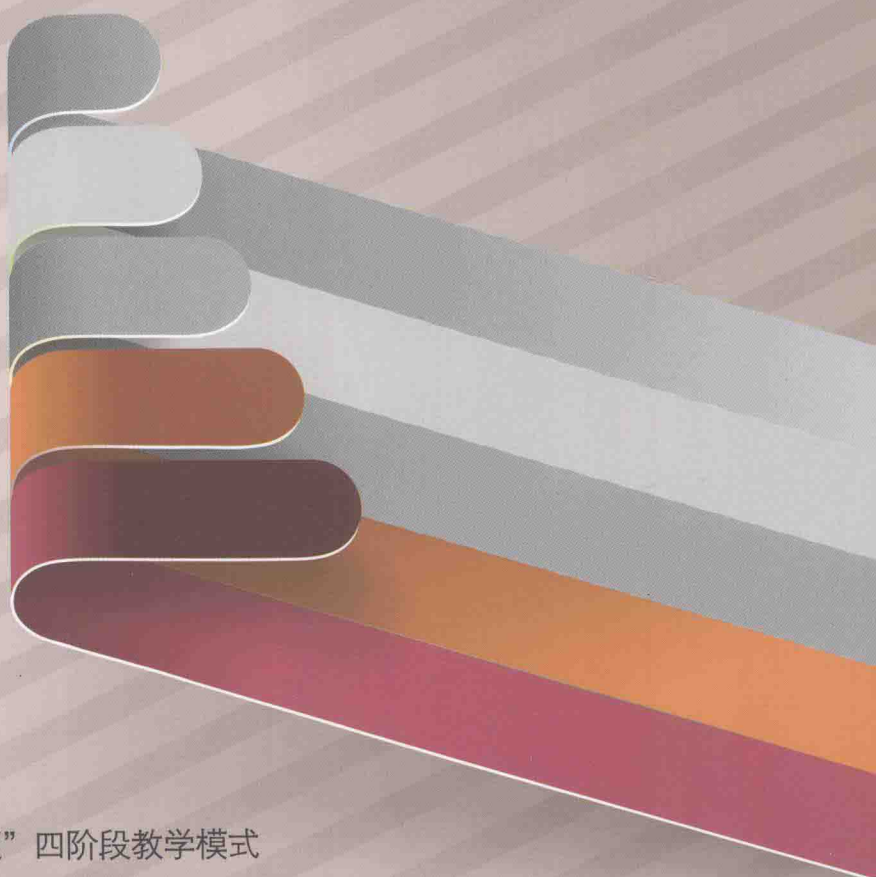
软件工程师培养丛书

# C#编程技术基础

武汉厚溥教育科技有限公司　编著



论→总结→上机→习题”四阶段教学模式

- 理论结合实践，注重动手能力培养
- 任务驱动讲解，有效激发学习兴趣
- 典型项目案例，扎实培养专业素质
- 教学做一体化，极大提高教学效率

清华大学出版社

软件工程师培养丛书

# C#编程技术基础

武汉厚溥教育科技有限公司　编著

清华大学出版社

北　京

## 内 容 简 介

本书按照高等院校、高职高专计算机课程基本要求，以案例驱动的形式来组织内容，突出计算机课程的实践性特点。本书共包括 12 章：深入了解.NET Framework、C#语法基础、类和对象(一)、类和对象(二)、C# OOP 深入、C#事件处理、继承和多态、抽象类和接口、常用类、集合和泛型、调试和异常处理以及 C#中的文件处理。

本书附赠 PPT 教学课件和案例源文件，这些教学资源可通过http://www.tupwk.com.cn/downpage下载。

本书内容安排合理，层次清楚，通俗易懂，实例丰富，突出理论和实践的结合，可作为各类高等院校、高职高专及培训机构的教材，也可供广大程序设计人员参考。



**图书在版编目(CIP)数据**

C#编程技术基础 / 武汉厚溥教育科技有限公司　编著. —北京：清华大学出版社，2014
(软件工程师培养丛书)
ISBN 978-7-302-34747-7

Ⅰ. ①C…　Ⅱ. ①武…　Ⅲ. ①C 语言—程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 295213 号

**责任编辑**：刘金喜　蔡　娟
**封面设计**：崔东方
**版式设计**：妙思品位
**责任校对**：邱晓玉
**责任印制**：刘海龙

**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
课 件 下 载：http://www.tup.com.cn,010-62794504

**印 装 者**：北京密云胶印厂
**经　　销**：全国新华书店
**开　　本**：185mm×260mm　　**印　　张**：20.25　　**字　　数**：332 千字
**版　　次**：2014 年 1 月第 1 版　　**印　　次**：2014 年 1 月第1 次印刷
**印　　数**：1～3000
**定　　价**：36.00 元

产品编号：057154-01

# 编委会

# 前　言

C#(C Sharp)是微软(Microsoft)公司为.NET Framework 量身订做的一种面向对象编程语言，于 2000 年 6 月发布。C#拥有 C/C++的强大功能和 Visual Basic 简单易用的特性，是第一个面向组件(component-oriented)的程序语言，与 C++和 Java 一样也是面向对象(object-oriented)程序语言，但是 C#程序只能在 Windows 下运行。C#是微软公司研究员 Anders Hejlsberg 的研究成果。

本书隶属于“软件工程师培养丛书”中的一本专业基础教材，该丛书是由武汉厚溥教育科技有限公司开发，以培养符合企业需求的软件工程师应用开发、实施为目标的 IT 职业教育丛书。在开发该丛书之前，我们对 IT 行业的岗位序列做了充分的调研，包括研究从业人员技术方向、项目经验和职业素质等方面的需求，通过对面向的学生的特点、行业需求的现状以及实施等方面的详细分析，结合“厚溥”对软件人才培养模式的认知，按照软件专业总体定位要求，进行软件专业产品课程体系设计。该丛书集应用软件知识和多领域的实践项目于一体，着重培养学生的熟练度、规范性、集成和项目能力，从而达到预定的培养目标。

本书共包括 12 章：深入了解.NET Framework、C#语法基础、类和对象(一)、类和对象(二)、C# OOP 深入、C#事件处理、继承和多态、抽象类和接口、常用类、集合和泛型、调试和异常处理以及 C#中的文件处理。

我们对本书的编写体系做了精心的设计，按照“理论学习—知识总结—上机操作—课后习题”这一思路进行编排。“理论学习”部分描述通过本案例要达到的学习目的与涉及的相关知识点，使学习目标更加明确；“知识总结”部分概括案例所

涉及的知识点，使知识点完整系统地呈现；“上机操作”部分对案例进行了详尽分析，通过完整的步骤帮助读者快速掌握该案例的操作方法；“课后习题”部分帮助读者理解章节的知识点。本书在内容编写方面，力求细致全面；在文字叙述方面，注意言简意赅、重点突出；在案例选取方面，强调案例的针对性和实用性。

本书凝聚了编者多年来的教学经验和成果，可作为各类高等院校、高职高专及培训机构的教材，也可供广大程序设计人员参考。

本书PPT教学课件和案例源文件可通过http://www.tupwk.com.cn/downpage下载。

本书由武汉厚溥教育科技有限公司编著，由翁高飞、王鹏等多名企业实战项目经理编写。本书编者长期从事项目开发和教学实施，并且对当前高校的教学情况非常熟悉，在编写过程中充分考虑到不同学生的特点和需求，加强了项目实战方面的教学。本书编写过程中，得到了武汉厚溥教育科技有限公司各级领导的大力支持，在此对他们表示衷心的感谢。

参与本书编写的人员还有：武汉商学院曹静老师、荆州职业技术学院方风波老师、武汉工程职业技术学院邹治伟和彭莉老师、湖北三峡职业技术学院李建利老师、武汉软件工程职业学院谢日星老师、黄冈职业技术学院吴金秀老师、湖北国土资源职业学院徐霞和明素华老师等。

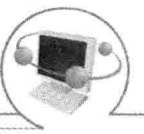
限于编写时间和编者的水平，书中难免存在不足之处，希望广大读者批评指正。

服务邮箱：wkservice@163.com。

编　者

2013年10月

# 目　录



# 第1章
# 深入了解 .NET Framework

## 课程目标

- 回顾.NET Framework
- .NET Framework的体系结构
- .NET Framework的组件
- .NET程序编译原理
- 框架类库简介和简单使用体验

## 简 介

.NET Framework 是一个框架，是一门新技术，是一个非常大的代码库。目前最新版本是.NET Framework 4.5，它包含很多概念，刚接触时不是很容易理解，但是快速浏览这些基础知识对于理解如何利用 C#进行编程是非常重要的，所以先学习.NET Framework 是不可避免的。

C#是一门新的面向对象的编程语言，我们不能孤立地使用 C#语言，而必须与.NET Framework 一起考虑。C#是专门为与 Microsoft 的.NET Framework 一起使用而设计的。

# 1.1 .NET Framework 回顾

.NET Framework 是 Microsoft 为开发应用程序而创建的一个富有革命性的新平台，它定义了应用程序的开发和运行环境。.NET Framework 可以创建 Windows 应用程序、Web 应用程序、Web 服务和其他各种类型的应用程序。

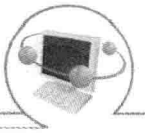
.NET Framework 的设计方式保证它可以用于各种语言，包括本书要介绍的 C#语言，以及 C++、Visual Basic、JScript，甚至一些旧的语言，如 COBOL。目前还在不断推出更多的.NET 版本的语言。

.NET Framework 包括最早期的 1.0 版，目前经历了 1.1、2.0、3.0、3.5、4.0 和最新的 4.5 版。

# 1.2 .NET Framework 的体系结构

.NET Framework 是一个创建、部署和运行应用程序的多语言多平台环境。.NET Framework 包含以下两个主要组件。

- 公共语言运行时(Common Language Runtime，CLR)



- .NET Framework 类库(Framework Class Library，FCL)

重用代码，避免重复开发并缩短时间，这一直是软件开发人员的目标。.NET Framework 提供了许多开发人员可重用的基础类，包括线程、文件、数据库支持、XML 分析和数据结构等各个方面，并且这些类库可用于所有支持.NET Framework 的编程语言。通过 CLR 支持，任何.NET 语言都可以使用.NET 类库中的所有类。

除 CLR 和 FCL 外，.NET Framework 还对应着两种编程模型：ASP.NET(Web Forms 及 Web Services)和 WinForms。每种编程模型又对应着很多不同的编程语言，如图 1-1 所示。

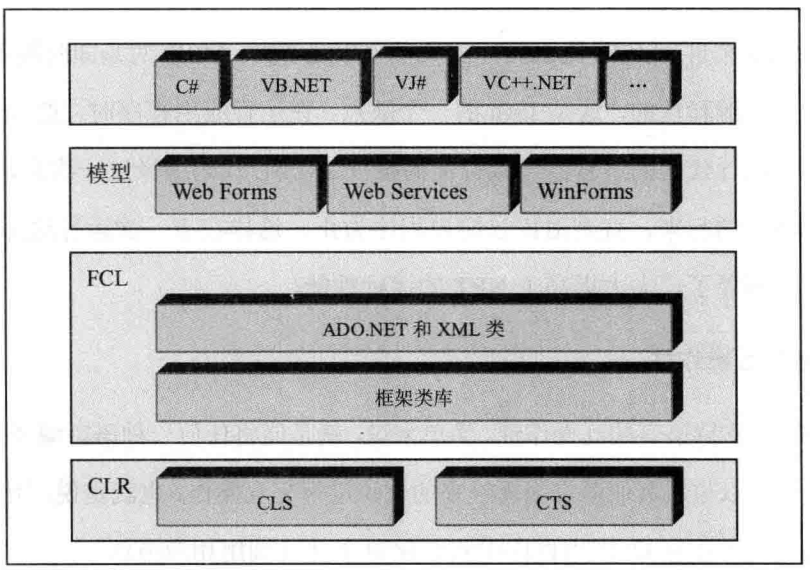



图 1-1

# 1.3　.NET Framework 的组件

## 1.3.1　公共语言运行时

.NET Framework 的核心是其运行库的执行环境，称为公共语言运行时(CLR)或.NET 运行库。通常将在 CLR 的控制下运行的代码称为托管代码。

但是，在 CLR 执行开发的源代码之前，需要编译它们(语言本身自己的编译器)。在.NET 中，编译分为如下两个阶段。

- 把源代码编译为 Microsoft 中间语言(Microsoft Intermediate Language，MSIL)。
- CLR 把 MSIL 编译为平台专用的机器代码。

CLR 从某种意义上理解相当于 Java 中的 Java 虚拟机(JVM)，而 MSIL 相当于 Java 中的字节码(.class 文件)。MSIL 总是即时编译(称为 JIT 编译)为相应平台的机器代码，这一点与 Java 也很相似。

.NET 代码托管有如下优点。

1. 提高性能

相对于 Java 来讲，MSIL 比 Java 字节码的作用还要大。MSIL 总是即时编译的，而 Java 字节代码常常是解释性的，这是 Java 的一个缺点，在运行应用程序时，把 Java 字节代码转换为内部可执行代码的过程会导致性能的损失。MSIL 代码编译过一次后，得到的内部可执行代码就存储起来，直到退出该应用程序为止。这样在下一次运行这部分代码时，就不需要重新编译了，大大提高了.NET 的运行性能。

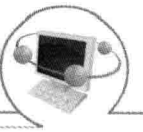
2. 语言的互操作性

使用 MSIL 支持语言的互操作性。简单来说，就是能将任何一种语言编译为中间代码，编译好的代码可以与从其他语言编译过来的代码进行交互操作。也就是说，用 Visual Basic 语言声明的类，可以在 C#应用程序中实例化对象并且调用相应方法。

无论用 Visual Basic 编写代码，还是用 C#或者 C++编写代码，CLR 都可以把源代码编译成 MSIL，也就是中间语言，这样才能实现语言互操作。那么，CLR 是如何做到这一点的呢？CLR 包含两个组成部分：CTS(Common Type System，通用类型系统)和 CLS(Common Language Specification，公共语言规范)。CLR 用这两个组件来完成语言互操作的功能。

## 1.3.2 通用类型系统

CTS 定义了一套可以在中间语言中使用的预定义数据类型，所有面向.NET Framework 的语言都可以生成最终基于这些类型的编译代码。也就是说，通用类型系统用于解决不同编程语言的数据类型不同的问题。这也为跨语言的实现做出了重大贡献。例如，在 Visual Basic 中定义整型变量，是关键字 integer，而在 C#中用的是关键字 int，但无论 Visual



Basic 还是 C#，经编译后都映射为 System.Int32，所以 CTS 实现了不同语言数据类型的最终统一。

### 1.3.3　公共语言规范

编程语言之间不仅仅是数据类型的不同，语法也有非常大的区别。所以需要定义公共语言规范，CLR 定义了所有编程语言必须遵守的共同标准。公共语言规范和通用类型系统一起确保语言的互操作性。CLS 是一个最低标准集，所有面向.NET 的编译器都必须支持它。只要遵守这个标准编写的程序，就可以在.NET 框架下实现互相操作。例如，在 C#中变量名是区分大小写的，而 VB.NET 不区分大小写，这时，CLS 就规定编译后的 MSIL 必须除了大小写以外有其他的区别。

## 1.4　.NET 程序编译原理

编译 C#.NET 应用程序操作步骤如下。

(1) 使用 C#语言编写应用程序代码。

(2) 把 C#源程序编译为 MSIL，以程序集的形式存在，CPU 不执行程序集，如图 1-2 所示。

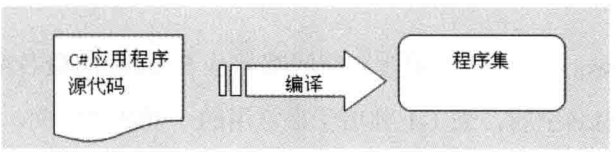



图 1-2

(3) 在执行代码时，必须使用 JIT 编译器将程序集编译为本机代码，如图 1-3 所示。

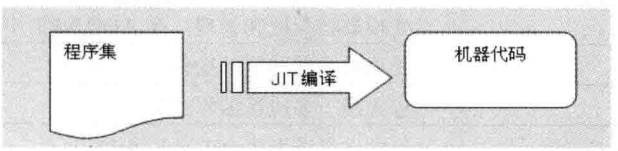



图 1-3

(4) 在托管的 CLR 环境下运行本机代码，程序执行结果显示出来，如图 1-4 所示。

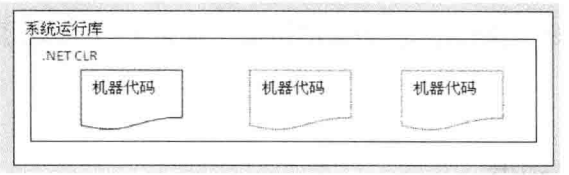



图 1-4

# 1.5 .NET 框架类库

## 1.5.1 框架类库

在前面讲解.NET 框架结构时，了解到框架类库(FCL)有非常多的命名空间和类，这些类使用非常方便，且功能强大。框架类库提供了实现基本功能的类，如输入/输出、字符串的操作、网络通信等。

在.NET 开发中，应用程序实现的很多功能不需要我们编写大量代码，只需要直接调用框架类库中相应的类就可以实现。那么，如何使用这些类呢？我们知道，命名空间用来将具有相关功能的一些类在结构上进行分组和管理，所以要使用框架类库中的类就必须了解.NET 框架中的常用命名空间。框架类库中包含了 170 多个命名空间和上千个类，下面就来学习类库中的一些主要命名空间。

## 1.5.2 框架类库中的命名空间

在.NET Framework 中，所有的命名空间都是从 System 的命名空间形成的。System 命名空间又称为根命名空间，表 1-1 列出了最常用的一些命名空间。

表 1-1

| 命 名 空 间 | 描　　述 |
|---|---|
| System.Drawing | 处理图形和绘图 |
| System.Data | 处理数据存取和管理，在 ADO.NET 中扮演重要角色 |
| System.IO | 管理对文件和流的操作 |
| System.Threading | 包含用于多线程编程的类 |
| System.Collections.Generic | 包含定义泛型集合的接口和类 |
| System.Collections | 包含定义各种对象集合的接口和类 |
| System.Windows.Forms | 包含用于开发 Windows 窗体应用程序的控件和类 |
| System.Net | 对网络协议进行编程 |