

21 世纪大学计算机专业教材

C#与ASP.NET 程序设计

(第2版)

曹 锰 舒新峰 编著



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

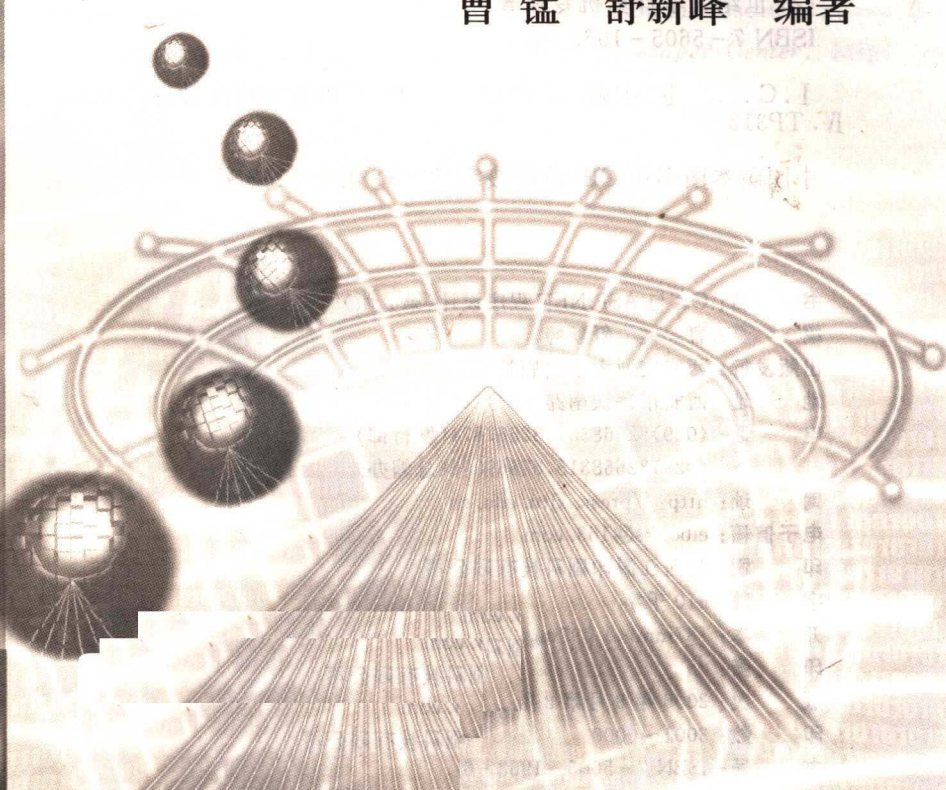
TP312
2126

21世纪大学计算机专业教材

C#与ASP.NET 程序设计

(第2版)

曹 锰 舒新峰 编著



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

· 西安 ·

内 容 提 要

本书对 C# 程序设计及使用 ASP.NET 进行 Web 应用程序开发进行了全面系统的介绍。全书共 12 章,第 1 章对 .NET 进行了概述;第 2 章对面向对象程序设计和 C# 做了简要介绍;第 3 章介绍了 C# 程序的基本结构;第 4~8 章对数据类型、变量和表达式、程序流程控制、类、继承与多态、文件等语法和编程规则进行了详细的分析说明,并通过实例介绍了相关的编程技巧;第 9~11 章详细介绍了 ASP.NET 的技术特点、服务器控件的使用、内部对象、Web 应用程序的配置以及 ADO.NET;第 12 章以“蓝天网上书店”为例,系统讲解了使用 ASP.NET 进行 Web 应用程序开发的过程与开发技巧。

本书在讲解基本知识的同时,结合大量典型范例以加深读者对知识的理解与掌握,逐步培养读者分析问题、解决问题的能力。本书每章结束时都有小结,对该章内容进行了概括,总结出重点和难点;另外,每章末附有适量习题和思考题,以帮助读者加深对该章内容的理解、提高编程能力及检验对知识的掌握程度。

本书适合作为高等学校计算机及相关专业的教材,也可供软件开发人员学习和参考。

图书在版编目(CIP)数据

C# 与 ASP.NET 程序设计 / 曹锰,舒新峰编著. —西安:
西安交通大学出版社, 2005. 6
(21 世纪大学计算机专业教材)
ISBN 7-5605-1963-6

I. C... II. ①曹... ②舒... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2005) 第 023153 号

书 名: C# 与 ASP.NET 程序设计(第 2 版)
编 著: 曹 锰 舒新峰
出版发行: 西安交通大学出版社
地 址: 西安市兴庆南路 25 号(邮编:710049)
电 话: (029)82668357, 82667874(发行部)
(029)82668315, 82669096(总编办)
网 址: <http://press.xjtu.edu.cn>
电子邮箱: eibooks@163.com
印 刷: 陕西江源印刷科技有限公司
字 数: 610 千字
开 本: 787mm×1 092mm 1/16
印 张: 25
版 次: 2006 年 8 月第 2 版 2006 年 8 月第 2 次印刷
印 数: 3001~6000
书 号: ISBN 7-5605-1963-6/TP·397
定 价: 30.00 元

版权所有 侵权必究

前言

.NET是微软公司推出的下一代互联网软件和服务战略,它代表了一个理想化的未来互联网环境和支持下一代 Internet 可编程结构的开发平台。微软的构想是建立一个“不再关注单个网站、单个设备与因特网相连的互联网环境,而是要让所有的计算机群、相关设备和服务商协同工作”的分布式网络计算环境。.NET作为下一代互联网开发平台,它建立在 XML 和因特网标准协议的基础上,包含了.NET的基础结构和基础工具,为开发新型的互动协作软件提供先进的体系结构模型。.NET平台将使分布式 Web 应用的开发和 Windows 应用开发同样简单。.NET彻底改变了软件的开发方式、发行方式和使用方式等,使软件变为服务。在.NET平台上,不同的网站通过相关的协定在程序级联系起来,网站之间形成程序级的自动交流、协同工作和相互提供全面的服务。

.NET Framework 是.NET的核心,是开发、配置和运行分布式的 Web 服务和 Web 应用的基础平台,它提供了一致的、面向对象的编程环境。无论对象代码在本地,还是在 Internet 上分布存储的,也不管是在本地执行,还是在远程执行的,所提供的编程环境都是一致的。

.NET Framework 底层的公共语言运行库 CLR(Common Language Runtime)是执行期间代码的管理者,它提供内存管理、即时编译、线程管理和远程处理等核心服务,强制实施严格的类型安全机制以确保代码的可靠性和准确性。Framework 的基础类库 BCL(Base Class Library)是一个丰富的综合性的面向对象的可用类型集合,可以为开发者提供所需的各种高级服务,为各种语言提供了统一的类库支持。

Visual Studio.NET是.NET平台上的多语言集成开发环境,在此开发环境中可以运行 VB、C++、C#、Jscript、J# 等多种语言。这些语言的程序都被编译为统一的 MSIL(微软中间语言)代码,CLR 的执行引擎为 MSIL 代码的运行提供公共的服务。执行时,在统一类库的支持下,MSIL 代码被 CLR 即时编译为本地代码受控执行。所以在.NET平台上,各种语言可以交互操作,互相调用,使每个开发人员都可以用自己熟悉的语言联合开发大型程序。

C#(读 C Sharp)是微软公司专门为 .NET平台量身订做的全新开发工具。.NET Framework 中所有的基础类库都是由 C# 编写的。C# 本身没有单独的类函数库,直接使用 Framework 所提供的类函数库。用 C# 编写的代码可以被编译为受管代码,直接受益于.NET的 CLR 提供的各种服务。因此,C# 是最适合开发.NET应用的程序设计语言。

C# 由 Turbo Pascal、Delphi 和 Visual J++的首席设计师 Anders Hejlsberg 倾心 3 年设计而完成的。它吸取了 C 语言 30 余年的过程开发经验和 C++语言 10 余年面向对象的开发经验,综合了 Visual Basic 高产和 C++底层控制能力强的特性,并继承了 Java 的优点,成为简单的、现代的、类型安全和完全面向对象的编程语言。

C# 是 C++的进一步发展,为了提高安全性,它割舍了与 C 的全兼容。它保留了 C++中好的特性,如枚举类型、引用参数、输出参数、数组参数等。对 C++ 中的艰深、晦涩的语法

语义进行了简化和改良。摒弃了 C++ 中函数及其参数的 const 修饰、宏代换、全局变量、全局函数等华而不实东西；在继承方面，采用了更安全、更容易理解的单根继承和多接口实现的方式；在源代码组织上，支持声明与实现一体的逻辑封装。它通过属性、方法、索引器、代理、事件、运算符重载、版本等，支持对组件的直接实现。它充分发挥了面向对象的所有优点。

ASP.NET 是微软公司继 ASP 之后推出的新一代 Web 开发技术。它克服了传统 Web 开发技术将脚本代码、HTML 的标记以及注释等混在一起所造成的维护和排错十分困难的缺点（在大型复杂的企业级应用程序中尤为严重）。ASP.NET 继承了现有的 Web 开发的优秀技术，又前瞻性地引入 Code-Behind 技术、服务器控件、编译执行、基于字符的分级配置、多处理器支持等新技术，使 Web 应用程序的开发和维护变得更为简单、快捷，使 Web 应用程序的执行更为安全和高效，为在 .NET Framework 上构建、管理和部署企业级 Web 应用提供了一个理想平台。

ASP.NET 建立在 .NET Framework 之上，基于面向对象技术，并且采用了程序员习惯的事件驱动的方式来开发动态网页。特别是 Web Form、Server Control 的独特设计，使开发时只需要对其属性进行必要的设置，并在相应事件中编写处理脚本即可，全部过程简单、高效，而且程序代码具有了更好的可读性与可维护性。而且，Server Control 在初始化时可以根据不同的浏览器自动生成相应的 HTML 代码，避免了不同浏览器对 HTML 代码的兼容性问题，极大地提高了开发人员的工作效率。

本书对 C# 程序设计及使用 ASP.NET 进行 Web 应用程序开发进行了全面系统的介绍。全书共 12 章，第 1 章对 .NET 进行了概述；第 2 章对面向对象程序设计和 C# 做了简要介绍；第 3 章介绍了 C# 程序的基本结构；第 4~8 章对数据类型、变量和表达式、程序流程控制、类、继承与多态、文件等语法和编程规则进行了详细的分析说明，并通过实例介绍了相关的编程技巧；第 9~11 章对 ASP.NET 的技术特点、服务器控件的使用、内部对象、Web 应用程序的配置、ADO.NET 等进行了详细介绍；第 12 章以“蓝天网上书店”为例，系统讲解了使用 ASP.NET 进行 Web 应用程序开发的过程与开发技巧。本书在讲解基本知识的同时，结合大量典型范例以加深读者对内容的理解与掌握，逐步培养读者分析问题、解决问题的能力。每一章结束后都有小结，对该章内容进行了概括，总结出重点和难点。另外，每一章后面都附有习题，以帮助读者加深对本章内容的理解、提高编程能力及检验对知识的掌握程度。

本书第 1 版 1~8 章由曹锰编写，9~12 章由舒新峰编写；修订版增加 9.1 节、10.1 节和第 13 章，由曹锰编写。作者在本校已讲授多遍 C#，根据学生学习的规律对本书内容进行了合理规划，力求循序渐进、由浅入深、层次分明、概念清晰。对于难点和重点都从学习的角度做了较详细的解释，适合于作为教材使用。书中例题丰富，所有示例的源程序都在 Microsoft 开发环境 2003 Version 7.1.3091 和 Microsoft .NET Framework Version 1.1.4322 简体中文版下调试验证。

本书第 1 版承蒙西安交通大学唐善成、沈林峰二位博士审阅书稿，并提出了重要修改意见和建议，在此表示衷心感谢。

由于作者学识有限，书中难免存在错误和不妥之处，恳请读者批评指正。

编 者

2006 年 5 月

目 录

第 1 章 .NET概述

1.1 Microsoft .NET简介	(1)
1.2 .NET的技术特征	(2)
1.3 .NET平台的组成	(3)
1.3.1 Visual Studio .NET和.NET Framework	(3)
1.3.2 公共语言运行库 CLR	(4)
1.4 Visual Studio .NET和 Framework 的系统要求	(7)
小结	(8)
习题 1	(9)

第 2 章 面向对象程序设计和 C# 简介

2.1 面向对象程序设计简介	(10)
2.1.1 程序设计语言的发展	(10)
2.1.2 面向对象的基本概念	(12)
2.1.3 面向对象的程序设计	(14)
2.2 C# 简介	(14)
2.2.1 C# 的主要特点	(15)
2.2.2 C# 与 C++ 的主要差别	(17)
小结	(18)
习题 2	(18)

第 3 章 C# 程序的基本结构

3.1 最简单的 C# 程序	(19)
3.1.1 仅含控制台输出的程序	(19)
3.1.2 含控制台输入的程序	(23)
3.1.3 输出格式控制	(23)
3.1.4 类和对象的简单实例	(26)
3.2 用 C# 开发动态链接库	(27)
3.3 Visual Studio .NET IDE 简介	(29)
3.3.1 Visual Studio .NET IDE 概述	(29)
3.3.2 用 VS.NET IDE 编辑并执行一个简单程序	(31)
小结	(33)
习题 3	(34)

第 4 章 数据类型、变量和表达式

4.1 数据类型和变量的基本概念	(35)
4.2 值类型	(36)
4.2.1 简单类型	(36)
4.2.2 枚举类型 enum	(40)
4.2.3 自定义结构类型 struct	(42)
4.3 引用类型	(44)
4.4 装箱和拆箱	(48)
4.5 常量和变量	(49)
4.5.1 常量	(49)
4.5.2 变量	(51)
4.6 类型转换	(53)
4.6.1 隐式类型转换	(53)
4.6.2 显式类型转换	(54)
4.7 运算符和表达式	(55)
4.7.1 运算符	(56)
4.7.2 特殊运算符	(60)
小结	(65)
习题 4	(66)

第 5 章 流程控制

5.1 选择语句	(68)
5.1.1 布尔表达式	(68)
5.1.2 if-else 语句	(70)
5.1.3 switch 语句	(72)
5.1.4 break 语句和 goto 语句	(72)
5.2 循环语句	(73)
5.2.1 while 语句	(73)
5.2.2 do-while 语句	(74)
5.2.3 for 语句	(75)
5.2.4 foreach 语句	(76)
5.3 预处理器指令	(79)
5.3.1 符号定义和条件编译指令	(79)
5.3.2 其它预处理器指令	(81)
5.3.3 附加属性和条件方法	(81)
5.4 异常处理	(82)
5.4.1 异常处理的过程	(83)

5.4.2 throw 语句	(83)
5.4.3 try-catch 语句	(84)
5.4.4 try-finally 语句	(86)
5.4.5 try-catch-finally 语句	(87)
5.4.6 检测控制语句 checked unchecked	(89)
小结	(90)
习题 5	(90)

第 6 章 类

6.1 类声明	(91)
6.2 类成员	(93)
6.3 常量和字段	(95)
6.3.1 常量	(95)
6.3.2 字段	(95)
6.4 方法	(97)
6.4.1 方法体	(98)
6.4.2 方法参数	(99)
6.4.3 重载(overloading)	(104)
6.4.4 外部方法	(108)
6.5 构造函数析构函数	(108)
6.6 代理类型	(116)
6.7 属性、事件和索引器	(120)
6.7.1 属性(property)	(120)
6.7.2 事件	(123)
6.7.3 索引器	(128)
6.8 应用举例	(131)
小结	(135)
习题 6	(135)

第 7 章 继承与多态

7.1 继承	(137)
7.1.1 C# 的继承机制	(137)
7.1.2 隐藏(hide)	(138)
7.2 多态性	(139)
7.2.1 虚方法和重写方法	(140)
7.2.2 抽象类	(146)
7.2.3 密封类和密封方法	(150)
7.3 组件与接口	(152)
7.3.1 组件与接口的概念	(152)
7.3.2 接口声明	(153)
7.3.3 接口成员	(154)
7.3.4 接口的实现	(156)
7.3.5 程序集和组件类	(171)
7.3.6 组件中的多态性	(173)
7.4 创作组件实例	(175)
7.4.1 创建类库项目和添加组件	(175)

7.4.2 添加构造函数和析构函数	(177)
7.4.3 将属性添加到组件类中	(178)
7.4.4 测试组件	(179)
7.4.5 理解对象生存期	(179)
小结	(181)
习题 7	(183)

第 8 章 文件

8.1 支持 I/O 操作的类	(184)
8.2 文件存储管理	(186)
8.2.1 目录管理	(186)
8.2.2 文件管理	(190)
8.3 文件读写	(192)
8.3.1 按字节流读写	(192)
8.3.2 按文本模式读写	(197)
8.3.3 按二进制模式读写	(203)
8.3.4 文件的随机读写	(205)
小结	(207)
习题 8	(208)

第 9 章 Web 开发技术概述

9.1 Web 基础知识	(209)
9.2 动态 Web 开发技术	(210)
9.2.1 动态 Web 开发技术回顾	(210)
9.2.2 动态 Web 开发的未来之路 ——ASP.NET	(213)
9.3 ASP.NET 开发环境的建立	(214)
9.3.1 软件要求	(214)
9.3.2 系统安装	(214)
9.3.3 IIS 配置	(215)
9.3.4 Microsoft SQL Server 2000 配置	(219)
小结	(222)
习题 9	(222)

第 10 章 ASP.NET 编程

10.1 超文本标记语言 HTML	(223)
10.1.1 HTML 文档的基本构成	(223)
10.1.2 用 HTML 语言制作简单网页	(224)
10.2 初识 ASP.NET 网页	(226)
10.2.1 一个简单的例子	(226)
10.2.2 ASP.NET 页面的基本构成	(229)
10.2.3 Web 网页的生命周期	(231)
10.3 Server Control	(233)
10.3.1 HTML Server Control	(233)
10.3.2 Web Server Control	(239)
10.3.3 Validation Server Control	(242)

10.4 ASP.NET 的内部对象	(250)
10.4.1 Application 对象	(250)
10.4.2 Session 对象	(251)
10.4.3 Server 对象	(252)
10.4.4 Request 对象	(254)
10.4.5 Response 对象	(256)
10.4.6 使用 Cookie	(258)
10.5 ASP.NET Web 应用程序	(262)
10.5.1 Global.asax 文件	(262)
10.5.2 在网页间传递数据	(265)
10.6 ASP.NET 配置	(270)
10.6.1 ASP.NET 配置文件的格式	(271)
10.6.2 访问 ASP.NET 配置数据	(273)
10.7 Code-Behind 技术简介	(274)
小结	(279)
习题 10	(280)

第 11 章 ADO.NET 数据库访问技术

11.1 ADO.NET 概述	(281)
11.1.1 ADO.NET 的主要特点	(281)
11.1.2 ADO.NET 的结构	(282)
11.2 Connection 对象	(283)
11.3 Command 对象	(285)
11.4 DataReader 对象	(288)
11.5 DataAdapter 对象	(290)
11.6 DataSet 对象	(292)
11.6.1 DataSet	(293)
11.6.2 DataTable	(293)
11.6.3 DataView	(296)
11.7 数据显示与维护	(298)
11.7.1 模板	(299)
11.7.2 Repeater 控件	(300)
11.7.3 DataList 控件	(302)
11.7.4 DataGrid 控件	(308)
小结	(316)

习题 11	(316)
-------------	-------

第 12 章 应用实例——蓝天网上书店

12.1 开发前的准备工作	(318)
12.2 系统设计	(318)
12.2.1 功能概述	(318)
12.2.2 系统架构	(319)
12.3 数据库设计	(320)
12.4 网页设计与实现	(324)
12.4.1 全局文件	(324)
12.4.2 网页模板	(327)
12.4.3 系统登录(主页)	(329)
12.4.4 会员注册	(334)
12.4.5 个人资料	(339)
12.4.6 图书维护	(341)
12.4.7 添加新图书	(347)
12.4.8 图书详细资料	(352)
12.4.9 新书推荐	(356)
12.4.10 分类查询	(360)
12.4.11 购物车	(361)
12.4.12 收银台	(367)
小结	(373)
习题 12	(373)

第 13 章 Web 服务

13.1 创建和调用 Web 服务	(374)
13.1.1 创建 Web 服务	(374)
13.1.2 调用 Web 服务	(376)
13.2 Web 服务的应用	(379)
习题 13	(387)

附录 C# 中的关键字

参考文献

第 1 章 .NET 概述

1.1 Microsoft.NET 简介

2000 年 6 月 22 日,微软公司推出“Microsoft.NET 下一代互联网软件和服务战略”,引起 IT 界的广泛关注。掀起一股“.NET 热潮”。2002 年 2 月,微软公司发布了 Visual Studio .NET 正式版,全面加速了.NET 进军市场的步伐。

Microsoft.NET 代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。.NET 首先是一个环境,这是一个理想化的未来互联网环境。微软的构想是一个“不再关注单个网站、单个设备与因特网相连的互联网环境,而是要让所有的计算机群、相关设备和服务商协同工作”的网络计算环境。未来的互联网,应该以一个整体服务的形式展现在最终用户面前,用户只需知道自己想要什么,而不需要一步步地在网上搜索、操作来达到自己的目的。.NET 将彻底改变软件的开发方式、发行方式和使用方式,软件将变为服务。在.NET 平台上,不同的网站通过相关的协定在程序级联系起来,网站之间形成程序级的自动交流、协同工作和相互提供全面的服务。

.NET 谋求的是一种理想的互联网环境。而要搭建这样一种互联网环境,首先需要解决的问题是针对现有因特网的缺陷,来设计和创造一种下一代 Internet 结构。这种结构不是物理网络层次上的拓扑结构,而是面向软件和应用层次的一种有别于浏览器只能静态浏览的可编程 Internet 软件结构,是以整个 Internet 资源作为编程对象的理想化的平台。因此 .NET 把自己定位为支持下一代 Internet 的可变编程结构的开发平台。

.NET 是下一代互联网开发平台,它建立在 XML 和因特网标准协议的基础上,包含了 .NET 的基础结构和基础工具,为开发新型的互动协作软件提供先进的体系结构模型。

.NET 的最终目的就是让任何用户在任何地方、任何时间以及利用任何设备都能访问他们所需要的信息、文件和程序。而用户不需知道这些东西存在什么地方,甚至连如何获得等具体细节都不知道。他们只需发出请求,然后只管接收,所有后台的复杂性是完全屏蔽起来的。对于企业的 IT 人员来说,他们也不需要管理复杂的平台,也不需要考虑各种分布应用之间的工作是如何协调的。

.NET 提供的多语言集成环境支持不同语言之间交互操作,具有强大的跨语言继承性、错误处理和调试功能。一种语言可以使用和继承另一种语言所定义的类。

.NET 包括通用语言运行库 CLR、基础类库 BCL、基于 XML 的数据存取和 Web 服务、公共语言规范 CLS。

1.2 .NET的技术特征

1. 软件变服务

传统的软件产品仅仅是存储软件的光盘,用户购买软件,亲自安装、管理和维护。而软件服务是来自因特网的服务,它替用户安装、跟踪和升级这些软件,并让它们和用户一同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。这些就是软件和软件服务各自不同的风格。

伴随着 ASP 产业的兴起,软件正逐渐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 Internet 发布的。Web 服务是一种包装后的可以在 Web 上发布的组件,.NET Web 服务描述语言(WSDL)来描述和发布这种 Web 服务信息,通过 Web 服务发现协议 DISCO(Discovery)来查找相关的服务,通过简单对象访问协议 SOAP(Simple Object Access Protocol)进行相关的简单对象传递和调用。

2. 基于 XML 的共同语言

.NET 体系中,XML 是各应用之间无缝接合的关键。

XML 是从 SGML 语言演化而来的可扩展标记语言。作为元语言,它可以定义不同种类应用的数据交换语言。在 .NET 体系结构中,XML 作为一种应用之间无缝接合的手段,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。.NET 通过 XML 语言定义了 SOAP, WSDL 和 DISCO。SOAP 协议提供了在分布式环境中使用 XML 交换结构化、有类型数据的简单机制。WSDL 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 定义了如何从资源或者资源集中提取服务描述文档、相关服务的发现算法等。

3. 融合多种设备和平台

随着 Internet 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入 Internet,成为其中的一部分。.NET 谋求与各种 Internet 接入设备包括无线设备和家庭网络设备及相关软件和平台的一体化。

4. 新一代的人机界面

新一代的人机界面主要体现在“智能与互动”两个方面。.NET 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 的可编辑复合信息架构——通用画布,个性化的信息代理服务,使机器能够更好地进行自动处理的智能标记等技术。

5. 多语言集成环境

在软件发展的历史中,软件的复用性对于开发人员仅仅是美好的憧憬。微软的组件对象模型(COM)的出现将组件改变为通用集成型的构件,使开发人员逐渐从过去的繁复编程事务中解放出来,可以选择自己得心应手的语言进行编程。然而,软件组件与应用程序间的联合仍然是松散的,开发平台限制了不同的编程语言部件间的互用性。.NET 框架的出现使一切问题迎刃而解。通过创建跨编程语言的公共 API 集,.NET 框架可提供强大的跨语言继承性、错误处理和调试功能。它的公共语言运行库(CLR)和类库紧密结合在一起,提供了不同系统之间交叉与综合的解决方案和服务,一种语言定义的类能够在另一种语言环境中使用、调试和继

承。目前支持的语言已有二十余种。

1.3 .NET平台的组成

NET 平台由以下几部分组成：

- Windows.NET
- Visual Studio.NET和.NET Framework
- .NET Enterprise Servers(企事业服务器组)
- .NET Building Block Service(构件模块服务)
- BizTalk Orchestration(交互协调处理环境)

Windows .NET是新一代的 Windows。它是.NET的基础平台,包括 Windows 2000 等一系列系统软件。提供各种服务:网站服务(IIS)、活动页面服务(ASP+)、活动目录服务(AD)、消息队列服务(MSMQ)、资料存储服务(ADO+/OLEDB)、构件服务(COM+/MTS)和分布事务处理服务(DTC)。

Visual Studio .NET是.NET的集成开发环境,.NET Framework(.NET框架)则是 .NET平台的核心部分。

1.3.1 Visual Studio.NET和.NET Framework

Visual Studio .NET是为建立 .NET Framework 应用而设置的集成开发环境(IDE),它在 .NET Framework 和公共语言规范 CLS(Common Language Specification)基础上可运行 VB、C++、C#、Jscript、J# 等多种语言。图 1.1 显示了它的体系结构。

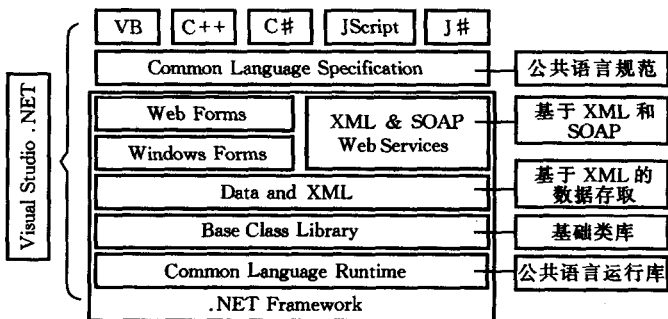


图 1.1 Visual Studio .NET 的结构

公共语言规范 CLS 定义了各种语言的必须遵守的公共标准,它是一组可以以编程方式验证的规则。这组规范控制用不同编程语言编写的类型的交互操作,包括许多应用程序所需要的基本语言功能,以确保多种语言之间的互用性。它使某种语言的对象只向调用方公开那些互相共有的功能,实现和其它语言的对象完全交互。CLS 应设计得足够大,以便能提供足够的功能满足各种语言的需要;同时又要足够小,让大多数语言都可以支持它。确切地说,它是大多数语言共有的功能的集合。

.NET Framework 是.NET的核心部分。它提供了.NET应用程序所需的核心服务。是开

发配置和运行分布式的 Web 服务和 Web 应用的基础平台。

.NET Framework 提供一致的面向对象的编程环境。无论对象代码在本地存储还是在 Internet 上分布存储的,也不管在本地执行还是在远程执行的,所提供的编程环境都是一致的。它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。

.NET Framework 提供一个安全的、最小化的代码执行环境。按照工业标准生成所有通讯,以确保基于 .NET 框架的代码可与任何其它代码集成。

.NET Framework 的最底层是公共语言运行库 CLR(Common Language Runtime)和基础类库 BCL(Base Class Library)。CLR 是 .NET Framework 的基础。BCL 是一个丰富的综合性的面向对象的可重用类型集合,是 .NET 的各种语言共享的标准类库,它通过各种命名空间为开发者提供所需的各种高级服务,为所有语言提供了统一的类库支持。

中间层 Data & XML 实现基于可扩展标记语言 XML 的数据存取。通过活动数据对象(Activex Data Object)技术向程序员公开数据访问服务的类,为创建分布式数据共享应用程序提供了一组丰富的组件,实现对关系数据、XML 和应用程序数据的访问。

顶层是应用程序模板 Windows Forms, Web Forms 和基于 XML 和 SOAP(简单对象访问协议)的 Web services,这些模板为开发 Windows 应用、Web 应用和 Web 服务提供了友好的用户界面。可以用它开发传统的命令行应用程序或图形用户界面(GUI)应用程序和基于 ASP.NET 所提供的应用程序。

与 Windows API 相比较,使用 .NET Framework 可以让编程变得非常简单。

1.3.2 公共语言运行库 CLR

CLR 是运行时的环境(或译为公共语言运行时),它是执行期间代码的管理者,提供核心服务(如内存管理、线程管理和远程处理),而且还强制实施严格的类型安全机制以确保代码的可靠性和准确性。管理代码的概念是 CLR 的基本原则。以 CLR 为目标的代码称为托管代码,而不以 CLR 为目标的代码称为非托管代码。图 1.2 表示 CLR 的结构。

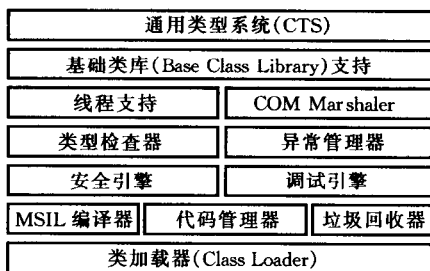


图 1.2 CLR 的结构

公共语言运行库各模块功能如下:

- 类加载器:管理元数据和类的加载和布局。
- MSIL 编译器:负责把微软中间语言(MSIL)转换到本地二进制码(即时编译)。
- 代码管理器:负责代码的执行。
- 垃圾回收器(Garbage Collector):为所有对象提供了自动的生命周期管理,是多任务的

可扩充的垃圾回收器。

- 安全引擎:提供了可信任的安全机制。
- 调试引擎:帮助调试应用程序并跟踪代码的执行。
- 类型检查器:不允许不安全的类型转换和未初始化的变量使用,MSIL 必须保证是类型安全的。
- 异常管理器:提供了结构化的异常处理。
- 线程支持:提供了多线程编程的类和接口。
- COM Marshaler:处理 .NET与 COM(Component Object Model 组件对象模型)之间相互调用的管理问题。
- 基础类库(BCL)支持:提供了支持基础类库的代码。
- 通用类型系统(CTS):解决语言的互用,统一处理类型。

从解决各种语言的互用性来看,CLR 分为通用类型系统、元数据和执行引擎。

1. 通用类型系统

通用类型系统(CTS)定义了如何在运行库中声明、使用和管理类型,同时也是运行库支持跨语言集成的一个重要组成部分。通用类型系统执行以下功能:

- 建立一个支持跨语言集成、类型安全和高性能代码执行的框架。
- 提供一个支持完整实现多种编程语言的面向对象的模型。
- 定义各语言必须遵守的规则,有助于确保用不同语言编写的对象能够交互作用。

CTS 支持值类型和引用类型两大类别的类型,细分类别如图 1.3 所示。

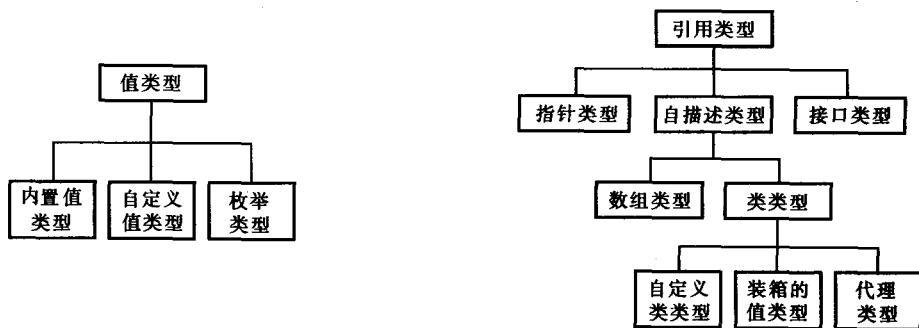


图 1.3 CTS 支持的类型

(1) 值类型直接包含它们的数据。值类型可以是内部预定义的(由运行库实现)、用户定义的或枚举的。

(2) 引用类型存储对值的内存地址的引用。引用类型可以是自描述类型、指针类型或接口类型。引用类型的类型可以由自描述类型的值来确定。自描述类型进一步细分成数组和类类型。类类型是用户定义类、装箱的值类型和代理。

作为值类型的变量,每个都有自己的数据副本,因此对一个变量的操作不会影响其它变量。作为引用类型的变量可以引用同一对象,因此对一个变量的操作会影响另一个变量所引用的同一对象。

2. 元数据

元数据(Metadata)是一种二进制信息,是“数据的数据”。元数据描述代码中的类型,包括每种类型的定义、每种类型的成员的签名、代码引用的成员和运行库在执行时使用的其它数据,可将它看作深层次的系统级属性数据。元数据是 CLR 中通用类型系统和执行引擎的联系纽带。有了元数据,一种语言中定义的类型就可以在另一种语言中使用。

元数据以非特定语言的方式来描述在代码中定义的每一类型和成员。存储以下信息:

(1) 程序集的说明

- 标识(名称、版本、区域性、公钥)。
- 导出的类型。
- 该程序集所依赖的其它程序集。
- 运行所需的安全权限。

(2) 类型的说明

- 名称、可见性、基类和实现的接口。
- 成员(方法、字段、属性、事件、嵌套的类型)。

(3) 属性

- 修饰类和成员的其它说明性元素。

在多语言混合程序中,各种语言中的类型都用形式相同的元数据来描述。使用元数据信息可以实现跨语言的类型引入。在执行时,CLR 始终根据元数据建立并管理程序中的类型,强制进行类型检查,控制着受管本地代码的执行,从而保证了跨语言的类型安全性。元数据是自描述的,它始终反映编译文件中的实际代码。元数据有助于组件间的无缝交互,使组件的互用更加简单。

3. 微软中间语言

微软中间语言 MSIL(Microsoft Intermediate Language)是可以有效地转换为本机代码且独立于 CPU 的中间语言。MSIL 包括用于加载、存储和初始化对象以及对对象调用方法的指令,还包括用于算术和逻辑运算、控制流、直接内存访问、异常处理和其它操作的指令。

在 .NET 平台上,.NET Framework 所支持的各种语言源程序,将被编译器转换为 MSIL 代码和元数据,共同存储在可移植可执行(PE)文件中。此文件基于 Microsoft PE 和公共对象文件格式(COFF)。其中,模块或程序集中定义和引用的每个类和成员都在元数据中进行说明。所有的代码都是 MSIL 代码并不是本机代码。

MSIL 的可执行文件不能直接执行。在执行前,必须将 MSIL 转换为受托管的本地 CPU 代码,这通常是通过即时(JIT)编译器完成的,公共语言运行库为它支持的每种计算机结构都提供了一种或多种 JIT 编译器。JIT 读取并解析元数据获得源程序中的类型、成员、继承等信息后,对 MSIL 代码进行即时编译产生受管本地代码,在此基础上根据源程序的要求建立类型的实例,并在执行引擎的控制之下执行。

4. CLR 执行引擎

CLR 执行引擎负责按照要求执行代码,执行引擎为 MSIL 代码的运行提供公共的服务,包括代码加载、验证、异常处理、即时编译、内存管理、安全性。图 1.4 是 CLR 的运行图。

整个执行过程包括下列步骤:

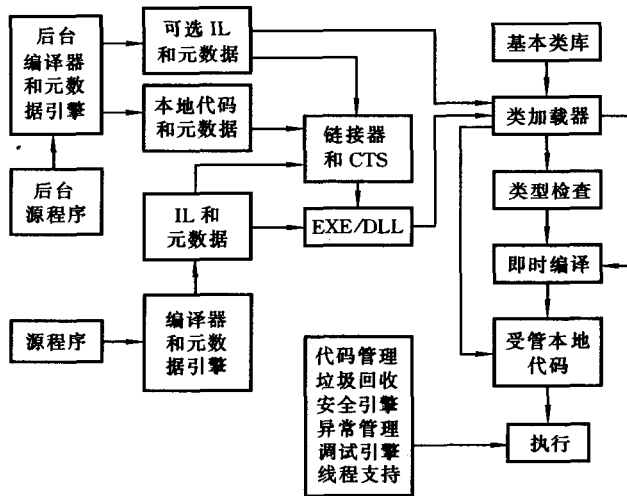


图 1.4 CLR 运行图

- 选择编译器: 选择一个或多个 CLR 支持的语言编译器。
- 将源代码翻译为 MSIL 并生成所需的元数据。
- 将 MSIL 编译为受管本机代码: 在执行时, 即时编译器将 MSIL 翻译为本机代码。在此编译过程中, 代码必须通过验证过程, 该过程检查 MSIL 和元数据以查看是否可以将代码确定为类型安全。
- 执行代码: 在执行期间, 本机代码受 CLR 的操控执行, 使用 CLR 提供的各种服务, 包括代码管理、垃圾回收、安全引擎、异常管理、调试引擎和线程支持。

1.4 Visual Studio.NET 和 Framework 的系统要求

现介绍 Visual Studio .NET 的 Microsoft 开发环境 2002 Version 7.0 和 Microsoft .NET Framework Version 1.0 的系统安装要求。

1. 安装 Visual Studio .NET 的系统要求

- CPU: Pentium II 450133 MHz
- RAM: Windows 2000 Professional 96 MB
Windows 2000 Server 192 MB
Windows NT 4.0 Workstation 64 MB
Windows NT 4.0 Server 160 MB
Windows XP Professional 160 MB
- 可用硬盘空间: 系统驱动器上有 500 MB, 安装驱动器上有 3 GB
- 显示器: 800×600, 256 色
- 光驱硬盘安装空间需要: 600 MB, 运行空间需要: 370 MB
- CD-ROM 或 DVD-ROM: 必选
- OS: Windows® 2000, Windows XP 或 Windows NT 4.0

2. 安装 .NET Framework 的系统要求

客户端:

- CPU: Pentium 90MHz
- RAM: 32 MB
- OS: 带 Service Pack 6.0a 或更高版本的 Microsoft® Windows NT® 4.0 Workstation
带 Service Pack 6.0a 或更高版本的 Microsoft® Windows NT® 4.0 Server
Microsoft® Windows® 2000 Professional
Microsoft® Windows® 2000 Server
Microsoft® Windows® 2000 Advanced Server
Microsoft® Windows® XP Home Edition
Microsoft® Windows® XP Professional

注意: 在所有这些系统上, 还要求有 Microsoft® Internet Explorer 5.01 或更高版本以及 Microsoft® Windows® Installer 2.0 或更高版本。

服务器:

- CPU Pentium 133 MHz
- RAM: 128 MB
- OS: 带 Service Pack 2.0 的 Microsoft® Windows® 2000 Professional
带 Service Pack 2.0 的 Microsoft® Windows® 2000 Server
带 Service Pack 2.0 的 Microsoft® Windows® 2000 Advanced Server
Microsoft® Windows® XP Professional

3. 附加软件

若要使用附加功能, 如 ASP.NET、COM+ 服务和 SQL Server .NET 数据提供程序, 需要附加相应软件。

客户端:

- SQL Server .NET 数据提供程序需要: Microsoft 数据访问组件 (MDAC) 2.6
- 对系统管理信息的访问需要: Windows Management Instrumentation (WMI)
- COM+ 服务需要: Windows 2000 Service Pack 2.0

服务器:

- SQL Server .NET 数据提供程序需要: Microsoft 数据访问组件 (MDAC) 2.7
- ASP.NET 需要: Microsoft Internet 信息服务 (IIS) 5.0

小 结

1. .NET 的技术特征: 软件变服务、基于 XML 的共同语言、融合多种设备和平台、新一代的人机界面、多语言集成环境。

2. .NET 平台的组成: Windows .NET, Visual Studio .NET, .NET Framework 和企业服务器组、构件模块服务、交互协调处理环境。

3. 集成开发环境 Visual Studio .NET。

4. .NET Framework 构架。
5. 公共语言运行库 CLR。
6. 集成开发环境 Visual Studio .NET 和其核心部分 .NET Framework 构架。

习 题 1

- 1.1 .NET 有哪些技术特征？
- 1.2 请介绍 Visual Studio .NET 的结构。
- 1.3 公共语言规范 CLS 起什么作用？
- 1.4 请介绍公共语言运行库 CLR 的结构及作用。
- 1.5 什么是元数据？
- 1.6 什么是微软中间语言 MSIL？
- 1.7 执行引擎 CLR 起什么作用？