普通高等教育“十一五”国家级规划教材　计算机系列教材

C#程序设计基础教程

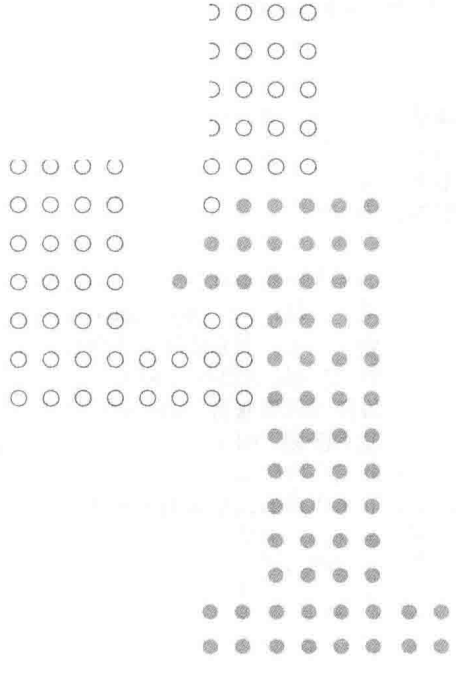
黄　艳　等 编著

清华大学出版社

计算机系列教材

黄　艳　等 编著

C#程序设计基础教程



清华大学出版社
北京

内 容 简 介

本书以 Visual C# 2013 为平台，紧跟 C#发展动向，介绍 C#程序设计各个方面的知识，内容安排兼顾广度、深度，知识新颖、示例丰富，比较系统地讲述了使用 C#语言进行程序开发从入门到实战应该掌握的各项技术。

全书共分为 10 章，内容包括 C#语言概述、C#程序设计基础、面向对象编程基础、面向对象高级编程、集合与泛型、Windows 窗体应用程序设计、高级窗体控件、C#文件与注册表操作、ADO.NET 数据库访问、网络编程。本书配备了大量示例，所有示例围绕一个实战项目，融知识性、趣味性于一体，逐层深入，循序渐进地介绍各个知识点。

本书可作为各类高等院校计算机及相关专业“C#程序设计”课程的教学用书，也可作为计算机应用人员和计算机爱好者的参考用书。

图书在版编目（CIP）数据

C#程序设计基础教程 /黄艳等编著. —北京：清华大学出版社，2015
计算机系列教材
ISBN 978-7-302-40823-9

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

中国版本图书馆 CIP 数据核字（2015）第 163197 号

责任编辑：白立军　薛　阳
封面设计：常雪影
责任校对：梁　毅
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**23.25　　**字　　数：**581 千字
版　　次：2015 年 9 月第 1 版　　**印　　次：**2015 年 9 月第 1 次印刷
印　　数：1～2000
定　　价：39.50 元

产品编号：060862-01

C#是一种安全的、稳定的、简单的、优雅的，由 C 和 C++衍生出来的面向对象的编程语言，也是微软.NET 公共语言运行环境中内置的核心程序设计语言。使用 C#语言可以开发在.NET Framework 上运行的多种应用程序，包括控制台应用程序、Windows 窗体应用程序、Web 应用程序以及 Web 服务等。C#集中了目前几乎所有关于软件开发和软件工程研究的最新成果，包括类型安全、面向对象、组件技术、内存自动管理、版本控制、代码安全管理等，为在.NET 环境下的计算机应用提供了功能强大、全新、易用的程序设计工具。

Visual Studio.NET 是微软公司推出的集成开发环境，是目前最流行的 Windows 平台应用程序开发环境，主要用来创建、运行、调试由各种.NET 编程语言编写的程序（包括 C#.NET）。Visual Studio 2013 是微软发布的 Visual Studio.NET 新版本，支持开发面向 Windows 8 的应用程序，其方便快捷的功能提供和简单明了的界面设计受到广大程序员的喜爱。

本书从教学实际需求出发，合理安排知识结构，从零开始、由浅入深、循序渐进地讲解了 C#语言的基本知识和 Visual Studio 2013 集成开发环境的使用方法。全书共分为 10 章，主要内容如下。

第 1 章简单地介绍了 C#语言的开发平台.NET 和开发环境 Visual Studio 2013，包括.NET Framework 的发展历程、Visual Studio 2013 的简单使用以及如何创建简单的 C#应用程序等。

第 2 章介绍 C#程序设计基础知识，包括 C#数据类型、流程控制语句等。

第 3 章讲述面向对象语言的编程基础，包括 C#中类的声明和对象的创建，C#常用类等。

第 4 章讲述面向对象语言高级编程知识，包括 C#中类的继承与派生、多态性的应用等。

第 5 章讲述集合和泛型，包括 ArrayList 集合类、Hashtable 集合类和泛型的使用等。

第 6 章主要介绍基于 C#的 Windows 窗体应用程序设计，包括窗体和基本控件的使用。

第 7 章介绍 C#高级窗体控件，包括菜单、工具栏、状态栏、列表视图、树视图和通用对话框等控件的使用。

第 8 章介绍 C#文件操作，包括文件和流的概念，读写文件的类，以及对文件和文件夹的多种操作方法等。

第 9 章讲述 ADO.NET 数据库访问技术，包括使用 ADO.NET 连接到这些数据源，并可以检索、处理和更新其中包含的数据等。

第 10 章讲述网络编程技术，包括用 C#进行各类网络应用程序的编程方法和技巧等。

本书以一个具体的综合实例为脉络，由浅入深地介绍面向对象编程思想，并在讲解每个面向对象知识点时由浅入深地逐渐丰富实例；书中内容条理清晰，图文并茂，通俗易懂；在难以理解和掌握的部分内容上给出相关注意事项，读者能够快速提高操作技能；此外，

本书配有章节练习，读者在解答时能更加牢固地掌握书中讲解的重点内容。

本书是集体智慧的结晶，参加本书编写和制作的人员有郑州轻工业学院软件学院的黄艳、郑倩、孙海燕、张玲、张志锋，郑州轻工业学院计算机与通信工程学院的朱会东和郑州轻工业学院物理与电子工程学院的陈鹏。感谢郑州轻工业学院教务处的大力支持和帮助！感谢清华大学出版社为本书出版所做的贡献！由于作者水平有限，书中不足之处在所难免，欢迎广大读者批评指正。

作　者

2015 年 3 月

FOREWORD

第1章 概 述

C#是微软公司推出的一种面向.NET 平台的、类型安全的面向对象编程语言，利用 C#语言和基于.NET 框架的 Visual Studio 2013 集成开发环境，程序员可以方便快捷地开发出各种安全可靠的应用程序。本章将对.NET 平台的相关内容做简单介绍，并通过图文并茂的方式介绍 Visual Studio 2013 集成开发环境及创建两种类型的 C#应用程序的操作步骤。通过本章的学习，读者将会对 C#语言和 Visual Studio 2013 集成开发环境有一个初步了解，并能够顺利地创建简单的 C#应用程序。

1.1 .NET Framework 概述

2000 年 6 月 22 日，微软公司正式对外宣布了.NET 战略。同年 11 月，微软在 COMDEX 计算机大展上发表了 Visual Studio.NET 软件，全面推进.NET 技术向市场进军的步伐。C#语言是微软公司针对.NET 平台推出的主流语言,它不但继承了 C++、Java 等面向对象语言的强大功能特性，同时还是继承了 VB、Delphi 等编程语言的可视化快速开发功能，是当前第一个完全面向组件的语言。作为.NET 平台的第一语言，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。本节主要介绍与 C#语言密切相关的.NET 平台和 Visual Studio 2013 集成开发环境。

1.1.1 .NET 平台简介

微软总裁兼首席执行官 Steve Ballmer 给.NET 下的定义为“.NET 代表一个集合，一个环境，一个可以作为平台支持下一代 Internet 的可编程结构。”即.NET = 新平台 + 标准协议 + 统一开发工具。作为微软的集成开发平台，.NET 技术提供迅速修改、部署、处理并且使用连接的能力，提高了 Web 服务的高效性；同时，.NET 技术也使创建稳定、可靠而又安全的 Windows 桌面应用程序更为容易。下面来简单了解一下.NET 的发展历程。

1. .NET 平台的发展历程

2000 年 6 月 22 日，比尔·盖茨向全球宣布其下一代软件和服务，即 Microsoft .NET 平台的构想和实施步骤。2000 年，微软的白皮书这样定义.NET：Microsoft .NET 是 Microsoft XML Web Services 平台。XML Web Services 允许应用程序通过 Internet 进行通信和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web Services 并将这些服务集成在一起之所需。

2002 年 2 月 13 日，微软正式发布了 Visual Studio .NET 2002，其中包含.NET Framework 1.0，除了引入一门全新的语言 C#之外，同时提供了对于 Java 的支持。C#大量借鉴了 Java 的语法，同时保留了 VB 方面的诸多便利性。ASP.NET 作为平台的关键组成部分，传承了微软一直以来的可视化设计风格，允许开发人员以拖放方式开发 Web 应用。然而.NET 1.0

作为全新的平台，许多类库仍然还不成熟。

2003年4月25日，曾被命名为Windows .NET Server的操作系统Windows Server 2003正式发布，同日还发布了Visual Studio .NET 2003，并将.NET Framework的版本升级到了1.1.4322。Windows Server 2003是微软发展史上一个非常重要的里程碑：一方面Windows操作系统在企业级应用方面的能力得到证实，另一方面.NET终于完成了和Windows操作系统的无缝集成，也真正意义上为开发人员提供了一套完整的.NET解决方案。

Visual Studio .NET 2003为程序开发人员提供了统一的开发语言和开发界面，不管开发桌面应用，还是Web应用，或者是手机设备的应用，Visual Studio .NET使开发人员能够在不同应用开发中自由切换。同时，随着.NET Framework的稳定，微软内部越来越多的产品采用.NET重新开发，或者提供了和.NET的无缝对接。例如，2003年发布的Exchange 2003、Office 2003以及2004年发布的Biztalk Server 2004，都允许开发人员使用.NET开发应用，并且做到了无缝集成。

2005年10月27日，微软将Visual Studio .NET重新命名为Visual Studio 2005，同时将.NET Framework的版本升级到2.0。另外，为了方便开发人员，内置了一个用于开发调试的Web服务器，使得开发人员在开发过程可以更加方便地测试与部署。同日发布的SQL Server 2005完全架构在.NET之上，并允许开发人员使用.NET编写存储过程、函数及用户自定义类型（UDT）。

2006年，微软将WPF（Windows Presentation Foundation）、WCF（Windows Communication Foundation）、WWF（Windows Workflow Foundation）和Windows Cardspace整合成代号为“WinFX”的.NET Framework 3.0，并于2006年11月6日发布。.NET Framework 3.0的发布是对.NET Framework 2.0的一个重要补充，它弥补了微软在企业级开发的软肋。

2007年11月19日，微软发布了Visual Studio 2008，随同发布了.NET Framework 3.5。.NET Framework 3.5引入了LINQ和XLINQ技术，LINQ和XLINQ为开发人员带来了激动人心的编程体验。开发人员可以混合对象与数据，然后用同样的查询方式进行数据处理，更重要的是允许开发人员在任意环节进行扩展，从而帮助开发人员以一致的方式进行数据处理。

2010年4月12日，微软发布了Visual Studio 2010以及.NET Framework 4.0。.NET Framework 4.0包括更好的多核心支持、后台垃圾回收和服务器上的探查器附加，增加了新的内存映射文件和数字类型，并支持新的动态数据功能，包括新的查询筛选器、实体模板、对Entity Framework 4的更丰富的支持以及可轻松应用于现有Web窗体的验证和模板化功能等。

2012年8月16日，微软发布了最新版本Visual Studio 2012以及.NET Framework 4.5。.NET Framework 4.5是一个针对.NET Framework 4.0的高度兼容的就地更新。.NET Framework 4.5包括针对C#、Visual Basic和F#的重大语言和框架改进（以便程序员能够更轻松地编写异步代码）、同步代码中的控制流混合、可响应UI和Web应用程序可扩展性，并提供比.NET Framework 4.0更高的性能、可靠性和安全性。

2013年的11月13日，微软举办了Visual Studio 2013全球发布会。Visual Studio 2013不仅支持Windows 8.1 App开发，还新增了很多提高开发人员工作效率的新功能，例如，自动补全方括号、使用快捷键移动整行或整块的代码等；完美支持Windows 8.1的程序开

发；提高了 Web 网站开发的工作效率和灵活性；改进了调试和优化的工具；对用户界面进行了许多方面的改进，使得其有着更好的用户体验。

2014 年的 11 月 13 日，微软发布了全新的 Visual Studio 2013 Community Edition（Visual Studio 2013 社区版），并且宣布将免费提供。不过该版本有个明显的限制，那就是不能用于企业应用程序的开发。Visual Studio Community 2013 是微软 VS 家族的最新成员，也是专门为学生、开源贡献者、小企业、初创企业，以及独立开发者们设计的一个虽然免费、但功能齐全的开发环境。此外，该版本还包括用于创建非企业跨平台（桌面、设备、云、Web、服务等）应用程序所需的所有特性，如编码效率、跨平台移动开发工具（Windows、iOS 和 Android），以及对成千上万的扩展的完整访问。

2．.NET 平台的组成

众所周知，微软的灵魂产品 Windows 操作系统是硬件设备和软件运行环境的平台，它消除了不同硬件设备之间的差别，使外部设备都变成了可以自由使用、无缝集成的一个整体。与 Windows 操作系统类似，微软推出的.NET 平台能够消除互连环境中不同硬件、软件、服务的差别，使不同的设备、不同的操作系统都可以相互通信，使不同的程序和服务之间都可以相互调用。

.NET 平台几乎包含微软正在研发或已经得到广泛应用的各种软件开发技术。对.NET 程序员来说，应主要关心.NET 平台的以下几个组成部分。

（1）.NET Framework：微软推出的一种运行于各个操作系统之上的新的软件运行平台，提供了.NET 程序运行时支持和功能强大的类库，是其他所有.NET 技术产品的坚实基础。

（2）.NET 编程语言：.NET 平台支持二十多种编程语言，传统的各种编程语言有许多都已经或正在被移植到.NET 平台，目前.NET 平台支持的编程语言种类仍在不断地增加中。目前由微软公司提供的.NET 编程语言主要有 Visual Basic .NET（改进过的 Visual Basic）、C++、C#、F#。

（3）Visual Studio .NET 集成开发环境：用来开发、测试和部署应用程序。Visual Studio .NET 历经微软公司持续多年的完善，已经成为世界一流的“软件集成开发环境（Integrated Development Environment,IDE）”。

（4）.NET 软件产品：几乎微软公司所有主要软件产品都基于.NET Framework 或包容.NET 技术，包括 Windows 操作系统、SQL Server 数据库服务器、Office 商业应用开发与运行平台、Azure 云计算平台等。

3．.NET 技术前景

从 2002 年发布.NET 1.0，历经 11 年的发展，.NET 版本已经发展到了 4.5。.NET 是一个庞大而复杂的软件开发与运行平台，包含一系列子技术领域。

1）桌面应用程序开发技术

在很长的一段时间内，Windows Form 成为.NET 桌面领域的主流技术，而且有一大批各式各样的第三方控件，其功能可谓应有尽有，使用方便。然而.NET 3.0 中出现的 WPF，在界面设计和用户体验上比 Windows Form 要强得多，比如其强大的数据绑定、动画、依赖属性和路由事件机制等。WPF 的性能在.NET 4.0 上有了进一步的改进。WPF 相对于 Windows 客户端的开发来说，向前跨出了巨大的一步，它提供了超丰富的.NET UI 框架，集成了矢量图形，丰富的流动文字支持，3D 视觉效果和强大无比的控件模型框架。

2）数据存取技术

.NET 平台融合了 ADO.NET、LINQ 和 WCF Data Service 等数据存取技术。ADO.NET 不仅提供了对 XML 的强大支持，还引入了一些新的对象，如驻于内存的数据缓冲区 DataSet、用来高效率读取数据并返回一个只读的记录集的 DataReader 等。LINQ（Language Integrated Query，语言集成查询）是 Visual Studio 2008 和.NET Framework 3.5 版中引入的一项创新功能，它在对象领域和数据领域之间架起了一座桥梁。LINQ 是编程语言的一个组成部分，在编写程序时可以得到很好的编译时语法检查、丰富的元数据、智能感知、静态类型等强类型语言的好处。同时，LINQ 还可以方便地对内存中的信息进行查询而不只是对外部数据源进行查询。WCF Data Service 原称 ADO.NET Data Service，体现了“数据是一种服务”的思想，让数据可以通过 HTTP 请求直接获取。WCF Data Service 设计了一套 URI 模式，可以完成投影、选择、分页等功能，用起来方便灵活。

3）Web 开发技术

.NET 平台底层使用 ADO.NET 实体框架或 LINQ to SQL 构造数据模型，通过提取数据模型中的元数据，动态选择合适的模板生成网页，避免了真实项目中不得不为每个数据存取任务设计不同网页的负担，而且提供了很多方式允许用户定制网站。.NET 平台的另一种 Web 应用架构代表技术 Silverlight 充分利用客户端的计算资源，极大地降低了对服务端的依赖，并且易于构造良好的用户体验。

4）插件技术

.NET 4.0 引入了 Managed Extensibility Framework（MEF）技术。MEF 通过简单地给代码附加“[Import]”和“[Export]”标记就可以清晰地表明组件之间的“服务消费”与“服务提供”关系，并在底层使用反射动态地完成组件识别、装配工作从而使得开发基于插件架构的应用系统变得简单。

5）函数式编程语言 F#

F#是微软.NET 平台上一门新兴的函数式编程语言，通过 F#，开发人员可以轻松应对多核多并发时代的并行计算和分布问题。

1.1.2 .NET Framework

.NET Framework 是.NET 平台的关键组件，提供了.NET 程序运行时支持和功能强大的类库。从开发各种应用软件的程序员角度来看，.NET Framework 用易于理解与使用的面向对象方式调用 Windows 操作系统所提供的各种系统功能。.NET Framework 在整个软件体系结构中的地位如图 1.1 所示。.NET Framework 在应用程序和操作系统之间起到承上启下的作用，向内包容着操作系统内核，向外给运行于其上的.NET 应用程序提供访问操作系统核心功能的服务。在.NET Framework 下编程，程序员不再需要与各种复杂的 Windows API 函数打交道，只需使用现成的.NET Framework 类库即可。

.NET Framework 的体系结构如图 1.2 所示，它主要由公共语言运行库（Common Language Runtime，CLR）和.NET Framework 类库所构成。CLR 是.NET Framework 的核心执行环境，也称为.NET 运行库。CLR 是一个技术规范，无论程序使用什么语言编写，只要能编译成微软中间语言（Microsoft Intermediate Language，MSIL），就可以在它的支持下运行。这意味着在不久的将来，可以在 Windows 环境下运行传统的非 Windows 语言。而.NET

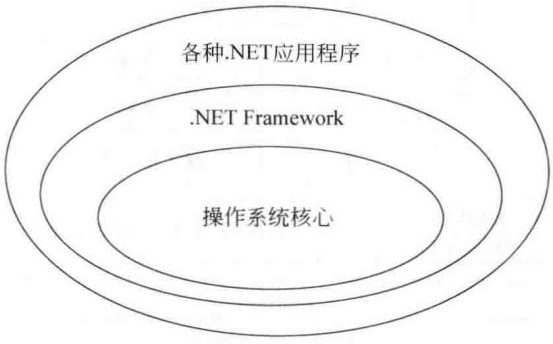


图 1.1 .NET 软件体系结构

Framework 类库是一个由 Microsoft .NET Framework SDK 中包含的类、接口和值类型组成的库，提供对系统功能的访问，是建立.NET Framework 应用程序、组件和控件的基础，该库可以完成以前要通过 Windows API 来完成的绝大多数任务。

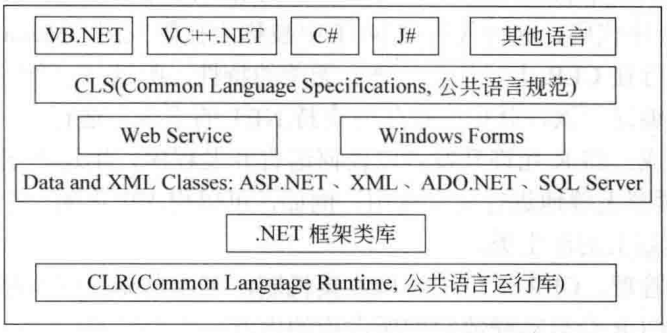


图 1.2 .NET Framework 的体系结构

1. 公共语言运行库

CLR 最早被称为下一代 Windows 服务运行时（NGWS Runtime），它是直接建立在操作系统上的一个虚拟环境，提供内存管理、线程管理和远程处理等核心服务，主要的任务是管理代码的运行。CLR 支持几十种现代的编程语言，在应用程序运行之前，CLR 使用 Just-In-Time 编译器把已经编译为 MSIL 的不同编程语言程序代码转换为本地可执行代码。

CLR 通过公共类型系统（Common Type System，CTS）和公共语言规范（Common Language Specification，CLS）定义标准数据类型和语言间的互操作性规则，实现跨语言开发和跨平台的战略目标。CTS 定义了如何在 CLR 中声明、使用和管理类型，使所有面向.NET Framework 的语言都可以生产最终基于这些类型的编译代码。任何以.NET 平台作为目标的语言必须建立其数据类型与 CTS 类型间的映射，以便通过共享 CLR 实现它们之间无缝的互操作。CLS 是 CLR 标识的一组语言特征的集合，所有.NET 语言都应该遵循此规则才能创建与其他语言可互操作的应用程序。

CLR 被认为是.NET 中编写的程序“管理器”，它能确保程序符合安全规则，并向程序提供资源，CLR 结构如图 1.3 所示。.NET Framework 所具有的许多特点都是由 CLR 提供的，如类型安全（Type Checker）、垃圾回收（Garbage Collector）、异常处理（Exception Manager）、向下兼容（COM Marshaler）等。