Broadview
www.broadview.com.cn

.NET技术丛书

◆最主流的技术与平台
◆专为快速学习和就业而设计
◆详细的实验步骤和讲解
◆手把手带您熟悉微软技术
◆知识 + 实验 = 快速掌握 + 就业

.NET平台与C#面向对象程序设计

周羽明　刘元婷　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

.NET技术丛书

.NET平台与C#面向对象程序设计

周羽明　刘元婷　编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

微软公司一直引领 IT 行业的发展，.NET 平台占据市场绝大多数份额。对微软整体技术的把握与发展，也是大多数 IT 从业人员的必然选择。

本书带读者全面学习掌握微软.NET 平台、.NET 平台上最重要的语言 C#、C#不同的版本之间的区别、面向对象的编程思想以及 UML 的基础知识。相信通过学习，您可以全面地掌握 .NET 平台基础知识和 C#面向对象的程序设计。

本书侧重实用性，按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训试验，通过实训试验让我们以最快的速度全面掌握微软平台与技术。

本书可作为高校相关专业教材或培训班教材，也可供对.NET 技术感兴趣的从业人员阅读和参考。

图书在版编目（CIP）数据

.NET 平台与 C#面向对象程序设计 / 周羽明，刘元婷编著. —北京：电子工业出版社，2010.4
（.NET 技术丛书）
ISBN 978-7-121-10336-0

Ⅰ. N… Ⅱ. ①周… ②刘… Ⅲ. ①主页制作－程序设计 ②C 语言－程序设计 Ⅳ. TP393.092 TP312

中国版本图书馆 CIP 数据核字（2010）第 022698 号

责任编辑：李 冰
印 刷：北京天宇星印刷厂
装 订：三河市皇庄路通装订厂
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：860×1092 1/16 印张：27 字数：740 千字
印 次：2010 年 4 月第 1 次印刷
印 数：4000 册 定价：52.00 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 zlts@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

软件产业的未来是我们的

由于经济危机等不利因素的影响，世界经济处在一种不确定中。IT 行业也不能独善其身，同样面临着严峻的挑战。很多 IT 企业开始收缩产品线，裁减开发团队规模以应对这场危机。然而在这样的形势下，我们看到世界基础软件开发以及中国的外包产业却逆风飞扬，呈现出一种前所未有的所谓“危机、危机、危中寻机”的态势。

十年寒窗，等我们毕业走向社会以后，却发现自己学到的知识与社会所有脱节。特别是计算机行业，技术发展日新月异。但是在学校所学知识真的就没有用么？不！这就像武侠小说，这十年我们已经练就了内功，但是却不会一套拳法、剑法，怎么能闯荡江湖？特别是计算机专业的学生，数学和计算机基础的学习，已经让我们有了不浅的内功，只需要把这些内功发挥出来。所以，我们可能需要的就是一套武林最正派的外家功夫！

.NET 技术丛书

微软公司一直引领 IT 行业的发展，平台占据市场绝大多数份额。而对于一个计算机专业的从业人员来说，对微软整体技术的把握与发展，也是大多数 IT 从业人员的必然选择。“.NET 技术丛书”将带领我们从基础开始进入微软平台开发领域，本套丛书包含：《.NET 平台与 C#面向对象程序设计》、《.NET 平台下 Windows 程序设计》、《.NET 平台下 Web 程序设计》。三本书分别面向基础的语言与面向对象的思想，Windows 平台与 Web 平台。提供最实用的市场主流知识和技术实训实验，让我们全面掌握微软开发平台的方方面面。本套丛书全部作者均来自一线开发人员，具有多年的实践项目经验，除封面署名作者外，其他参与编写人员有：张丹、许建平、朱益铭、沈宁、俞峰。

按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训实验，通过实训实验最快地学习所有技术的一招一式。除了知识点以外，详细地讲解了 150 多个实验，手把手地带领读者从零开始，进入到.NET 开发的各个方面的知识点。200 多个基础项目实验的源码，而当我们学习知识和实验后，还有 4 个不同方向的中小型真实项目源码供我们理解，掌握它们以后就可以达到胜任著名外企开发职位或一般企业初级项目经理职位的水准。到此，我们可以真正下山，闯荡江湖了！☺

关于本书实验部分的源码

本书中涉及的所有实验都有完整的代码文件及工程文件供读者下载。

下载网站是：www.broadview.com.cn。

除此之外，我们还给读者提供了 4 个晋级的项目源代码，分别针对不同的方向，涉及 Windows 窗体、Web、网络通信、移动设备、游戏等。

希望读者通过对这 4 个晋级项目的自学，能成长为一名微软技术的高手。

项目名称	项 目 简 介
SMTP Client	SMTP 邮件客户端。通过此项目学习，让学生掌握一般的 Windows Form 项目开发。包含技术有：.NET Framework Windows 基本的控件使用，多线程编程，I/O 流，网络功能（mail），字体编码及文件格式定义保存使用
Club Site Starter Kit	入门级的 ASP.NET 2.0 站点。通过学习，学生对网络程序的开发有一定认识，对基本的数据库连接、页面与代码逻辑的结构及服务器控件编程有一定掌握
Pocket Sudoku	趣味性的 Windows Mobile 游戏。通过学习，学生熟悉掌握一般 Mobile 程序开发流程，对 Mobile 设备上的图形绘制、设备的使用、用户界面及简单的网络功能有一定认识
RSS Reader	RSS 阅读器。通过此项目学习让学生认识智能客户端的要素和一般结构，学习掌握 XML 和 RSS 技术，进一步提高.NET 开发技术。可以尝试做 RSS Reader 的 Web 版本和 Mobile 版本

适用读者

- 如果你是计算机专业的毕业生，这套书能最快地把我们大学的知识与积累，转换成为就业的资本和能力，让我们最快地发挥出我们的积累，创造机会。
- 如果你想进入计算机行业，这套书能让我们最快地学到最实用的技术，给我们带来更多的发展与工作机会，以及以后的方向。

未来是我们的！

目 录

第 1 章

微软.NET 平台介绍

1.1 .NET Framework 概述

1.1.1 Microsoft .NET 计划

.NET 是为数字生活和信息时代的每一个个体、分子、单位和组织提供支撑，并且用来建立.NET 体验的基础软件平台。该平台提供编程模型和一些工具，可编程 XML Web 服务，保证让用户，而非应用程序来控制交互和交流的过程，也是一种向用户提供个性化、简单、一致且安全的应用程序、服务和设备的方法。

微软公司推出.NET 计划的目标是将范围广泛的微软产品及其各种服务组织起来，置于各种互联设备共同的视野范围之内，其中包括服务器、固定和移动的设备等。在技术层次，微软瞄准了如下 3 个层面。

（1）Web 服务。

（2）部署平台（服务器和客户端）。

（3）开发平台。

XML Web 服务正在成为业界标准，各种异构的编程框架和软件系统要采用 XML Web 服务相互沟通。换言之，XML Web 服务是工业界为了实现软件技术服务的顺畅提供而采用的一种通信服务标准，.NET 框架可以通过提供模型和工具帮助开发人员实现这种服务。

采用 XML Web 通信标准，.NET 提供包括数据、电子邮件、商务基础结构、集成的商务进程、Web 应用和管理、安全，以及移动实时访问等服务。开发人员使用 Microsoft .NET 框架（Microsoft .NET Framework）来创建解决方案，开发实际的软件产品，帮助用户展开真正的实施。

1.1.2 .NET Framework

.NET Framework 是支持.NET 计划伟大远景的一个重要组成部分，它提供“公共语言运行规范”和基本类库来支撑面向服务的整合应用，如图 1-1 所示。

.NET Framework 是支持生成和运行下一代应用程序和 XML Web Services 的内部 Windows 组件。.NET Framework 旨在实现下列目标。

（1）提供一个一致的面向对象的编程环境，无论对象代码在本地存储执行，还是在本地执行但在 Internet 上分布或者在远程执行。

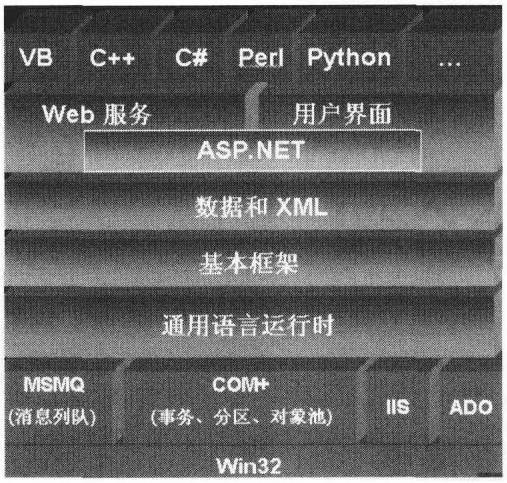

图 1-1 .NET 框架

（2）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。

（3）提供一个可提高代码（包括未知或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。

（4）提供一个可消除脚本环境或解释环境性能问题的代码执行环境。

（5）使开发人员可以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。

（6）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。

.NET Framework 的两个主要组件是公共语言运行库（Common Language Runtime，CLR，也称为“公共语言运行时”）和.NET Framework 类库（Framework Class Libraries，FCL），公共语言运行库是.NET Framework 的基础。可以将其看做一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且强制实施严格的类型安全，以及可提高安全性和可靠性的其他形式的代码准确性。事实上，代码管理的概念是公共语言运行库的基本原则。以其为目标的代码称为“托管代码”，反之称为“非托管代码”；.NET Framework 的类库是一个综合性的面向对象的可重用类型集合，可以用其开发多种应用程序，包括传统的命令行、图形用户界面（GUI）应用程序和基于 ASP.NET 所提供的最新创新的应用程序（如 Web 窗体和 XML Web Services）。

.NET Framework 可由非托管组件承载，这些组件将公共语言运行库加载到进程中并启动托管代码执行，从而创建一个可以同时利用托管和非托管功能的软件环境。.NET Framework 不但提供若干个运行库宿主，而且支持第三方运行库宿主的开发。例如，ASP.NET 承载运行库为托管代码提供可伸缩的服务器端环境。ASP.NET 直接使用运行库以启用 ASP.NET 应用程序和 XML Web Services。

Internet Explorer 是承载运行库（以 MIME 类型扩展的形式）的非托管应用程序的一个示例，用其承载运行库能够在 HTML 文档中嵌入托管组件或 Windows 窗体控件，以这种方式承载运行库使得托管移动代码（类似于 Microsoft® ActiveX®控件）成为可能。但需要进行重大改进（如不完全受信任的执行和独立的文件存储），而这种改进只有托管代码才能提供。

1.2 公共语言运行库

.NET 的目标是：一次编写，到处运行。公共语言运行库使其成为可能。

公共语言运行库的功能通过编译器和工具公开，编写利用此托管执行环境的代码具有许多优点，如跨语言集成、跨语言异常处理、增强的安全性、版本控制和部署支持、简化的组件交互模型，以及调试与分析服务等。

公共语言运行库自动处理对象布局并管理对象引用，当不再使用对象时释放它们，按这种方式实现生存期管理的对象称为“托管数据”。垃圾回收消除了内存泄漏及其他一些常见的编程错误。如果编写的代码是托管代码，则可以在.NET Framework 应用程序中使用托管数据或非托管数据，或者同时使用这两种数据。由于语言编译器会提供自己的类型（如基元类型），因此并不总能知道（或需要知道）这些数据是否托管。

有了公共语言运行库，就可以很容易地设计出对象能够跨语言交互的组件和应用程序。即用不同语言编写的对象可以互相通信，并且其行为可以紧密集成。如定义一个类，然后使用不同的语言从原始类派生出另一个类或调用原始类，还可以将一个类的实例传递到用不同的语言编写的另一个类。这种跨语言集成之所以成为可能，是因为基于公共语言运行库的语言编译器和工具使用由公共语言运行库定义的通用类型系统，而且它们遵循公共语言运行库关于定义新类型及创建、使用、保持和绑定到类型的规则。

所有托管组件都带有生成它们所基于的组件和资源的信息，这些信息构成了元数据的一部分。公共语言运行库使用这些信息确保组件或应用程序具有其需要的所有内容的指定版本，这样使代码降低了由于某些未满足的依赖项而发生中断的可能性。注册信息和状态数据不再保存在注册表中（因为在注册表中建立和维护这些信息很困难），取而代之的是定义的类型及其依赖项的信息作为元数据与代码存储在一起，这样大大降低了组件复制和移除任务的复杂性。

语言编译器和工具公开公共语言运行库功能的方式对于开发人员来说不仅很有用，而且很直观。这意味着，公共语言运行库的某些功能可能在一个环境中比在另一个环境中更突出。对公共语言运行库的体验取决于所使用的语言编译器或工具，如 Visual Basic 开发人员可能会注意到，有了公共语言运行库，面向对象的功能比以前更为强大。公共语言运行库的一些优点如下。

（1）性能得到了改进。

（2）能够轻松使用其他语言开发的组件。

（3）类库提供的可扩展类型。

（4）新的语言功能，如面向对象的编程的继承、接口和重载；允许创建多线程的可缩放应用程序的显式自由线程处理支持，以及结构化异常处理和自定义属性支持。

如果使用 Microsoft® Visual C++® .NET，则可以使用 C++托管扩展来编写托管代码。C++托管扩展提供了托管执行环境以及强大功能和富于表现力的数据类型的访问等优点。其他公共语言运行库功能如下。

（1）跨语言集成，特别是跨语言继承。

（2）垃圾回收管理对象生存期，使引用计数变得不再必要。

（3）自我描述的对象使得使用接口定义语言（IDL）不再必要。

（4）编译一次即可在任何支持公共语言运行库的 CPU 和操作系统上运行。

还可以使用 C#语言编写托管代码，该语言提供了下列优点。

（1）完全面向对象的设计。

（2）非常强的类型安全。

（3）很好地融合了 Visual Basic 的简明性和 C++的强大功能。

（4）垃圾回收。

（5）类似于 C 和 C++的语法及关键字。

（6）使用委托取代函数指针从而增强了类型安全和安全性，函数指针通过 unsafe C#关键字和 C#编译器（csc.exe）的/unsafe 选项可用于非托管代码和数据。

1.3 .NET Framework 类库

.NET Framework 基本类库提供标准功能，包括标准输入/输出、字符串操作、安全管理、网络通信、线程管理、数据操作、用户界面设计、数据库操作和字符集等功能。

使用.NET Framework 类库能够完成一系列常见编程任务，包括字符串管理、数据收集、数据库连接及文件访问等任务。除这些常见任务之外，还包括支持多种专用开发方案的类型。例如，用其开发下列类型的应用程序和服务。

（1）控制台应用程序。

（2）Windows GUI 应用程序（Windows 窗体）。

（3）ASP.NET 应用程序。

（4）XML Web Services。

（5）Windows 服务。

ADO.NET 类库允许开发人员以 XML 的形式存取数据，可以通过 OLE DB、ODBC、Oracle 及 SQL Server 等的接口来进行，从而大大丰富和简化了数据操作开发；XML 类库允许开发人员操作 XML、搜索和转化；ASP.NET 类库支持基于 Web 的应用和服务。Windows Form 类支持开发桌面应用系统。

.NET 框架类库统一了应用程序的开发模式，使用任何一种.NET 支持的编程语言，均可自如使用.NET 类库中提供的强大功能。在.NET 框架上使用一种开发语言开发的代码和应用能被轻而易举地使用到其他语言开发的项目和系统中，掌握多种开发语言的开发人员之间的协作变得毫无悬念。

所有的类库都为.NET 框架所支持的开发语言提供了一个通用且一致的开发接口，这些语言包括 C++、Microsoft® Visual Basic、J#®和 Microsoft 的最新语言 C#。后来，也有大量的第三方语言可以用来创建.NET 框架应用程序，包括 COBOL、Eiffel、Perl、Python 和 Smalltalk 等。这是微软公司在人性化方面努力作为的一个体现，它既要保有共性，又要照顾到选择的差异性，以及尊重其他非原生语言的价值。

1.3.1 .NET Framework 类库概述

.NET Framework 包括类、接口和值类型，它们可加速和优化开发过程并提供对系统功能的访问。为便于语言之间进行交互操作，.NET Framework 类型是符合公共语言规范（CLS）的，只要这种语言的编译器符合公共语言规范。就可在任何编程语言中使用。

.NET Framework 类型是生成.NET 应用程序、组件和控件的基础，它执行下列功能。

（1）表示基础数据类型和异常。

（2）封装数据结构。

（3）执行 I/O。

（4）访问关于加载类型的信息。

（5）调用.NET Framework 安全检查。

（6）提供数据访问、多客户端 GUI 和服务器控制的客户端 GUI。

.NET Framework 提供一组丰富的接口以及抽象类和具体类（非抽象类）。可以按原样使用这些具

体的类，或者在多数情况下从这些类派生自己的类。若要使用接口的功能，既可以创建实现接口的类，也可以从某个实现接口的.NET Framework 类中派生类。

1.3.2　命名约定

.NET Framework 类型使用点语法命名方案，该方案隐含了层次结构。此技术将相关类型分为不同的命名空间组，以便更容易地搜索和引用它们。全名的第 1 部分即最右边的点之前的内容是命名空间名，最后一部分是类型名。例如，System.Collections.ArrayList 表示 ArrayList 类型，该类型属于 System.Collections 命名空间。System.Collections 中的类型可用于操作对象集合。

此命名方案使扩展.NET Framework 的库开发人员可以轻松创建分层类型组，并用一致且带有提示性的方式为其命名。库开发人员在创建命名空间的名称时应使用“公司名称.技术名称”原则，如 Microsoft.Word 命名空间符合此原则。

利用命名模式将相关类型分组为命名空间是生成和记录类库的一种非常有用的方式，但是此命名方案对可见性、成员访问、继承、安全性和绑定无效。一个命名空间可以被划分在多个程序集中，而单个程序集可以包含来自多个命名空间的类型，程序集为公共语言运行库中的版本控制、部署、安全性、加载和可见性提供外形结构。

1.3.3　重要命名空间说明

（1）System 命名空间

该命名空间是.NET Framework 中基本类型的根命名空间，其中包括表示由所有应用程序使用的基础数据类型的类，如 Object（继承层次结构的根）、Byte、Char、Array、Int32 和 String 等。在这些类型中，有许多与编程语言所使用的基元数据类型相对应。若使用.NET Framework 类型编写代码，可以在使用.NET Framework 基础数据类型时使用编程语言的相应关键字。

表 1-1 所示为.NET Framework 提供的一些值类型，其简要描述了每个类型，并指出 Visual Basic、C#和 C++中的相应类型。表中还包括 Object 和 String 类的项，这些项在许多语言中都有相应的关键字。

表 1-1　.NET Framework提供的一些值类型

类　别	类　名	说　　明	Visual Basic 数据类型	C#数据类型	C++数据类型	JavaScript 数据类型
整数	Byte	8 位无符号整数	Byte	byte	char	Byte
	SByte	8 位有符号整数 不符合 CLS	SByte	sbyte	signed char	SByte
	Int16	16 位有符号整数	Short	short	short	short
	Int32	32 位有符号整数	Integer	int	int 或 long	int
	Int64	64 位有符号整数	Long	long	_int64	long
	UInt16	16 位无符号整数 不符合 CLS	UShort	ushort	unsigned short	UInt16
	UInt32	32 位无符号整数 不符合 CLS	UInteger	uint	unsigned int 或 unsigned long	UInt32
	UInt64	64 位无符号整数 不符合 CLS	ULong	ulong	unsigned _int64	UInt64

续表

类 别	类 名	说 明	Visual Basic 数据类型	C#数据类型	C++数据类型	JavaScript 数据类型
浮点	Single	单精度（32 位）浮点数字	Single	float	float	float
	Double	双精度（64 位）浮点数字	Double	double	double	double
逻辑运算	Boolean	布尔值（真或假）	Boolean	bool	bool	bool
其他	Char	Unicode（16 位）字符	Char	char	wchar_t	char
	Decimal	十进制（128 位）值	Decimal	decimal	Decimal	Decimal
	IntPtr	大小取决于基础平台的有符号整数(32 位平台上为 32 位值，64 位平台上为 64 位值)	IntPtr 无内置类型	IntPtr 无内置类型	IntPtr 无内置类型	IntPtr
	UIntPtr	大小取决于基础平台的无符号整数(32 位平台上为 32 位值，64 位平台上为 64 位值) 不符合 CLS	UIntPtr 无内置类型	UIntPtr 无内置类型	UIntPtr 无内置类型	UIntPtr
类对象	Object	对象层次结构的根	Object	object	Object*	Object
	String	Unicode 字符的不变的定长串	String	string	String*	String

除基础数据类型外，System 命名空间还包含 100 多个类。范围从处理异常的类到处理核心运行时概念的类，如应用程序域和垃圾回收器，并且包含许多二级命名空间。

（2）System.Collections

该命名空间包含定义各种对象集合（如列表、队列、位数组、哈希表和字典等）的接口和类。

（3）System.Collections.Generic

该命名空间包含定义泛型集合的接口和类，泛型集合允许用户创建强类型集合，它比非泛型强类型集合提供更好的类型安全和性能。

（4）System.Configuration

该命名空间包含提供用于处理配置数据的编程模型的类型。

（5）System.Data

该命名空间包含组成大部分 ADO.NET 结构的类，这种结构会生成可用于有效管理来自多个数据源的数据的组件。在断开连接的方案（如 Internet）中，它提供了一些可以在多层系统中请求、更新和协调数据的工具。ADO.NET 结构也可以在客户端应用程序（如 Windows 窗体）或 ASP.NET 创建的 HTML 页中实现。

（6）System.Data.Common

该命名空间包含由.NET Framework 数据提供程序共享的类，这种提供程序描述用于在托管空间中访问数据源的类的集合。

（7）System.Data.Design

该命名空间包含可用于生成自定义的类型化数据集的类。

（8）System.Data.Odbc

该命名空间包含封装 ODBC .NET Framework 数据提供程序的类，这种提供程序描述了用于在托管空间中访问 ODBC 数据源的类集合。