河南省高等学校计算机教育研究会统编教材

C#程序设计

刘克成 张凌晓 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

河南省高等学校计算机教育研究会统编教材

C#程序设计

刘克成　张凌晓　主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书以 Visual Studio 2005 为开发平台，介绍了使用 C#语言进行可视化开发 Windows 应用程序的方法和技术。全书共分 12 章，主要介绍了.NET 平台的基本知识、Visual C#.NET 集成开发环境和 C#语言程序设计语法基础以及如何使用 C#语言可视化开发各种应用程序的知识和方法。

本书既可作为高等学校可视化程序设计教材，又可作为计算机程序设计培训教材或其他从事计算机程序设计人员的参考书。

图书在版编目（CIP）数据

C#程序设计/刘克成，张凌晓主编. —北京：中国铁道出版社，2007.1

河南省高等学校计算机教育研究会统编教材

ISBN 978-7-113-07779-2

Ⅰ. C… Ⅱ. ①刘… ②张… Ⅲ. C 语言－程序设计－高等学校－教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 018002 号

书　　名：C#程序设计

作　　者：刘克成　张凌晓　等

出版发行：中国铁道出版社（100054，北京市宣武区右安门西街 8 号）

策划编辑：严晓舟　秦绪好

责任编辑：苏　茜　李晶璞　王慧亮

封面设计：薛　为

封面制作：白　雪

责任校对：王春霞

印　　刷：北京新魏印刷厂

开　　本：787×1092　1/16　**印张**：22　**字数**：510 千

版　　本：2007 年 3 月第 1 版　2007 年 3 月第 1 次印刷

书　　号：ISBN 978-7-113-07779-2/TP・2144

定　　价：29.00 元

河南省高等学校计算机教育研究会

教材编审委员会

主任委员：段银田

副主任委员：甘　勇　普杰信　王贺明

秘　书：李学相

委　员：段银田　甘　勇　普杰信　王贺明　李学相

翁　梅　曲宏山　郭清溥　申石磊　周清雷

刘克成　陆桂明　程万里　马占欣　陈　涛

张东升　朱国华　李　敏　黄贻彬　商信华

连卫民　杨立峰　商其坤

序

PREFACE

自上个世纪80年代初到本世纪初的20年间，由于计算机奇迹般地展示出它惊人的运算速度、海量的存储能力和神奇的创造性，使人类社会深深地感受到了计算机的存在和它的不可或缺性。在这种背景下，全国各类高等学校已陆续开展了计算机基础教育，普及了计算机文化基础知识和技术基础知识。相应地，这两种类型的教材也大量涌现，为计算机教育和应用的普及提供了丰富的智力资源。

然而，进入21世纪以来，高等学校的计算机基础教育面临着新的挑战。首先，一个时期以来，信息技术自身愈来愈向技术多元化的方向发展。网络、数据库、多媒体等技术已从科学的殿堂里走了出来，并日益得到应用和普及，各种信息技术在工程中的综合应用程度越来越高，这一切促使全社会计算机的应用水平提升到了一个新高度。应用的普及也推动了需求的进一步多样化，社会也因此更加迫切地需要实用型信息技术人才。在这种背景下，大学现行计算机基础教育教材已远远不能适应技术发展和应用的要求。其次，由于近年来中小学信息技术教育的普遍开展，使得原本在大学要完成的信息技术学习任务的一部分已经提前完成，因此也需要调整当前高等学校计算机应用基础教学的内容，可见更新现行教材已成为当前一项十分紧迫的任务。作为高等学校计算机基础教育教材改革创作的尝试，河南省高等学校计算机教育研究会与中国铁道出版社共同策划了这套系列教材。

本套教材的创作是以社会对信息技术的应用需求为目标，学习的方向应瞄准应用，学习的目的是能够做事。要知道，仅能在操作层面上使用计算机并不是真正意义上的应用，开发才是真正的应用，也就是常说的开发应用，这也就是大学生学习信息技术的方向和应采取的行动。这些观念应逐步成为教材创作的指导思想。

突出信息技术教育的目的性是本系列丛书内容的最大特色。信息技术何其多！究竟学什么、写什么？要改变那种无的放矢的、包罗万象的教材创作模式；要有目的地去写过程，摒弃那种遍历知识过程就是一切没有目的、文字堆砌式的创作观念和方法。应当明白，学习信息技术是为了做事情，而不是为了其他。此外，计算机基础教育的教材要提倡精简。要树立信息量观点，能够释疑解惑的文字构成信息量，可以写入教材，不能起到释疑解惑作用的文字或冗余文字只能形成垃圾信息，应当从教材中剔除出去。

例如，对于操作技能类的教材来说，完全可以按“展示一种目的，精讲一个案例，完成一个练习，创造一个作品”这四句话的要求来进行教材创作。对于程序设计类教材来说，教材应逐步体现并满足从程序设计向软件设计延伸的社会需求。

在教材创作中，应努力完成相关知识的整合，这不仅是本套教材所提倡的创作特色之

一，也是信息技术教育改革的出路所在。对于计算机基础教育来说，知识整合主要体现在两个方面。其一，用公用事件整合适用的信息技术。把面向社会大众所发生的信息技术应用事件用其所必须的信息技术，而不是某个领域的全部信息技术加以整合应用。把从目的到技术的逆向思维作为新一代信息技术教材创作的思维方法和行为方法。整个创作过程应按照“目的决定过程，过程决定事件，事件决定对象，对象决定技术”的思路进行。其二，信息技术与其他特定学科的相互整合。这种整合开辟了信息技术与专业相结合进行教材创作的途径。更加有利于实现从目的到技术进行教材创作的思想，使特定学科的内容和信息技术实现“我中有你，你中有我”，达到更高层次的融合。这种融合有利于双方共同提高教学效率，拓宽知识领域，增加知识深度，激发创造性思维。总之，本丛书的创作特色主要体现在用目的、事件、对象去整合适用的信息技术。帮助读者为了达到目的而学会利用信息技术做一些实实在在的事情。

最后，本人深知新一代计算机基础教育教材的创作远不是一蹴而就的事情，目标的实现尚需时日。序言的目的仅在于简要阐明本套教材在策划过程中提出的一些基本思想和对创作的原则要求，正确与否还须经过实践的检验。望作者和读者在创作与实践中不断斧正。

河南省高等学校计算机教育研究会理事长　段钦田

前言

FOREWORD

C#语言是Microsoft公司近年推出的一种基于.NET平台的新的面向对象编程语言，它功能强大，安全而灵活，具有清晰的面向对象的语法结构、优秀的编程开发环境和高效率的编译工具，它解决了存在于许多程序语言中的安全、垃圾收集、与其他语言交互的能力和跨平台的兼容性等问题。在Visual Studio 2005中使用C#语言，不仅能够让开发人员在.NET平台上快捷、方便地开发图形设计、图像处理、多媒体技术、网络技术和数据库技术等Windows应用程序，还可以进行组件编程、多线程编程和分布式编程。相对于C++而言，C#更容易被理解和接受，是一种理想的面向对象程序设计语言，赢得了广大程序员的喜爱。

本书以Visual Studio 2005为开发平台，重点介绍了C#语言和使用C#进行可视化开发Windows应用程序的方法和技术。全书共分12章，第1章介绍了.NET平台的基本知识和Visual C#.NET集成开发环境；第2章介绍了C#语言的语法基础；第3章对C#面向对象程序设计的基本思想及方法进行了介绍；第4章对Windows程序框架、常用Windows窗体控件和菜单编程进行了阐述，并通过实例介绍了进行Windows应用程序界面设计的方法；第5章通过实例介绍了对话框的使用方法和多文档编程思想；第6章讨论了文件操作所涉及的知识和技术；第7章介绍了使用Visual C#进行图形设计和图像处理方面的技术及应用实例；第8章对如何进行进程与多线程的编程作了介绍；第9章通过实例分析了C#网络应用的套接字和域名服务知识；第10章介绍了数据库应用程序的开发过程和方法；第11章对C#组件编程进行了阐述；第12章介绍了.NET Web的应用。

本书是作者在近年来从事C#可视化程序设计的教学实践基础上，以讲义为基础编写而成的，其内容充实，选材上注意系统性、实用性，所提供的程序实例的设计思路、程序代码、技术要点等内容简明易读，所有程序均在Visual Studio 2005环境下调试通过。本书既可作为高等学校可视化程序设计语言课程教科书，又可作为工程技术人员参考用书。

本书由刘克成、张凌晓担任主编。第2、9章由刘克成编写；第3、6章由张凌晓编写；第4、5章由袁东锋编写；第7章由杨彩霞编写；第8章由闫朝华编写；第12章由黄宪通编写；第1章由杨新锋编写；第10章由孙晓莹编写；第11章由杨艳燕编写。刘克成、张凌晓对全书的程序进行了调试，全书总纂工作由刘克成、张凌晓负责完成。

在本书的编写过程中，承蒙河南省高等学校计算机教育研究会和中国铁道出版社的热情支持与指导，在此表示衷心地感谢，同时，又参阅了大量的网上资源和其他参考文献，在此对它们的作者和提供者一并表示感谢。

由于计算机科学技术发展迅速，程序设计的教学内容、方法和手段日新月异，加之编者水平有限，书中难免有不足之处，敬请读者批评指正，以便再版时修改完善。

编　者

2006年10月

目录

CONTENTS

第 1 章 C#概述

C#语言是 Microsoft 公司为推行.NET 战略而发布的一种全新的面向对象编程语言，它具有清晰的面向对象语法结构、优秀的编程开发环境和高效的编译工具，能快捷、方便地开发 Windows 应用程序。本章主要介绍.NET 平台、简单认识 C#语言和 Visual C#.NET 集成开发环境。

1.1 Microsoft.NET 平台概述

2000 年 6 月 22 日，Microsoft 公司正式推出了其下一代计算计划——Microsoft.NET，该平台为开发商进行网络计算应用提供了两个技术支持，一个是 Microsoft.NET 虚拟机，也就是说在该平台下，开发的程序源代码首先被编译成与处理器无关的中间语言（Microsoft Intermediate Language，MSIL）代码，当程序运行时，利用即时编译器（Just-in-Time，简称 JIT）把中间语言代码编译成特定 CPU 和操作系统的本机机器语言代码进行运行，这就实现了程序跨平台的良好移植性；二是开发出了面向对象的程序开发语言——C#，该语言是专门为.NET 应用而开发的。

那么，什么是 Microsoft.NET 平台？首先它是一个开发平台，它定义了一种公用语言子集（Common Language Subset，CLS），这是一种为符合其规范的语言与类库之间提供无缝集成的混合语言；其次，Microsoft.NET 统一了编程类库，提供了对下一代网络通信标准 XML（Extensible Markup Language）的完全支持；再者，Microsoft.NET 还实现了人机交互方面的革命，如在软件中添加手写和语言识别等功能，增加了对各种用户终端的支持能力。总之，Microsoft.NET 是一种面向网络支持各种用户终端的开发平台环境。

Microsoft.NET 平台主要由 5 个部分组成，分别是 Windows.NET、.NET Enterprise Servers（.NET 企业级服务器）、.NET Framework（.NET 框架）、Microsoft Visual Studio.NET 和.NET Web 服务构件。

1. Windows.NET

Windows.NET 是 Microsoft.NET 的基础平台，主要是一些在手机、微机和服务器群集上的操作系统（如 Windows XP/2000/CE）。另外，还包括各种应用软件服务，如 IIS（因特网信息服务系统）、ASP+（活动服务页面）、Active Directory（活动目录服务）、MSMQ（Microsoft 公司报文队列服务）、COM+/MTS（ActiveX 数据对象＋/Microsoft 公司交易服务系统）和 Distributed Transaction Coordinator（分布式事务处理协调器）等。

2. .NET Enterprise Servers

.NET Enterprise Servers 是 Microsoft 公司推出的进行企业集成和管理所有基于 Web 的各种服务器应用的系列产品，包括 Application Center 2005、BizTalk Server 2005、Commerce Server 2005、SQL Server 2005、Exchange Server 2005 等。它的主要目标是帮助企业快速集成并构架电子商务及其应用系统。

3. .NET Framework

.NET Framework 是 Microsoft.NET 的核心部分，它提供了建立和运行.NET 应用程序所需要的编辑、编译等核心服务。

.NET Framework 主要由两部分组成：一是公共语言运行时（Common Language Runtime，CLR）；二是.NET 的基础类库（Basic Class Library，BCL）。.NET Framework 框架如图 1-1 所示。

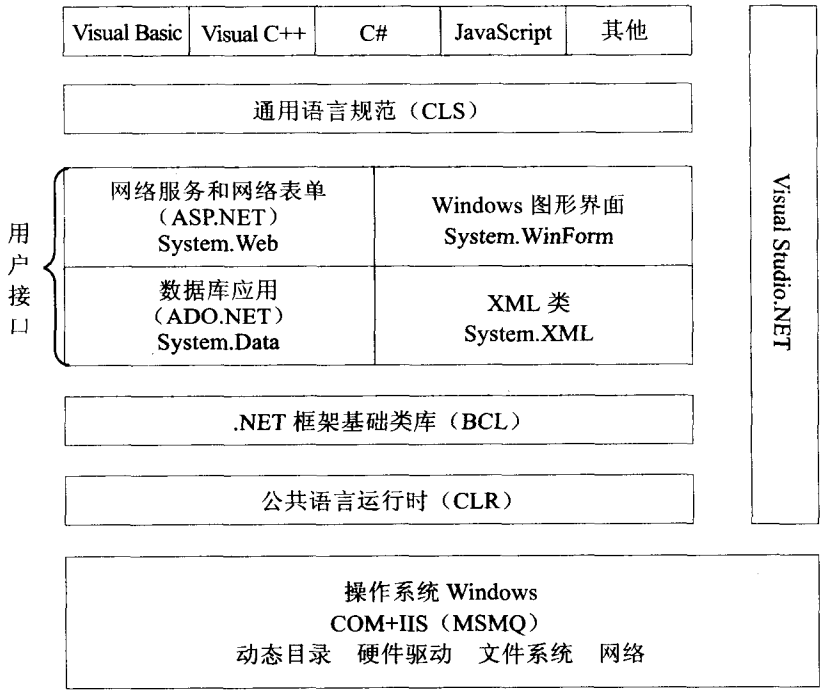

图 1-1 .NET Framework 框架

.NET Framework 框架是一组用于建立 Web 服务器应用程序和 Windows 桌面应用程序的软件组件，在该平台下开发的应用程序是在 CLR 的控制下运行的。在开发技术方面，.NET Framework 提供了 ADO.NET（数据库访问技术）、ASP.NET（网络应用开发技术）和 WinForm（Windows 编程技术）；在语言方面，.NET Framework 提供了 Visual Basic、Visual C++、C#、JavaScript 等多种语言；而 Visual Studio.NET 则全面支持.NET 的开发工具。

（1）CLR（公共语言运行时）

CLR 是.NET Framework 的核心，管理着.NET 代码的执行，也就是说，它是一个软件引擎，用于加载应用程序、检查错误、进行安全许可认证、执行和清空内存等。所谓的运行时（Runtime Environment，Runtime）是指那些支持在特定的平台上，用于运行特定编程语言的软件的库和程序集，它一般要处理软件和操作系统之间的接口细节（如系统调用、内存管理等）。运行时分为纯静态环境（如 Fortran）、基于堆栈环境（如 C、C++）和纯动态环境（如 Java）三种。CLR 是属于纯动态运行时的一种。总之，CLR 提供了一个可靠而完善的多语言运行环境，简化了应用程序的开发配置和管理，从而实现组件能够在多语言环境下跨平台工作。

① CLR 由以下 12 个功能器件组成。

- 类加载器（Class Loader）：将程序的汇编加载到内存中，包括 MSIL 代码、元数据和其他应用程序所需的组件。
- 即时编译器（JIT）：将 MSIL 翻译成本地执行代码。
- 代码管理器（Code Manager）：管理代码的执行。
- 垃圾回收器（Garbage Collection）：负责整个.NET 运行时托管代码的内存分配与释放，并进行自动的垃圾收集。
- 安全引擎（Security Engine）：提供基于认证的安全机制。
- 调试引擎（Debugger）：使程序开发者进行调试和跟踪应用程序代码。
- 类检查器（Type Checker）：检查并禁止非安全的类型转换及未初始化的变量的使用。
- 异常管理器（Exception Manager）：提供结构化的异常处理。
- 线程支持（Thread Support）：提供了多线程编程的类和接口。
- COM 汇集器（COM Marshaler）：处理与 COM 之间的配置。
- .NET 基础类库（BCL）：集成具有支持.NET Framework 类库运行时的代码。
- 公共类系统（CTS）：为了实现语言的互用性，.NET Framework 采用两种方法来解决语言的划分问题。一是标准化数据类型。建立通用语言运行环境中的公共类型系统（Common Type System，CTS），它为最常用的数据类型（如整数）定义了标准的内部描述和运算，并提供了将这些类型向所有的.NET 语言和 CLR 扩展的机制。这种机制能够表示绝大多数现代编程语言的语法，消除了每种语言自己唯一且不兼容的方法。CTS 是一套 CLR 中的数据类型都必须遵守的规则，如果某种语言在创建数据类型时遵守了 CTS，则它创建和存储的数据将能够与其他遵守 CTS 的编程语言互相兼容。二是标准化应用程序格式。.NET 有自己的 MSIL、元数据和清单的汇编，所有.NET 语言的编译器都生成这种格式，即通过从元数据中提取有关的 MSIL 的信息，编译器、调试器和协调器等工具可以分析处理任何一种源程序设计语言的数据。

② CLR 工作原理

选择一种语言编写源代码，然后用相应的.NET 编译器（如 C#编译器、Visual Basic 编译器）编译，当编译完成后，所产生的是 PE（Portable Executable）格式的文件，在 PE 文件中还包括两部分：MSIL 代码和元数据，然后形成扩展名为 exe 或 dll 的应用程序的汇编文件，该文件包含有 MSIL、详细描述 MSIL 代码正确执行所需的各种相关数据类型的元数据以及一个列出了所有文件和软件组件的文档清单。这种格式的文件不是可以在操作平台上执行的程序代码，还不能运行。

接下来加载一个应用程序，CLR 使用汇编的清单确定应用程序所需汇编的正确版本，然后 CLR 检查应用程序的全部汇编——MSIL 代码和描述它的元数据，从而确认代码是“类型安全”的。接下来 CLR 加载应用程序的汇编中的 MSIL，并在此过程中，收集有关.NET Framework 的安全要素，判断是否运行应用程序，以及运行时需要什么许可。例如，汇编是从哪儿下载或安装的、需要执行什么功能、什么用户要运行它、汇编是否拥有可信任的数字签名，以及进行数字签名后汇编是否有改动等。

当需要运行该程序时，CLR 的 JIT 再把相应的 MSIL 翻译成可执行代码。因此，无论是何种语言编写的.NET 程序实际上都是以本地代码运行的。

CLR 可以监控翻译代码的运行，并且使用一个称为“碎片整理”的进程定期清理应用程序释放的内存。

③ 元数据（Metadata）

元数据包括对 CLR 保存的每个元素的描述信息。这些元素包括汇编、类型、方法、属性、执行引擎以及另一个组件的调用数据等。运行时利用这些信息进行操作，如调用和垃圾回收。

当编译.NET 程序时，元数据由编译器自动生成，并与 MSIL 一起位于同一个 PE 文件中。

④ 托管与非托管代码

.NET 的托管代码（Managed Code）是指符合 CLR 运行规范，能受控于 CLR 下的内存管理、线程管理、远程管理、代码强制安全类型的代码。非托管代码（Unmanaged Code）指对内存、文件、数据库等非托管资源进行操作的代码，它们通常不受控于 CLR 的代码管理规范，是不安全代码。对托管代码的编程只需专注于编程的逻辑，对于内存等资源的管理由 CLR 负责；而对非托管代码的编程，则要求开发人员负责各种非托管资源的分配和回收工作。默认情况下，所有的 C#代码都是托管代码。

在 CLR 中使用托管代码和非托管代码表现出程序在 CLR 下运行时的受限制程度。托管的任何对象都受到 CLR 的严格控制，受到的限制有编译器必须生成面向 CLR 的 MSIL 文件，必须使用.NET Framework 库函数；而非托管代码不受 CLR 控制，本身也不必向执行引擎提供元数据信息；反之，执行引擎也不给非托管代码提供各方面的安全服务，如 COM 组件和 Win32 API 函数等。

由 CLR 在垃圾回收器自动回收的堆上分配和释放的数据是托管数据，它们只能被托管代码访问，而且托管代码既能访问托管数据也能访问非托管数据。

⑤ 公共语言规范（Common Language Specification，CLS）

CLS 描述了不同的编程语言都应具有的特征集（它是 CTS 的一个子集），所有的.NET 语言都支持它。这样任何使用与 CLS 兼容的类型的程序，都可以和以任何语言编写的.NET 程序进行互操作。

（2）BCL（基础类库）

BCL 提供了几乎所有应用程序都需要的公共代码。.NET Framework 的 BCL 为开发者提供了面向对象的特性所需的和 CLR 紧密集成的一组可重用类的集合，并且此类库仅使用 CTS 数据类型和标准应用程序格式编制，简化了应用程序开发过程和难度，很容易与第三方组件无缝集成，也能被使用任何一种.NET 编程语言的应用程序所使用。

BCL 中有几百个类，它包括从输入/输出到数据访问等方面，提供了一个统一的面向对象的、层次化的可扩展编程接口，这些类以命名空间（namespace）的分级制度划分其功能，并使用点分隔各层次。例如，System.Collections.ArrayList，说明 ArrayList 类属于 System.Collections 命名空间。Collections 命名空间包含了链表、哈希表等集合类型，可用于控制对象间的连接。System 是基础类库的根。

BCL 是.NET 语言共享的标准类库，任何遵循.NET 的语言都可以调用它。使用时，开发者只需在自己的应用程序中添加所需的基础类库的引用，就可以使用这个类库中的所有方法。