2008 集成开发环境

1.2.1 Visual Studio .NET 2008 简介

1. 解决方案资源管理器 (Solution Explorer)

使用 Visual Studio .NET 开发的每一个应用程序叫解决方案，每一个解决方案可以包含一个或多个项目。一个项目通常是一个完整的程序模块，可以有多个项。“解决方案资源管理器”子窗口显示 Visual Studio .NET 解决方案的树型结构。在解决方案资源管理器中可以浏览组成解决方案的所有项目和每个项目中的文件，可以对解决方

案的各元素进行组织和编辑。

一个项目中有很多文件，但是大部分文件（如 Properties 文件夹中的文件）不需要开发者直接进行编辑。开发者所要做的是双击解决方案资源管理器中的 Properties 文件夹图标，打开属性编辑器对项目进行配置。现在我们并不需要对项目属性做任何修改。

解决方案资源管理器标题栏下方有  工具栏，单击“所有文件”  图标按钮后，可见 bin 目录和 obj 目录，它们在生成项目执行文件时会用到。其中，obj 目录中存放的是用来创建最终可执行文件的中间代码（MSIL），目录 bin 中存放的是“二进制”文件或者应用程序最终的编译版本。需要注意的是，把 MSIL 代码称作二进制文件可能会带来误解，因为最终转换为二进制是在运行程序时由 Just In Time Compiler 来完成的。尽管如此，微软仍然把默认的项目编译输出目录称作 bin 目录。

在解决方案资源管理器中单击“查看类关系图”  图标按钮后，显示 Visual Studio .NET 当前项目中的类和类型的层次信息。在“类关系视图”中，可以对类的层次结构浏览、组织和编辑。如果双击“类关系视图”中的某一个类名称，将打开该类定义的代码视图，并定位在该类定义的开始处；如果双击类中的某一个成员，将打开该类定义的代码视图，并定位在该成员声明处。

2. 工具箱窗口

“工具箱”用于向 Windows 应用程序或 Web 应用程序添加控件，使用选项卡分类管理其中的控件。根据当前正在使用的设计器或编辑器，“工具箱”中可用的选项卡和控件会有所变化。在 Visual Studio .NET 开发环境左边的“工具箱”窗口中列出了可以应用到窗体上的控件，如图 1-8 所示。

3. 窗体窗口

在 IDE 的中部是开发环境的主窗口，用来显示指定的窗体。窗体是一小块屏幕区域，通常为矩形，可以用来向用户显示信息并接受用户的输入。设计窗体用户界面的最简单方法是将控件放在其表面上（见图 1-7），在默认情况下，窗体在设计视图状态下。在解决方案资源管理器中单击“查看代码”  图标按钮后，会切换到程序代码编辑窗口；单击“查看设计器”  图标按钮后，会切换到窗体设计视图窗口。

4. 属性（Properties）窗口

“属性”窗口如图 1-7 所示，在默认情况下位于 Visual Studio .NET 的右下角。与 IDE 的其他许多窗口一样，如果关闭了“属性”窗口，可以按 F4 键重新打开该窗口。属性窗口同时采用了两种方式管理属性和方法，分别是按分类方式和按字母顺序方式。面板的下方还有简单的帮助选项，方便开发人员对控件的属性和方法进行操作和修改。图 1-9 按字母顺序方式列出了窗体的属性。