清华 电脑学堂

DVD

超值多媒体光盘
长达300分钟的教学视频
全书实例完整源代码
附赠QAST授权网络和应用软件

- ✓ 总结了作者长期教学培训成果，难易适中，实用性强
- ✓ 系统全面介绍C# 2008技术要点
- ✓ 围绕丰富实例讲解C#编程实践知识
- ✓ 精心编写大量“实验指导”，引导读者深入学习编程实践
- ✓ 课后提供丰富习题，巩固学习成果
- ✓ 网站提供代码下载和课件支持

■ 李乃文 傅游 沈学利 等编著

C# 2008程序设计

实践教程

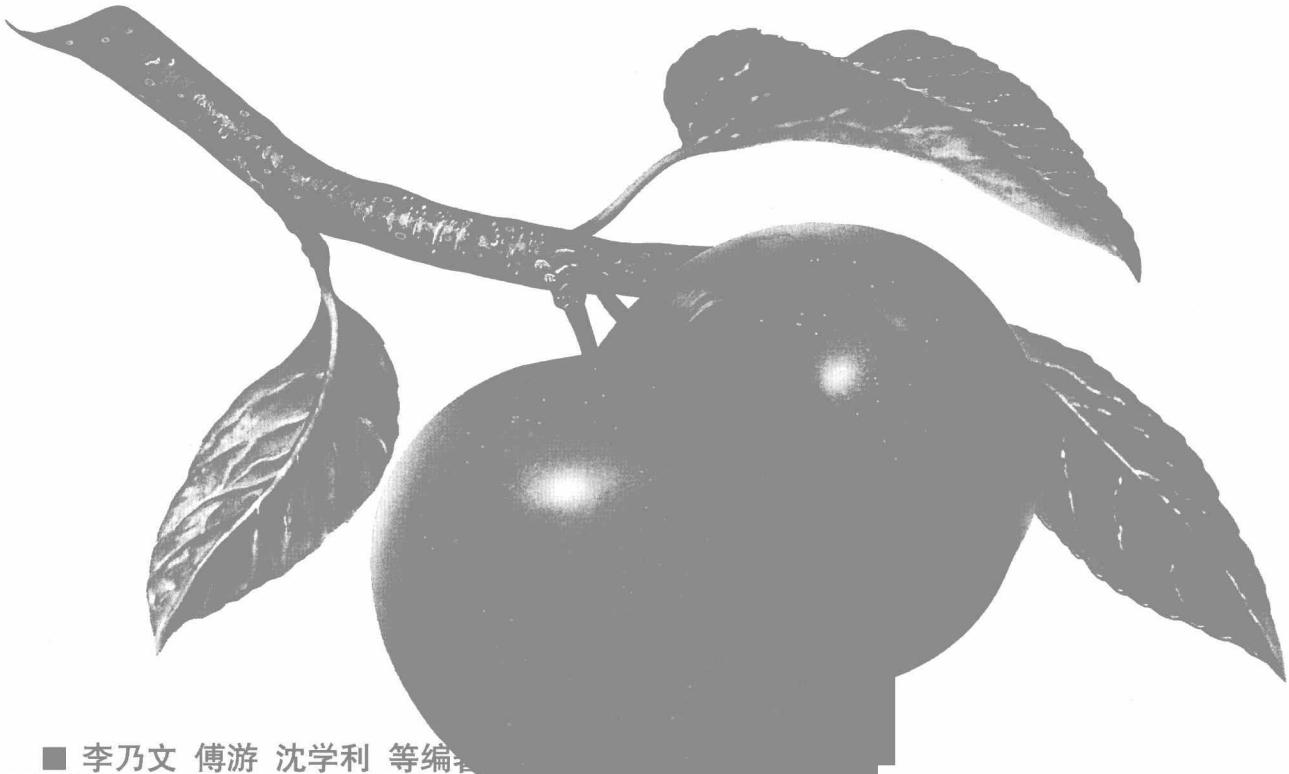
清华大学出版社

清華 电脑学堂

DVD

超值多媒体光盘
长达300分钟的教学视频
全书实例完整源代码
附赠QAST授权网络和应用软件

- ✓ 总结了作者长期教学培训成果，难易适中，实用性强
- ✓ 系统全面介绍C# 2008技术要点
- ✓ 围绕丰富实例讲解C#编程实践知识
- ✓ 精心编写大量“实验指导”，引导读者深入学习编程实践
- ✓ 课后提供丰富习题，巩固学习成果
- ✓ 网站提供代码下载和课件支持



■ 李乃文 傅游 沈学利 等编著

C# 2008程序设计

实践教程

清华大学出版社
北　京

内 容 简 介

本书介绍最新版本的.NET 3.5 和 Visual C# 2008 程序开发知识。全书共分 14 章，内容包括.NET Framework 3.5 简介，C#编程基础知识，C#对象和类型，面向对象中的两个重要特性：继承和多态，数组知识，类型转换的使用，C#的字符串和正则表达式，集合的使用，try/catch 块、throw 子句、异常涉及的类以及如何创建用户自定义异常等，开发 Windows 窗体应用程序时所需的各种控件，C#中如何创建多文档界面(MDI)应用程序，C#的 ADO.NET 数据库编程，数据库的高级编程知识，在.NET Framework 上的其他应用程序的开发，包括 LINQ、WF、WCF 以及 Office 等。

本书内容丰富，实践性强，面向所有 C#程序设计人员，可作为普通高等院校 C#程序设计课程的教材，也可作为学习和使用.NET 和 C#编程的开发人员的参考资料。

图书在版编目（CIP）数据

C# 2008 程序设计实践教程 / 李乃文等编著. —北京：清华大学出版社，2009.5
ISBN 978-7-302-19651-8

Ⅰ.C…　Ⅱ. 李…　Ⅲ. C 语言－程序设计－教材　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 030564 号

责任编辑：夏兆彦
责任校对：徐俊伟
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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：北京市清华园胶印厂
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：185×260　**印　张**：23.25　**字　数**：578 千字
附光盘 1 张
版　　次：2009 年 5 月第 1 版　　**印　　次**：2009 年 5 月第1 次印刷
印　　数：1～5000
定　　价：39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　　产品编号：031603-01

前言

FOREWORD

C#是随.NET Framework 一起发布的一种新语言，是一种崭新的面向对象的编程语言，强调以组件为基础的软件开发方法。它不但结合了 Visual Basic 的简单易用性，同时也提供了 Java 和 C++语言的灵活性和强大功能。C#在.NET Framework 构架中扮演着一个重要角色，可以说它是 Microsoft 公司面向下一代互联网软件和服务战略的重要内容，也是编写.NET Framework 应用程序的首选。

1. 本书主要内容

本书以最新版本的.NET 3.5 和 Visual C# 2008 为例进行介绍，注重从初学者的认识规律出发，强调实用性、可操作性。使用通俗语言对 Visual C# 2008 基本概念和基本设计方法的讲解浅显易懂、深入浅出，并且安排了大量典型实用的例题，使学习者结合实例学习、掌握设计的方法和技巧。全书各章概要如下。

第 1 章介绍.NET Framework 3.5，包括.NET Framework 3.5 简介、.NET Framework 平台结构及其重要组成部分，还介绍了如何配置.NET Framework 环境及使用 C#创建各类型的.NET 应用程序。

第 2 章通过对 C#中的变量、数据类型、运算符及控制语句进行讲解，使读者掌握 C#编程的基础知识。

第 3 章介绍 C#对象和类型，包括定义类和结构、为类添加成员、类构造函数、结构的继承和构造函数及这些类型的基类 Object 类等。

第 4 章介绍面向对象中的两个重要特性：继承和多态，包括继承的类型、如何实现继承、继承时的构造函数、多态的虚方法和隐藏方法以及自定义接口等。

第 5 章主要介绍数组，包括数组的声明、数组初始化、访问数组元素、多维和锯齿数组、数组的排序以及常见接口的实现，如 IComparable、ICollection 和 IList 等。

第 6 章介绍类型转换的使用，包括类型的安全性、装箱和拆箱、对象比较、运算符重载以及用户如何自定义数据类型转换。

第 7 章介绍 C#的字符串和正则表达式，包括 String 类、StringBuilder、格式化字符串、正则表达式概述、使用正则来匹配、组合和捕获等。

第 8 章介绍集合的使用，包括常用的集合类以及各种集合类型，如列表、队列、栈、链表、有序表、字典和位数组等。

第 9 章介绍有关 try/catch 块、throw 子句、异常涉及的类、finally 块以及如何创建用户自定义异常等方面的知识。

第 10 章主要介绍开发 Windows 窗体应用程序时所需的各种控件，包括基本控件、显示控件、图形和图像控件、按钮类控件、列表类控件、容器类控件、Timer 和 NotifyIcon 控件等。

第 11 章介绍 C#中如何创建多文档界面（MDI）应用程序，以及 MDI 应用程序中需要用的工具栏和状态栏，在父窗体和子窗体之间集成菜单和其他控件。

第 12 章介绍 C#的 ADO.NET 数据库编程，包括：ADO.NET 的概述、使用 ADO.NET 定义数据库连接、DataAdapter、DataReader、CommandBuilder 以及 DataSet 的应用等。

第 13 章介绍数据库的高级编程知识，包括使用 DataGridView 控件、简单绑定数据、复杂绑定以及使用 Crystal Report 报表等。

第 14 章介绍在.NET Framework 上的其他应用程序开发，包括 LINQ、WF、WCF 以及 Office 等。

2. 本书主要特色

本书通过实例介绍 Visual C# 2008 程序开发知识，具有实用性教程的特色。

- 本书汇总了作者多年的程序员职业教学培训经验，内容组织更合理，实例丰富全面。
- 本书使用 Visual C# 2008 语言开发了大量实例，读者可以通过这些丰富实例学习 Visual C# 2008 编程实践知识。
- 本书编写了大量“实验项目”，引导读者应用该章知识独立练习编程项目。
- 每章课后练习题帮助学生检查对 Visual C# 2008 开发理论知识的掌握程度。
- 本书光盘提供了实例的完整源文件和教学视频文件。

提示：本光盘使用之前需要首先安装光盘中提供的 tscc 插件才能运行视频文件。

3. 本书读者对象

书中采用大量的实例进行讲解，力求通过实例使读者更形象地理解面向对象思想，快速掌握 C#编程技术。本书难度适中，内容由浅入深，实用性强。每章附有精心编写的实验和习题，便于读者实践和巩固所学知识。本书内容丰富、实践性强，面向所有 C#程序设计人员，本书可作为普通高等院校 C#程序设计课程的教材，也可作为对.NET 感兴趣的读者的参考资料。

参与本书编写的除了封面署名人员外，还有胡丽霞、铙美君、肖新峰、宋强、马海军、许勇光、王泽波、孙江玮、田成军、刘俊杰、李海庆、王树兴、朱俊成、王敏、张瑞萍、王黎、安征、亢凤林、康显丽、李海峰、崔群法、孙岩、祁凯、倪宝童、王立新、吴越胜、何方、张银鹤等。

由于时间仓促，水平有限，疏漏之处在所难免，欢迎读者登录清华大学出版社的网站 www.tup.com.cn 与我们联系，帮助我们改进提高。

目 录

CONTENTS

第 1 章　.NET Framework 体系结构

内容摘要 | Abstract

Microsoft 发布的.NET Framework 简称.NET，是支持生成和运行下一代应用程序和 Web 服务的内部 Windows 组件，它提供了托管执行环境、简化的开发和部署以及与各种编程语言的集成。本章以最新的.NET Framework 3.5 版本为例，向读者介绍.NET Framework 及其重要组成部分、.NET Framework 3.5 的新增特性，如何配置.NET Framework 的开发环境及使用 C#创建.NET 应用程序。

学习目标 | Objective

- C#简介
- 理解.NET Framework 概念
- 理解并熟悉公共语言运行时的概念及组成
- 熟悉 CTS 和 CLS
- 理解中间语言的概述
- 了解 CLR 的执行过程和内存管理机制
- 熟悉.NET Framework 类库的结构
- 了解.NET Framework 中程序集的概念
- 掌握 CLR 命名空间的定义及引用方法
- 掌握.NET 环境的配置方法
- 了解各种.NET 应用程序的概念

1.1 C#与.NET Framework

C#是随.NET Framework 一起发布的一种新语言，是一种崭新的面向对象的编程语言，强调以组件为基础的软件开发方法。它不但结合了 Visual Basic 的简单易用性，同时也提供了 Java 和 C++语言的灵活性和强大功能。C#在.NET Framework 构架中扮演着一个重要角色，它是 Microsoft 公司面向下一代互联网软件和服务战略的重要内容，也是编写.NET Framework 应用程序的首选。

1.1.1 C#简介

C#是用于创建运行在.NET 公共语言运行库上应用程序的语言之一，它从 C 语言和 C++语言演化而来，是 Microsoft 专门为使用.NET 平台而创建的，并且考虑了其他语言的许多优点，例如 Visual Basic 的易用性。

C#本身是面向对象的语言，C#还进一步提供了对面向组件（Component Oriented）编程的支持。现代软件设计日益依赖于自包含和自描述功能包形式的软件组件。这种组件的关键在于，它们通过属性（Property）、方法（Method）和事件（Event）来提供编程模型；它们具有提供了关于组件的声明信息的属性（Attribute）；它们还编入了自己的文档。C#提供的语言构造直接支持这些概念，这使得 C#语言自然而然成为创建和使用软件组件的首选。

C#具有几个非常优秀的用于构造健壮和持久应用程序的特性，如下所示。

- 垃圾回收将自动回收不再使用的对象所占用的内存。
- 异常处理提供了结构化和可扩展的错误检测和恢复方法。
- 类型安全的语言设计则避免了读取未初始化的变量、数组索引超出边界或执行未经检查的类型强制转换等情形。

此外，C#还具统一的类型系统，所有 C#类型（包括 int 和 string 之类的基础数据类型）都继承于一个唯一的基类型：Object。因此，所有类型都共享一组通用操作，并且任何类型的值都能够以一致的方式进行存储、传递和操作。另外，C#同时支持用户定义的引用类型和值类型，既允许对象的动态分析，也允许轻量结构的内联存储。

为了确保 C#程序和库能够以兼容的方式逐步演进，C#的设计中充分强调了版本控制。许多语言都不太重视这一点，导致采用那些语言编写的程序常常因为其所依赖的库的更新而无法正常工作。C#的设计在某些方面直接考虑到了版本控制的需要，其中包括单独使用的 virtual 和 override 修改、方法重载决定规则以及对显式接口成员声明的支持。

C#是一个易于使用的、能够开发出功能强大、安全、稳定应用程序的语言，在本书的其余部分将详细描述 C#的这些特性。

1.1.2 .NET Framework 简介

.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。

- 提供一致的面向对象的编程环境，而无论对象代码是在本地存储和执行还是在本地执行但在 Internet 上分布、或者是在远程执行的。
- 提供将软件部署和版本控制冲突最小化的代码执行环境。
- 提供可提高代码（包括由未知的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。
- 提供可消除脚本环境或解释环境性能问题的代码执行环境。
- 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。
- 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。

1. .NET Framework 组件

.NET Framework 主要有两个组件：公共语言运行库和.NET Framework 类库。公共

语言运行库是.NET Framework 的基础，可以将运行库看作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，并且还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，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。

.NET Framework 的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，可以使用它开发多种应用程序，这些应用程序包括传统的命令行或图形用户界面（GUI）应用程序，也包括基于 ASP.NET 所提供的最新创新的应用程序（如 Web 窗体和 XML Web Services）。

如图 1-1 所示的.NET Framework 平台上显示了公共运行时和类库与应用程序以及与整个系统之间的关系。

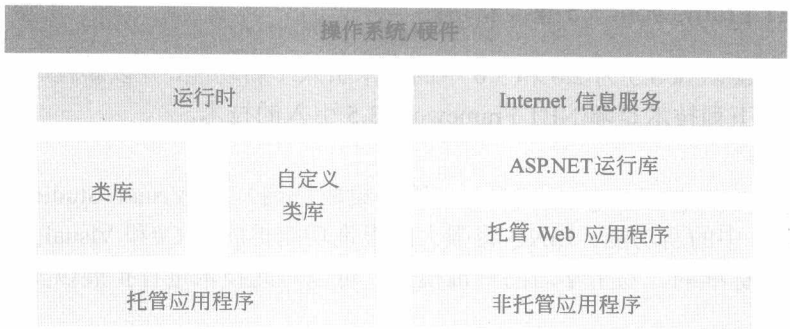


图 1-1　.NET Framework 平台

2. .NET Framework 3.5 简介

.NET Framework 3.5 版以.NET Framework 2.0 版和.NET Framework 3.0 版为基础，包括.NET Framework 2.0 和 3.0 版的 Service Pack。主要包括如下的组件。

- .NET Framework 2.0。
- .NET Framework 2.0 Service Pack 1，它更新包含在.NET Framework 2.0 中的程序集。
- .NET Framework 3.0，它使用.NET Framework 2.0 或.NET Framework 2.0 SP1（如果已安装）中存在的程序集，并且包含.NET Framework 3.0 中引入的技术所必需的程序集。例如，Windows Presentation Foundation(WPF)所必需的 Presentation Framework.dll 和 PresentationCore.dll 就随.NET Framework 3.0 一起安装。
- .NET Framework 3.0 Service Pack 1，它更新在.NET Framework 3.0 中引入的程序集。
- 一些新程序集，它们为.NET Framework 2.0 和 3.0 提供附加功能，同时还提供.NET Framework 3.5 中新采用的技术。

如果在计算机上安装 .NET Framework 3.5 时缺少上述任何组件，则会自动安装它们。应用程序无论针对的是.NET Framework 2.0、3.0 还是 3.5 版，都使用相同的程序集。

例如，对于使用 WPF 并针对.NET Framework 3.0 的应用程序，其所使用的 mscorlib 程序集实例与使用 Windows 窗体并针对.NET Framework 2.0 的应用程序是相同的。如果.NET Framework 2.0 SP1 已安装在计算机上，则 mscorlib.dll 已更新，并且两个应用程序将都使用 mscorlib.dll 的更新版本。

.NET Framework 2.0、3.0 和 3.5 版之间的关系不同于 1.0、1.1 和 2.0 版之间的关系。.NET Framework 1.0、1.1 和 2.0 版是彼此完全独立的，对于其中任何一个版本来说，无论计算机上是否存在其他版本，自己都可以存在于该计算机上。当 1.0、1.1 和 2.0 版位于同一台计算机上时，每个版本都有自己的公共语言运行库、类库和编译器等。应用程序可以选择是针对 1.0、1.1 还是 2.0 版。

3. .NET Framework 3.5 重要新功能

.NET Framework 3.5 为 2.0 和 3.0 中的技术引入了新功能，并以新程序集的形式引入了其他技术。下列技术是随.NET Framework 3.5 引入的技术。

- **LINQ**

LINQ（Language Integrate Query，语言集成查询）是 Visual Studio 2008 和.NET Framework 3.5 中的新功能。LINQ 将强大的查询功能扩展到 C#和 Visual Basic 的语言语法中，并采用标准的、易于学习的查询模式。可以对此技术进行扩展以支持几乎任何类型的数据存储。

- **外接程序和扩展性**

.NET Framework 3.5 中的 System.AddIn.dll 程序集向可扩展应用程序的开发人员提供了强大而灵活的支持。它引入了新的结构和模型，可帮助开发人员完成向应用程序添加扩展性的初始工作，并确保开发人员的扩展在宿主应用程序发生更改时仍可继续工作。

- **Windows Presentation Foundation**

在.NET Framework 3.5 中，Windows Presentation Foundation 包含多个方面的更改和改进，其中包括版本控制、应用程序模型、数据绑定、控件、文档、批注和三维 UI 元素。

- **WCF 和 ASP.NET AJAX 集成**

WCF 与 ASP.NET 中的异步 JavaScript 和 XML(AJAX)功能的集成提供了一个端对端的编程模型，可用于构建可以使用 WCF 服务的 Web 应用程序。在 AJAX 样式的 Web 应用程序中，客户端（例如 Web 应用程序中的浏览器）通过使用异步请求来与服务器交换少量的数据。在 ASP.NET 中集成 AJAX 功能可提供一种生成 WCF Web 服务的简单方法，通过使用浏览器中的客户端 JavaScript 可以访问这些服务。

- **ClickOnce 清单**

新增了一些密码类，用于验证和获取有关 ClickOnce 应用程序的清单签名的信息。

在这里仅列举了.NET Framework 3.5 中的重要新功能和新特性，但不是全部。读者如果需要了解更多，可到网站 http://www.microsoft.com 上查找。

1.2 公共语言运行时

.NET Framework 提供了一个称为公共语言运行时（Common Language Runtime，CLR）的运行环境，它运行代码并提供使开发过程更轻松的服务。作为.NET Framework 的核心组件，它是运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。图 1-2 所示为公共语言运行时环境中的各个部分及其提供的重要服务。

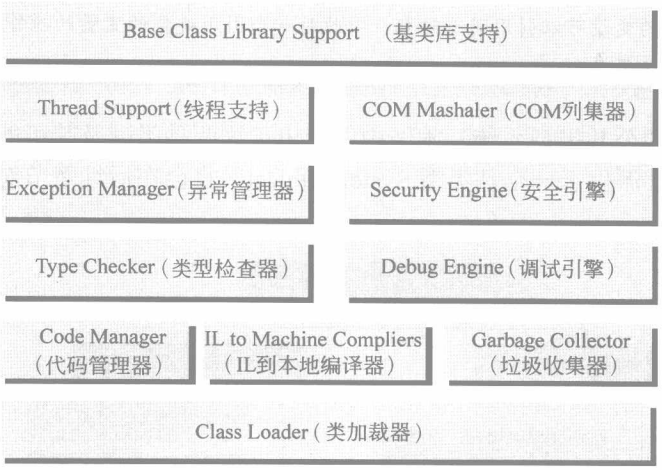


图 1-2 公共语言运行时环境

公共语言运行时通过公共类型系统（Common Type System，CTS）和公共语言规范（Common Language Specification，CLS）定义了标准数据类型和语言间互操作性的规则。Just-In-Time 编辑器在运行应用程序之前把中间语言（Intermediate Language，IL）代码转换为可执行代码。CLR 还管理应用程序，在应用程序运行时为其分配内存和解除分配内存。

1.2.1 公共类型系统

公共类型系统（Common Type System，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。公共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。

公共类型系统支持.NET Framework 提供的两种常用类型，其中每一类又可细分成子类型。

- **值类型**

值类型直接包含它们的数据，值类型的实例要么在堆栈上，要么内联在结构中。值

类型可以是内联的（由运行库实现）、用户定义的或枚举的。

❑ 引用类型

引用类型存储对值的内存地址的引用，位于堆上。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。类类型是用户定义的类、装箱的值类型和委托。

每个值类型的变量都有自己的数据副本，因此对一个值类型变量的操作不会影响其他变量。引用类型的变量可以引用同一对象，因此对一个引用类型的变量的操作会影响另一个变量所引用的同一对象。

图 1-3 所示的公共类型系统基本结构中给出了这两种类型及其可能出现的子类型。所有的这些类型都派生于一个基类型 System.Object，将会在后续章节对这些类型进行详细介绍。

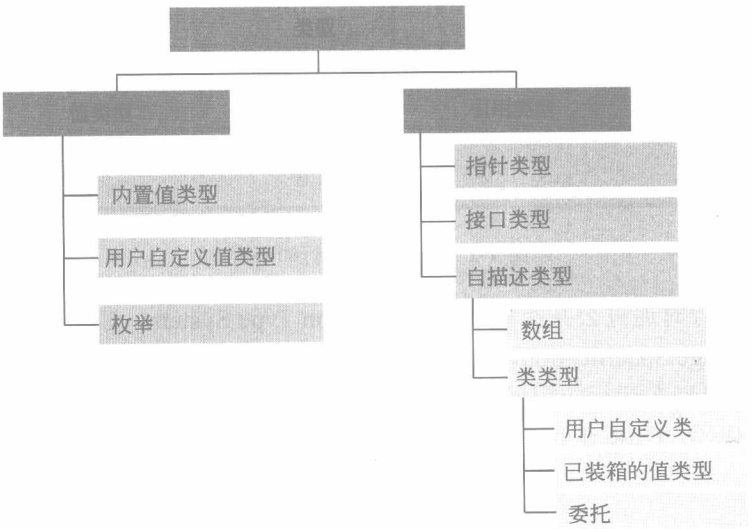


图 1-3 公共类型系统基本结构

1.2.2 公共语言规范

公共语言规范（Common Language Specification，CLS）是一组结构和限制条件，用作库开发者和编译器编写者的指南。它使任何支持 CLS 的语言都完全使用库，并且使用这些语言相互集成。公共语言规范是公共类型系统的子集，它们一起定义了允许不同编程语言的标准集，由这些编程语言编写的应用程序可以互操作。

CLS 和.NET 自身都依赖于 Windows API（Application Programming Interface，应用程序编程接口）提供的低级服务。Windows API 提供了像菜单、按钮、列表框和标签等基本控件的类，提供了基本的 Windows 服务来管理文件、进程和内存。

CLS 定义了所有基于.NET Framework 的语言都必须支持的最小功能集。CLS 定义的规则可以概括如下。

- CLS 定义了命名变量的标准规则。例如，与 CLS 兼容的变量名都必须以字母开始，并且不能包含空格。变量名之间必须有所区别，除了变量名之间的大小写之外。
- CLS 定义了原语数据类型，如 Int32，Int64，Single，Double 和 Boolean。
- CLS 禁止无符号数值数据类型。有符号数值数据类型的一个数据位被保留来指示数值的正负。无符号数据类型没有保留这个数据位。
- CLS 定义了对基于 0 的数组的支持。
- CLS 指定了函数参数列表的规则以及参数传递给函数的方式。例如，CLS 禁止使用可选的参数。
- CLS 定义了事件名和参数传递给事件的规则。
- CLS 禁止内存指针和函数指针，但是可以通过委托提供类型安全的指针。

这里将完全兼容 CLS 的语言称为兼容 CLS 语言，除此之外任何语言都可以扩展基本的 CLS 需求。例如，有些语言支持无符号整型。但是，不鼓励使用非标准的功能，因为这样做就妨碍了语言之间的互操作性。

1.2.3 中间语言

使用.NET 语言开发的任何应用程序都必须在执行之前编译为可执行文件。传统的可执行文件包含了允许在特定 CPU 体系结构上执行的本机指令。不同厂商生产的 CPU 体系结构都不同，它们具有不同的寄存器，并且执行一套独有的指令。除了不同的 CPU 厂商会制作各种不兼容的硬件之外，即使是同一个厂商，如 Intel，也常常制造出带有特殊指令附加功能的 CPU。把应用程序编译为独立于 CPU 的中间语言，当用户执行应用程序时，就会把中间语言优化为特定 CPU 的可执行文件。因此，Microsoft 公司为每一种目标 CPU 版本提供了不同版本的 JIT（Just In Time）编译器，以便利用各种 CPU 的独特功能，进而提高性能。

使用.NET 语言开发的任何应用程序在执行之前都会编译为目标计算机能够理解的语言，即本机代码，在.NET Framework 下这个过程可分为两个阶段。由于 CPU 体系结构的不同，首先把应用程序编译成一种称为中间语言（Microsoft Intermediate Language，MSIL）的独立于硬件的格式，中间语言的主要特征如下。

- 面向对象和使用接口。
- 值类型和引用类型之间的巨大差别。
- 强数据类型。
- 使用异常来处理错误。
- 使用特性（Attribute）。

在编译应用程序时，创建的 MSIL 代码存储在一个程序集中，程序集包括可执行的应用程序文件（这些文件可以直接在 Windows 上运行，不需要其他程序，其扩展名是.exe）和其他应用程序使用的库（扩展名为.dll）。除了 MSIL 之外，程序集还包括元信息（程

序集中包含的数据，也称为元数据）和可选的资源。元信息允许程序集完全自我描述，不需要其他信息就可以使用程序集。这样部署应用程序就非常简单了，只需要把文件复制到远程计算机上的目录下即可，因为不需要目标计算机上的其他信息，所以只在安装了.NET CLR 的计算机中执行。

第二个阶段就是使用 JIT 的阶段，它把 MSIL 编译为专用于目标操作系统和目标机器结构的本机代码，只有这样目标操作系统才能执行应用程序。使用 JIT 编译器把中间语言转换为可执行文件的过程通常称为 JITting。

1.2.4 托管执行过程

CLR 执行的代码称为托管代码（Managed Code），它的作用之一就是防止一个应用程序干扰另一个应用程序的运行，这个过程称为类型安全性（Type Safety）。使用类型安全的托管代码，一个应用程序就不会覆盖另一个应用程序分配的内存。C#开发的应用程序的执行由 CLR 控制，可以被视为托管代码。创建托管代码的方法如下。

（1）选择一个合适的编译器，它能够生成适合 CLR 执行的代码，并且使用.NET Framework 提供的资源。Microsoft 公司目前提供了 4 种兼容.NET Framework 的语言。

（2）把应用程序编译为独立于机器的中间语言。

（3）在执行时，必须把中间语言代码转换为本机可执行文件。本机可执行文件可以在目标 CPU 上执行。这个过程称为 Just-In-Time（JIT）编译。

（4）应用程序执行时，会调用.NET Framework 和 CLR 提供的资源。

上述托管代码的创建过程如图 1-4 所示。

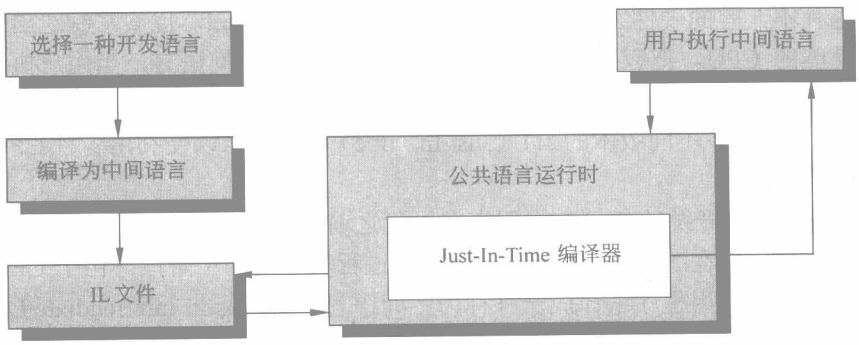


图 1-4 托管代码的创建过程

有些 Visual Studio .NET 生成的可执行文件不依赖于 CLR 提供的服务，这种类型的可执行文件称为未托管的可执行文件（Unmanaged Executable）或者未托管代码。

.NET Framework 使用托管代码执行过程主要有如下优点。