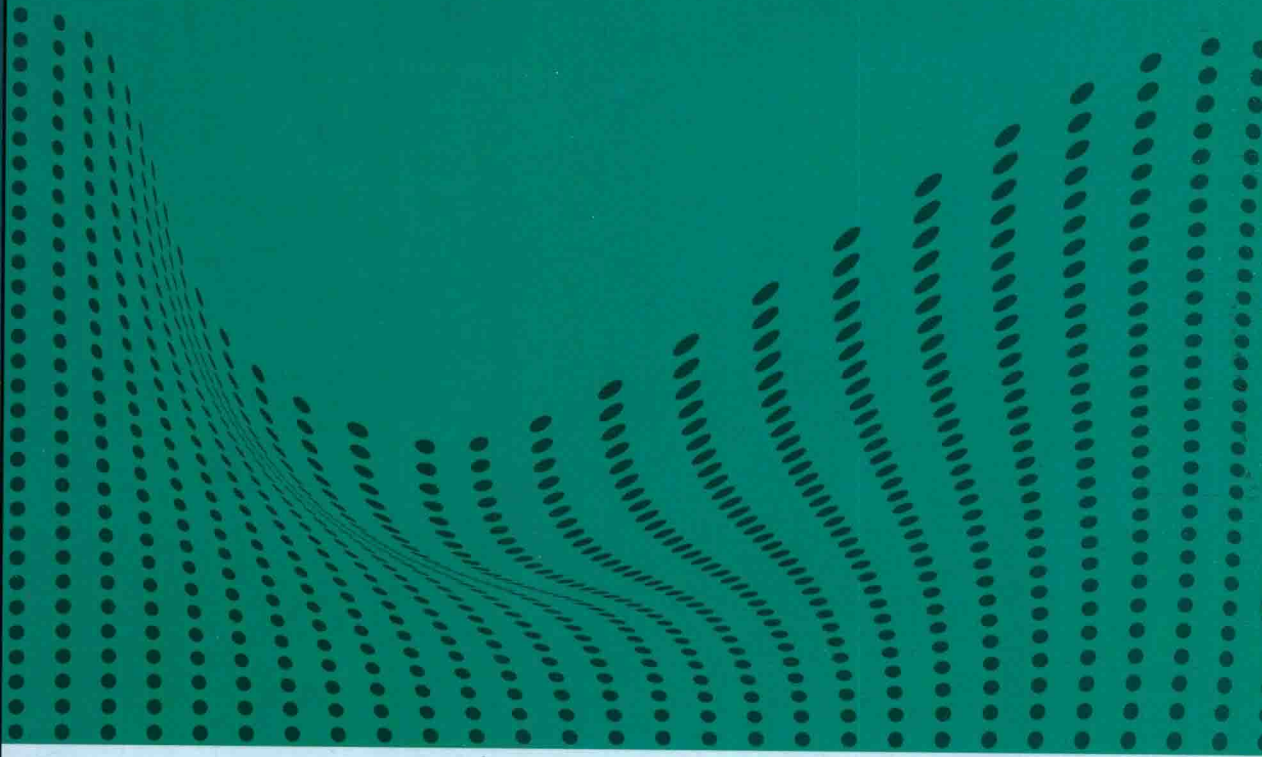
21世纪高等学校计算机系列规划教材



C#程序设计

杨律青　编著

上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

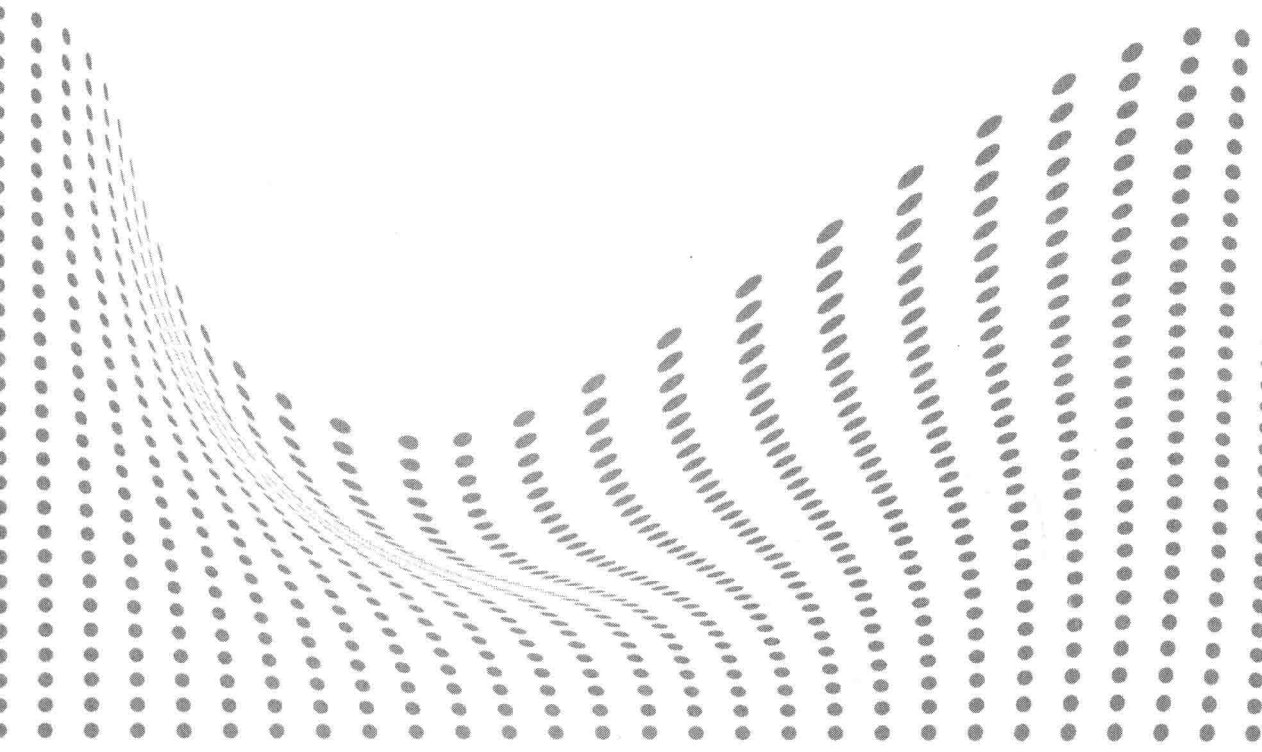
巍巍交大　百年书香
www.jiaodapress.com.cn
bookinfo@sjtu.edu.cn

策划编辑/蒋湘群
责任编辑/于江红　陈祖英
封面设计/黄燕美

杨律青　博士后，高级工程师，现任厦门大学软件学院副教授，曾任多家大型集团企业IT高级经理、技术总监、CIO等职，目前兼任中国计算机模拟与信息技术学会理事、厦门物联网协会专家组成员。20多年IT项目开发和管理经验，参与国家自然科学基金项目2项，主持过企事业大型IT项目20多项，主持或指导各类IT项目100多项，获取2010年IBM中国优秀教师奖教金、2006年全国化工行业企业信息化优秀成果奖、全国化工行业企业信息化实施贡献奖等。主要研究方向为.NET技术平台、C#面向对象开发、企业信息化、RFID技术与物联网等，出版《软件项目管理》等教材多部，发表IT类高水平学术论文19篇，其中EI检索14篇。

ISBN 978-7-313-09198-7
9 787313 091987 >
定价：39.00元

21世纪高等学校计算机系列规划教材



C#程序设计

杨律青　编著

上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内 容 提 要

本书以基础性内容为主，深入浅出地介绍了.NET平台基础、C#基础知识、C#的面向对象设计、使用C#开发窗体应用程序、ASP.NET开发、ASP.NET控件与内置对象、文件操作、C#的多线程、数据库应用开发、GDI+编程、C#应用程序的安装和部署、.NET的扩展内容(包括WPF、WCF和WF)等。

本书可以作为高等院校“C#程序设计”课程的教材，也可作为从事基于.NET的软件系统分析与设计、软件开发及应用等工作的工程师和工作人员的参考书。

图书在版编目(CIP)数据

C#程序设计/杨律青编著. —上海：上海交通大学出版社，2013

ISBN 978-7-313-09198-7

Ⅰ.①C… Ⅱ.①杨… Ⅲ.①C语言—程序设计 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2012)第268853号

C#程序设计

杨律青 编著

上海交通大学出版社出版发行

(上海市番禺路951号 邮政编码：200030)

电话：64071208 出版人：韩建民

北京振兴源印务有限公司印刷 全国新华书店经销

开本：787mm×1 092mm 1/16 印张：21.5 插页1 字数：523千字

2013年1月第1版 2013年1月第1次印刷

ISBN 978-7-313-09198-7/TP 定价：39.00元

前 言 Preface

随着信息技术的发展，微软的.NET平台已成为当今两大软件开发平台之一，它有着很好的发展前景。现实中使用.NET作为开发技术和解决方案的案例越来越多，企事业单位对.NET开发的需求也日益增加。

学习面向对象的Windows应用程序设计可采用不同的工具和语言。C#语言是.NET框架中新一代的开发工具，也是.NET平台上最为重要的开发语言，是一种现代的、完全面向对象的语言。它简化了C++语言在类、命名空间、方法重载和异常处理等方面的操作，摒弃了C++语言的复杂性，更易使用，更少出错；它使用组件编程，和VB一样容易使用。并且C#语言的语法和C++语言、C语言的语法非常相似，在第一门高级程序设计语言课程是C语言或C++语言的情况下，学习C#语言是比较轻松的，也可以在有高级语言基础、无C语言或C++语言学习基础的情况下快速掌握C#语言。因此，把C#语言作为学习面向对象Windows应用程序设计的语言是比较合适的。

微软公司的Microsoft .NET框架是新一代软件开发模型，在ASP.NET中提供的设计Web应用程序的可视化工具能像设计Windows应用程序一样，使用控件的属性、事件和方法等面向对象的概念设计Web应用程序。因此，在Microsoft .NET框架中，设计Windows应用程序的方法和设计Web应用程序的方法十分类似，而且两种设计方法联系紧密，如文件系统和ADO.NET是两种方法都要使用的。在这种情况下，把Windows应用程序和Web应用程序的设计作为一门课程开设是比较合适的。本书的前半部分讲述Windows Forms应用程序的设计方法，后半部分讲述Web应用程序和ASP.NET的设计与开发方法，也适当加入了部分深入的编程内容。

C#语言是继C语言和C++语言后微软重点推出的重要、易学和实用的计算机程序设计语言。也正是因为.NET平台的强大和易用，目前开设.NET、C#程序设计课程的高校和专业如雨后春笋般地增加。该课程不仅在计算机类专业开设，也出现在信息管理、自动化等相关专业中。对于那些非计算机类专业，也通过编写C#程序来解决学习和科研工作中的各类问题。一方面，C#为操作系统、数据库、计算机网络等课程提供了程序设计基础；另一方面，它也尽早为学生课外实践提供了一个实用的工具。

本书是在结合计算机教学的发展趋势和满足现实开发迫切需求的情况下编写的，循序渐进地介绍基于C#编程的基础知识，以及.NET平台上的开发过程、开发方法，内容几乎涵盖了使用C#语言开发Windows应用程序和Web应用程序的各个方面，在内容组织上力

求结构合理、内容全面且系统、富有践性、重点突出。本书还融合了软件工程、项目开发等思想与方法，书中的程序代码、实际案例等部分的内容丰富、翔实，以帮助读者提高动手和操作能力。

在内容表达上，本书做到图文结合、表述通俗且清晰，用开发工具中产生的图文和数据来说明和解释较抽象的编程思路和编程语言。为了说明问题，在介绍程序设计方法时，一般使用具有针对性的例子进行说明，在例子中尽量避免不相关的知识点和无关的代码，使例子短小精悍。对于比较复杂的问题，将问题分解为多个步骤，分别给出详细的设计步骤，减少读者的阅读难度。书中的例子都比较完整，绝大部分都可以在计算机中运行。

本书是在讲授 Visual Studio 2010 开发工具和 C# 应用程序课程的讲义基础上整理编辑而成的。它整合了作者多年积累的. NET 和 C# 课程的教学资源，通过多种渠道收集了许多实用的案例和实验，是作者多年教学经验的结晶和总结。

本书定位于基础性教材，主要提供给计算机和信息管理相关专业的学生使用，可作为计算机科学与技术、软件工程、信息管理等专业的本科生的教材。对于希望了解. NET 和 C# 开发的各类读者，本书也是一本较好和实用的参考读物。而对于部分较深入的内容，可供有经验的程序员开发时参考。

本书由厦门大学杨律青副教授编著。作者在编写本书的过程中投注了大量时间与精力，阅读了大量书籍文献，也参考了专业论坛中的讨论及一些专业网站的文章，力图将自身的经验系统化地论述，但由于时间的原因，书中仍存在不足，欢迎读者批评指正。联系邮件：lqyang@xmu. edu. cn。

编　者

目 录 Contents

第 1 章 .NET 平台基础

.NET Framework 是微软公司为开发应用程序而创建的一个富有革命性的新平台，其版本从 1.0 开始发展到现在的 4.0，除了在 Windows 类的操作系统上运行外，还可以在个人数字助手(PDA)类设备和一些智能电话上使用。.NET Framework 可以作为集成各种设备的开发平台，它并没有限制应用程序的类型，利用它可以创建 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 的设计方式保证了它可以用于多种语言，除了本书介绍的 C# 语言，还有 C++、Visual Basic、JScript 语言，甚至包括 COBOL 语言。为此，还推出了这些语言的 .NET版本，目前还在不断推出更多的 .NET 版本的语言。所有这些语言都可以访问 .NET Framework，它们还可以彼此交互，如 C# 开发人员可以使用 Visual Basic 程序员编写的代码，反之亦然。

本章主要介绍 C#(发音为 C Sharp)语言的特点以及 .NET Framework 的工作原理。通过本章的学习，使读者对 C# 和 .NET Framework 有一个概括性的认识，这对于理解如何利用 C# 进行编程是非常重要的。

1.1 .NET 平台介绍

1.1.1 .NET 发展及解决方案

1). NET Framework 简介

.NET Framework 提供了一种开发平台，主要分为 4 个部分：通用语言开发环境、.NET 基础类库、.NET 开发语言和 Visual Studio .NET 集成开发环境。微软公司从发布第 1 个

.NET Framework 以来，已经发布了 1.0 版、1.1 版、2.0 版、3.0 版、3.5 版，目前最新的版本是. NET Framework 4.0，也是目前功能最强大和最完善的一个版本。开发人员可以使用 .NET Framework 创建 Web 网站、Web 服务应用程序、Windows 应用程序以及智能设备应用程序等。Microsoft . NET Framework 的体系结构如图 1-1 所示。

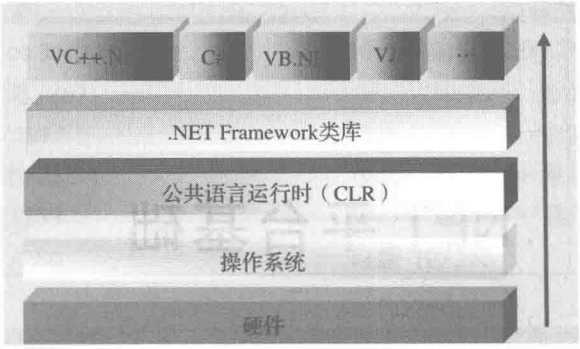


图 1-1　Microsoft . NET Framework 的体系结构

简单地说，. NET Framework 类似于 JVM，其包含两个主要的组件：公共语言运行时(CLR)和统一的类库。CLR 是. NET Framework 的核心执行环境，又称. NET 公共语言运行库。. NET Framework 类库是一个内容丰富的类集合，它可以完成以前要通过 Windows API 来完成的绝大多数任务。其中，. NET 的统一类库又包含线程、文件输入/输出(I/O)、数据库支持、XML 解析、数据结构等内容。. NET Framework 的组件如图 1-2 所示。

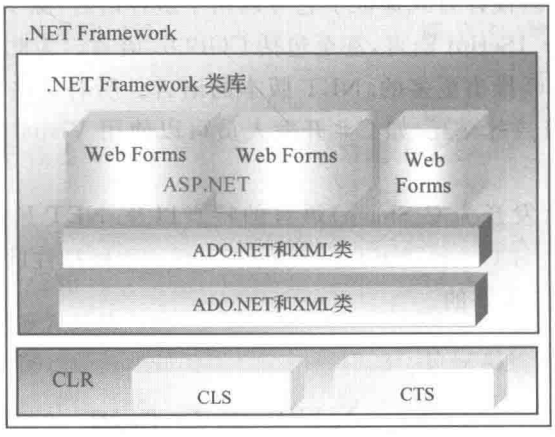


图 1-2　. NET Framework 的组件

2). NET 是演变的结果

. NET 的产生是多重作用的结果，它融合计算和通信，形成组合体，而不是单一的一座孤岛。. NET 实现了分布式计算在本地进行，将软件变成服务，使得应用程序可以由任何人在任何地方使用。

3). NET 以互联网为核心

. NET 以互联网为核心，主要体现在以下方面：

(1)用户数据存放在网络上可以随时随地进行访问；

(2). NET 是以 Internet 为中心的一种全新的平台；

(3)可以创建通过任何浏览器、任何设备访问的应用程序；

(4). NET 应用程序利用了 Internet 的功能；

(5)可以从任何. NET 设备中访问数据。

4)执行. NET 程序

. NET 程序经过两次编译，形成. exe 等可执行文件，其过程如图 1-3 所示。

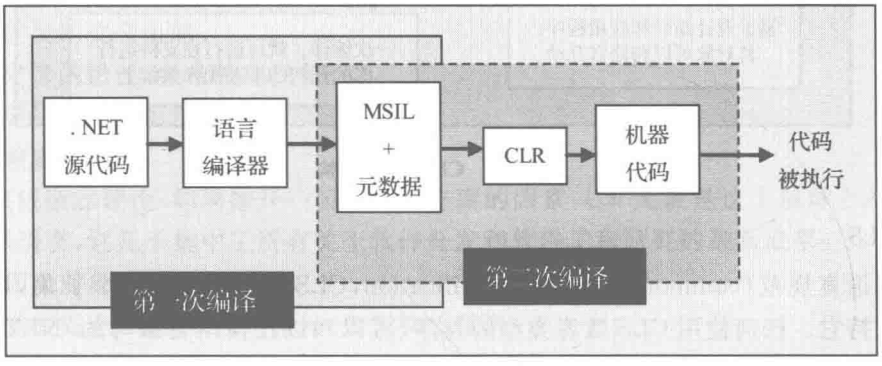


图 1-3　执行. NET 程序需要两次编译

1.1.2　.NET 平台构造块简介

. NET 平台构造块包括 CLR、CLS 和 CTS，下面分别介绍其基本概念和含义。

1)CLR

公共语言运行时(common language runtime，CLR)又称公共语言运行库，它和 Java 虚拟机一样也是一个运行时环境，负责资源管理(内存分配和垃圾收集)，并保证应用和底层操作系统之间必要的分离。

公共语言运行库是. NET Framework 的基础，它提供了一个执行时的管理环境，以及内存管理、线程管理和远程处理、类型安全检查等核心服务。通常将在 CLR 中运行的代码称为托管代码(managed code)。

打个比喻，可以将公共语言运行库想象为人类生存的地球，它提供能源、水、自然资源，生活在地球上的人们则可以看做托管代码。

Microsoft 中间语言(MSIL)由一组特定的指令组成，这些指令指明如何执行代码。即时编译器(Just-In-Time Compiler，JIT 编译器)的主要工作是将普通 MSIL 代码转换为可以直接由 CPU 执行的计算机代码。

CLR 的编译架构如图 1-4 所示。

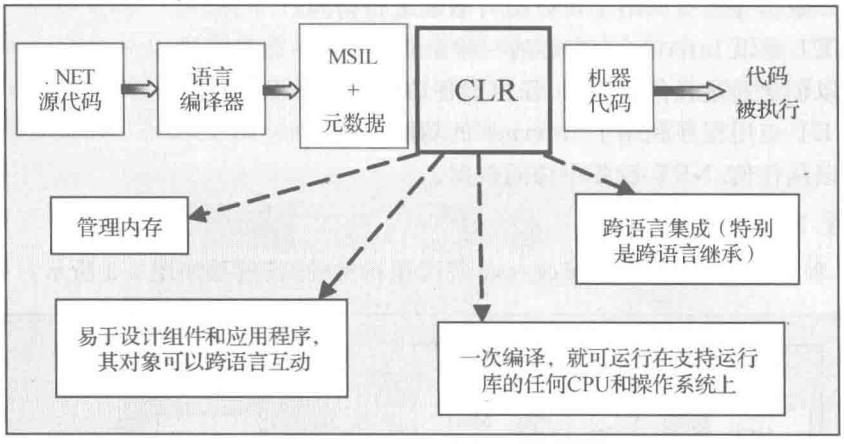


图 1-4　CLR 编译架构

2)CLS

通用语言规范(common language specification,CLS)是 CTS 的一个子集,所有.NET 语言都支持它。任何使用 CLS 兼容类型的程序,可以和以任何语言编写的.NET 程序进行互操作。它规定所有 .NET 语言都应遵循的规则,生成可与其他语言互操作的应用程序。

3)CTS

公共类型系统(common type system,CTS)是.NET 运行时能够理解的一套类型,因此,.NET 应用程序都可以使用它们。应注意,并不是所有.NET 语言都支持 CTS 中所有类型的。CTS 是 CLS 的一个超集,包含标准数据类型和准则集。

CLS、CTS 和 MSIL 紧密配合以实现语言互操作性,理论上,它允许在不同的.NET 语言之间进行非常紧密的互操作,如允许一个 C#类继承自一个由 VB 开发的类。

1.1.3　面向.NET 的全新开发语言——C#

.NET 平台支持的语言主要有 C#、VB.NET、VC++、JScript.NET、ADA、Python、COBOL、Eiffel、SmallTalk 等,其中最重要和使用最广的是 C#语言。

1)C#的诞生

C 和 C++一直是最有生命力的编程语言,这两种语言的优点是提供了强大的功能,具有高度的灵活性以及完整的底层控制能力;缺点是开发周期较长。许多开发效率更高的语言,如 Visual Basic,在功能方面又具有局限性。于是,在选择开发语言时,许多程序设计人员面临着两难的抉择。

针对这个问题,微软公司发布了称为 C#(C Sharp)的编程语言。C#是为.NET 平台量身定做的开发语言,采用面向对象的思想,支持.NET 最丰富的基本类库资源。C#提供快捷的开发方式,同时保留了 C、C++强大的控制能力。C#与 C、C++非常相似,熟悉 C 和 C++的程序设计人员能够很快掌握 C#。

C#是纯面向对象的编程语言,它具有简洁、严谨、表现力强的特点。很多人将 C#语言

看做是 Pascal 和 Java 的混合产品，因为 C# 的创始人 Anders Hejlsberg 曾为 Borland 公司创建了 Pascal 语言，在微软，他又从事了很长时间的 J++ 的研发工作。C# 具有 Pascal 语言的严谨和 Java 语言的简洁，因此一经推出，便得到广大开发人员的认可。

C# 是与 .NET 框架的完美结合，在 .NET 类库的支持下，C# 能够全面地体现 .NET Framework 的各种优点，具体如下：

(1)语法简洁。

(2)彻底的面向对象设计。

(3)与 Web 应用紧密结合。

(4)强大的安全机制。

(5)完善的错误、异常处理机制。

(6)灵活的版本处理技术。

(7)兼容性。

某权威杂志评论：如果抛开一切非技术方面的因素，C# 无疑是这个星球上有史以来最好的编程语言，它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：

(1)面向对象。

(2)类型安全(不建议使用指针、变量初始化等)。

(3)组件技术。

(4)自动内存管理。

(5)跨平台异常处理。

2)C# 与 Java

C# 的语法要比 Java 强大，因为 C# 支持运算符重载和类型安全的枚举；另外，如果需要，还可以在 C# 代码中选择嵌入式指针和其他不合法的语法，只要把它们放在“非安全”的代码块中即可。

C# 可以与其他 .NET 语言编写的代码进行无缝的交互操作，IT 部门不需要标准化 C#，就可以在工程中使用它。

.NET 基类为 C# 提供了一个统一的、标准的源，以满足常用功能的需要，如 XML、互联网和图形化。为了访问相同的功能，Java 程序设计人员有时必须从各种不同的源中获取。

1.2　公共语言运行库

公共语言运行库(CLR)主要负责托管代码的编译和运行。在 .NET 中，代码的编译分为两个阶段：

(1)把源代码编译为 Microsoft 中间语言(MSIL)。

(2)CLR 把 MSIL 编译为平台专用的代码。

在 CLR 的控制下运行的代码是托管代码。托管代码的优点如下：

(1)平台无关性。源代码先编译成中间语言，运行时由 CLR 将中间语言编译成平台专用的代码，跟 Java 的字节代码一样，这样即可实现平台无关性。

(2)提高性能。首先,MSIL 比 Java 的字节码作用还要大,因为 MSIL 是即时编译的,而 Java 的字节码常常是解释性的,在转换为平台可执行代码时可能会导致性能损失。其次,.NET的即时(JIT)编译器并非一次把全部代码编译完才执行,而是只编译调用的那部分代码,并把得到的这部分内部可执行代码保存起来,下次需要调用的时候无须重新编译。Microsoft 认为这个过程要比一开始编译整个应用程序代码的效率高得多,因为任何程序的大部分代码实际上并不是在每次运行过程中都执行。最后,传统的编译器会优化代码,但它们的优化过程是独立于代码所运行的特定处理器的。例如,Visual Studio 2010 优化了一台一般的 Pentium 机器,它所生成的代码就不能利用 Pentium Ⅲ 处理器的硬件特性。而 JIT 编译器与平台无关,所以它可以针对不同的机器完成不同的优化。

(3)语言的互操作性。互操作性是指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。在.NET 中可以交互操作的语言有C#、VB.NET、VC++、JScript、COM 和 COM+。

1.3 微软中间语言

在编译.NET Framework 类库的代码时,不是立即创建操作系统特定的本机代码,而是把代码编译为 Microsoft 中间语言(Microsoft Intermediate Language,MSIL)代码。这些代码不专用于任何一种操作系统,也不专用于 C#。其他.NET 语言,如 Visual Basic .NET 也可以在第 1 阶段编译为这种语言,当使用 Visual Studio 2010 开发 C#应用程序时,编译过程就由 Visual Studio 2010 完成。其原理相当于图 1-5 所示的过程。

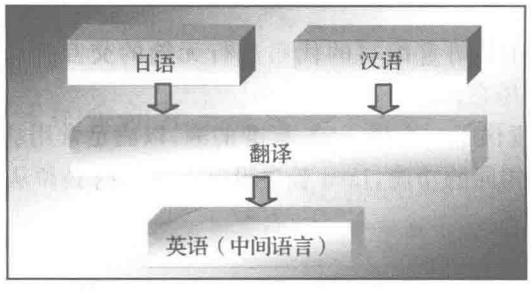


图 1-5 与 MSIL 的多语言"翻译"过程类似的翻译过程

显然,要执行应用程序必须完成更多的工作,这是 JIT(Just-In-Time)编译器的任务。它把 MSIL 编译为专用于 OS 和目标机器结构的本机代码,这样 OS 才能执行应用程序。这里编译器的名称 Just-In-Time 反映了 MSIL 仅在需要时才编译的事实。

过去,常常需要把代码编译为几个应用程序,每个应用程序都用于特定的操作系统和 CPU 结构。这通常是一种优化形式(如为了让代码在 AMD 芯片上运行得更快),并且有时是非常重要的(如对于工作在 Windows 9x 和 Windows NT/2000 环境下的应用程序)。现在就不必要了,因为 JIT 编译器使用 MSIL 代码,而 MSIL 代码是独立于机器、操作系统和 CPU 的。目前有几种 JIT 编译器,每种编译器都用于不同的结构,我们总能找到一个合适

的编译器创建所需的本机代码。

这样，用户需要做的工作就比较少了。程序设计人员可以不再考虑与系统相关的细节，而把注意力放在代码的功能上。

.NET 采用特殊的方式编译和执行程序，先通过编译器编译，程序会被编译成微软中间语言(MSIL)文件，待需要时会启动 MSIL 编译器将 MSIL 编译成机器码，然后加载 CPU 执行。.NET 源代码的执行过程如图 1-6 所示。

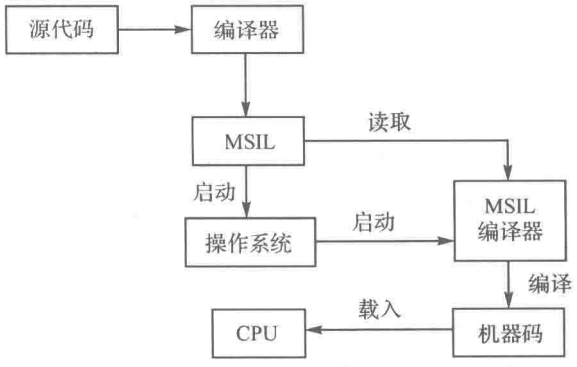


图 1-6　.NET 源代码的执行过程

MSIL 在.NET Framework 中有非常重要的作用，所有面向.NET 的语言都要先编译成 MSIL，那么它们在逻辑上都需要支持 MSIL 的主要特征。MSIL 的主要特征有：

(1)面向对象和使用接口。

(2)值类型和引用类型之间的巨大差别。

(3)强数据类型。

(4)使用异常来处理错误。

(5)使用特性。

下面详细讨论前 3 项主要特性。

1)面向对象和对接口的支持

Microsoft 为 MSIL 选择的特定道路是传统的面向对象的编程，带有类的单一继承性。此外，MSIL 还引进了接口的概念。.NET 接口与 COM 接口不同，它们不需要支持任何 COM 基础结构，例如，它们不是派生自 IUnknown，也没有 GUID。但是它们与 COM 接口共享下述理念：提供一个契约，实现给定接口的类必须提供该接口指定的方法和属性的实现方式。

2)值类型与引用类型

与其他编程语言一样，MSIL 提供了许多预定义的基本数据类型。MSIL 的一个特征就是值类型和引用类型有明显的区别。对于值类型，变量直接保存其数据，在堆栈上面分配存储空间；而对于引用类型，变量保存数据的地址，在堆上分配数据存储空间。

3)强数据类型

MSIL 的一个重要方面是它基于强数据类型，即所有的变量都清晰地标记为属于某个特定数据类型。MSIL 一般不允许对模糊的数据类型执行任何操作。