21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C#程序设计及应用教程（第2版）

C# Programming and Applications (2nd Edition)

马骏 主编　邓居英 杨湖 副主编

- 内容全面：C#程序设计 + Windows程序设计+ Web程序设计
- 实用性强：应用实例 + 使用技巧 + 每章实践练习题
- 配套丰富：PPT + 实例源程序代码 + 习题参考答案

精品系列

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C#程序设计及应用教程（第2版）

C# Programming and Applications (2nd Edition)

马骏 主编　邓居英 杨湖 副主编

精品系列

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

C#程序设计及应用教程 / 马骏主编. —2版. —北京：人民邮电出版社，2009.5
21世纪高等学校计算机规划教材
ISBN 978-7-115-19825-9

Ⅰ. C… Ⅱ. 马… 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（2009）第033656号

内 容 提 要

本书详细介绍 C#程序设计及其应用技术。全书由 C#程序设计基础、Windows 应用程序设计和 Web 应用程序设计三部分组成，主要内容包括 C#语言基础知识、面向对象的编程技术、Windows 窗体控件、目录与文件管理、图形图像处理、ADO.NET、报表设计、类库与控件库设计以及 Web 应用程序开发。

本书可作为高等院校计算机及相关专业的教材，也可作为初、中级程序员的参考用书。

21 世纪高等学校计算机规划教材

C#程序设计及应用教程（第 2 版）

◆ 主　　编　马　骏
副 主 编　邓居英　杨　湖
责任编辑　邹文波

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：22.5
字数：588 千字　　2009 年 5 月第 2 版
印数：12 001 – 15 000 册　　2009 年 5 月北京第 1 次印刷

ISBN 978-7-115-19825-9/TP

定价：35.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

出版者的话

计算机应用能力已经成为社会各行业从业人员最重要的工作技能要求之一，而计算机教材质量的好坏会直接影响人才素质的培养。目前，计算机教材出版市场百花争艳，品种急剧增多，要从林林总总的教材中挑选一本适合课程设置要求、满足教学实际需要的教材，难度越来越大。

人民邮电出版社作为一家以计算机、通信、电子信息类图书与教材出版为主的科技教育类出版社，在计算机教材领域已经出版了多套计算机系列教材。在各套系列教材中涌现出了一批被广大一线授课教师选用、深受广大师生好评的优秀教材。老师们希望我社能有更多的优秀教材集中地呈现在老师和读者面前，为此我社组织了这套“21 世纪高等学校计算机规划教材——精品系列”。

本套教材具有下列特点。

（1）前期调研充分，适合实际教学需要。本套教材主要面向普通本科院校的学生编写，在内容深度、系统结构、案例选择、编写方法等方面进行了深入细致的调研，目的是在教材编写之前充分了解实际教学的需要。

（2）编写目标明确，读者对象针对性强。每一本教材在编写之前都明确了该教材的读者对象和适用范围，即明确面向的读者是计算机专业、非计算机理工类专业还是文科类专业的学生，尽量符合目前普通高等教育计算机课程的教学计划、教学大纲以及发展趋势。

（3）精选作者，保证质量。本套教材的作者，既有来自院校的一线授课老师，也有来自 IT 企业、科研机构等单位的资深技术人员。通过他们的合作使老师丰富的实际教学经验与技术人员丰富的实践工程经验相融合，为广大师生编写出适合目前教学实际需求、满足学校新时期人才培养模式的高质量教材。

（4）一纲多本，适应面宽。在本套教材中，我们根据目前教学的实际情况，做到“一纲多本”，即根据院校已学课程和后续课程的不同开设情况，为同一科目提供不同类型的教材。

（5）突出能力培养，适应人才市场要求。本套教材贴近市场对于计算机人才的能力要求，注重理论知识与实际应用的结合，注重实际操作和实践动手能力的培养，为学生快速适应企业实际需求做好准备。

（6）配套服务完善。对于每一本教材，我们在教材出版的同时，都将提供完备的 PPT 课件，并根据需要提供书中的源程序代码、习题答案、教学大纲等内容，部分教材还将在作者的配合下，提供疑难解答、教学交流等服务。

在本套教材的策划组织过程中，我们获得了来自清华大学、北京大学、中国人民大学、浙江大学、吉林大学、武汉大学、哈尔滨工业大学、东南大学、四川大学、上海交通大学、西安交通大学、电子科技大学、西安电子科技大学、北京邮电大学、北京林业大学等院校老师的大力支持和帮助，同时获得了来自信息产业部电信研究院、联想、华为、中兴、同方、爱立信、摩托罗拉等企业和科研单位的领导或技术人员的积极配合。在此，向他们表示衷心的感谢。

我们相信，“21 世纪高等学校计算机规划教材——精品系列”一定能够为我国高等院校计算机教学做出应有的贡献。同时，对于工作中的欠缺和不妥之处，欢迎老师和读者提出宝贵的意见和建议。

前 言

C#语言是一种完全面向对象的基于.NET 的编程语言，已先后被欧洲计算机制造商协会和国际标准化组织批准为高级语言开发标准（ECMA-334、ISO/IEC 23270）。随着.NET 技术的普及，C#语言已成为开发基于.NET 的企业级应用程序的首选语言。

本书第 1 版以高度的实用性和通俗易懂的讲解，受到读者的普遍欢迎。

本书在继承第 1 版教材特色的基础上，结合作者多年的教学经验，并特别根据近几年教学改革的实践以及对人才培养的高标准要求，对其内容做了进一步的优化、补充和完善。本书在第 1 版教材的基础上做了较大改进，使各章结构更加紧凑，知识点的介绍也更突出，更适合学生学习，同时增加了一些新的技术。另外，对一些初学者比较容易糊涂的地方，也作了更为详细的阐述。

本书具有以下特色。

1. 语言简洁、通俗易懂

对一些初学者比较难以理解的知识点，尽量用简洁、形象而又浅显易懂的语言来表达。比如泛型、接口、组件、特性、纹理、图形图像用法等。

2. 代码详细、注重实用

对项目开发中比较常用的技术，都用与实际应用非常接近的例子来说明，而且有详细的代码解释。这些例子都是在项目开发中初学者迫切想实现而又不知如何做的内容，使读者学习后能很快明白具体用法，能马上编写出生动实用的应用程序。

3. 循序渐进、由浅入深

章节内容的安排采用循序渐进的办法，对于后面例子中用到的知识点，在前面章节的介绍中都有相应地铺垫，使读者从最简单的知识入手，一步步走入很有“艺术性”的各种“深奥”的编程技术中。

4. 重点介绍、适当扩充

对常用的技术，本书花费了较多的篇幅来介绍，并以图文并茂的形式，将重点内容展现出来。同时，也适当地介绍了一些比较新的编程技术，如 LINQ 等，使读者能够根据介绍的内容，很快对所学知识举一反三、灵活运用。

各高校在教学过程中，可以根据专业课程体系和学期总学时数，选取本书的全部或部分内容讲解，建议各章学时分配如下。

54 学时				72 学时			
第 1 章	2 学时	第 9 章	8 学时	第 1 章	2 学时	第 9 章	10 学时
第 2 章	4 学时	第 10 章	2 学时	第 2 章	6 学时	第 10 章	2 学时
第 3 章	4 学时	第 11 章	2 学时	第 3 章	4 学时	第 11 章	2 学时
第 4 章	4 学时	第 12 章	2 学时	第 4 章	4 学时	第 12 章	4 学时
第 5 章	2 学时	第 13 章	2 学时	第 5 章	4 学时	第 13 章	2 学时
第 6 章	8 学时	第 14 章	6 学时	第 6 章	10 学时	第 14 章	6 学时
第 7 章	2 学时	第 15 章	2 学时	第 7 章	4 学时	第 15 章	6 学时
第 8 章	4 学时			第 8 章	6 学时		

对于讲授课时为54学时的学校，若学时紧张，也可以将第5章和最后一章穿插到其他章节中讲解。

对于讲授课时为72学时的学校，由于第5章属于较高级的内容，学生在基本开发技术没有掌握之前，真正理解并将这些技术应用到实际中可能有一定难度，建议在讲解完其他章节后，再系统讲解这一章的内容，这样效果可能更好。

本书由马骏担任主编，邓居英、杨湖担任副主编，马骏对全书进行了统稿、修改和定稿。参与各章编写和代码调试等工作的还有侯彦娥、王寅、付征叶、乔保军、孙清伟、姚远、张娜等。

为了配合教学需要，本书还提供配套的教学课件、全书所有例题的源程序代码以及所有习题参考解答。读者可以与人民邮电出版社联系索取，或到人民邮电出版社教学服务与资源网（http://ptpedu.com.cn）上下载。

由于编者水平有限，书中难免存在错误之处，敬请读者批评指正。

编　者

2009年2月

目　录

第 1 篇　C#程序设计基础

第 1 篇
C#程序设计基础

C#语言是在 C 和 C++的基础上重新构造的、语法与 C++和 Java 都比较相似的、基于.NET 框架支持的一种完全面向对象的、类型安全的编程语言，也是.NET 的首选编程语言。微软公司设计 C#语言的目的主要是为了简化网络应用编程的难度。从开发效率来讲，C#为应用程序开发人员提供了快速的开发手段，但又保持了 C++语言的特点和优点。从继承性来讲，C#在更高层次上重新实现了 C 和 C++。从语法形式和易用性来讲，C#语言几乎综合了目前流行的所有高级语言的优点，提供了一种语法优雅、功能完善而又容易使用的外在表现形式。

本篇主要介绍 C#语言程序设计基础知识。

为了让读者把注意力集中在对基本知识的理解上，在介绍其他类型的应用程序之前的这些章节中，均使用控制台应用程序作为例子，但是这些语言本身的基本用法同样适用于 Windows 窗体应用程序和 ASP.NET Web 应用程序。

控制台应用程序使用标准命令行输入和输出，而不是在图形窗体上显示。控制台应用程序使用 System 命名空间中的类处理输入和输出。

C#语言的基本语法和面向对象的设计思想是实际应用项目开发的基础，希望读者能很好地理解和掌握。

第1章 概述

C#语言和 Microsoft.NET 框架简化了软件开发的复杂度，利用 C#语言和基于.NET 框架的 Visual Studio 2008（简称 VS2008）集成开发平台，程序员可以非常方便地开发出各种应用程序。

这一章我们对 C#语言、.NET 框架以及 VS2008 作一个基本的了解。

1.1 C#语言及其发展简介

C#（读作“see sharp”）的叫法很有创意，意思是让我们看看这种语言多么锋利无比。C++不是有两个“+”号吗，那就让它有 4 个“+”号吧，这就是“#”的来历。

1.1.1 C#语言的特点

C#语言具有以下主要特点。

（1）简洁的语法

C#语言和 Java 语言一样，使用了统一的操作符，淘汰了 C++语言中乱糟糟的表示符号和伪关键字，使用最简单、最常见的形式进行描述。

（2）精心的面向对象设计

C#语言是完全按照面向对象的思想来设计的，因此，它具有面向对象所应有的一切特性，如封装、继承、多态性等。

在类的继承方面，C#语言只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。

在 C#语言中，每种类型都是一个对象，不存在全局函数、全局变量等概念，所有常量、变量、属性、方法、索引、事件等都必须封装在类中，从而使代码具有更好的可读性，也避免了发生命名冲突的可能。

（3）与 Web 的紧密结合

在 C#语言中，对于复杂的 Web 编程和其他网络编程看起来更像是对本地对象进行操作，从而简化了大规模、深层次的分布式开发。用 C#语言构建的 Web 组件能够方便地作为 Web 服务（Web Service）并可以通过 Internet 被运行在任何操作系统上的任何语言所调用。

（4）可靠的安全性与错误处理

语言的安全性与错误处理能力是衡量一种语言是否优秀的重要依据。C#语言可以消除许多软件开发中的常见错误，并提供了包括类型安全在内的完整的安全性能。

在默认情况下，从 Internet 和 Intranet 下载的代码都不允许访问任何本地文件和资源；C#语言不允许使用未初始化的局部变量，并提供了完善的边界检查与溢出检查等功能。内存管理中的垃圾回收机制也极大地减轻了开发人员对内存管理的负担。

（5）版本处理技术

C#语言内置了版本控制功能，如对方法重载和接口的处理方式以及对特性（Attribute）的支持等，从而保证方便地开发和升级复杂的软件。

（6）灵活性和兼容性

在托管状态下，C#语言不能使用指针，而是用委托（Delegate）来模拟指针的功能。如果确实需要在类或者类的方法中使用指针，只需要声明这些代码为非托管的代码就可以了。另外，虽然 C#语言不支持类的多继承，但是却可以通过接口来实现多继承。

兼容性是指 C#语言允许与 Win32 API（Application Programming Interface）进行交互操作，允许 C#语言组件与其他语言组件间的交互操作等。

1.1.2　C#语言的发展过程

C#语言的发展过程主要经历了以下阶段。

2000 年，C#语言诞生。

2003 年，微软公司发布了 C#语言规范 1.2（简称 C#1.2），VS.NET 2003 使用的是 C#1.2。

2005 年，微软公司发布了 C#语言规范 2.0（简称 C#2.0），VS2005 使用的是 C#2.0。

2007 年，微软公司发布了 C#语言规范 3.0（简称 C#3.0），VS2008 使用的是 C#3.0。

由于 C#还在快速发展，因此版本更新较快。C#2.0 在 C#1.2 的基础上，增加了泛型、迭代、分部类、匿名方法等，C#2.0 版本已经被国际标准化组织定为高级语言开发标准。

C#3.0 在 C#2.0 的基础上，又增加了查询表达式、Lambda 表达式、自动实现的属性、匿名变量、隐式类型的本地变量和数组，以及扩展方法等功能，C#3.0 由于刚推出不久，尚未通过标准化组织审核。

1.1.3　C#语言与其他语言的比较

本小节将 C#语言与目前常用的高级语言做一个简单的比较，以便熟悉这些语言的读者可以更好地了解 C#语言的特性，更快地掌握 C#语言的编程方法。

1. C#语言与 C++语言的比较

C#语言对 C++语言进行了多处改进，现将主要区别介绍如下。

- 编译目标：C++代码直接编译为本地可执行代码，而 C#语言默认编译为中间语言（IL）代码，执行时再通过 JIT（Just-In-Time）编译器将需要的模块临时编译成本地代码。
- 内存管理：C++语言要求程序员显式地删除动态分配给堆的内存，而 C#语言不需要这么做，C#语言采用垃圾回收机制自动在合适的时机回收不再使用的内存。
- 指针：C++语言中大量地使用指针，用起来比较复杂，对不熟练的程序员来说，编写的程序容易引起内存泄露以及管理漏洞；而 C#语言使用对类实例的引用，虽然在非类型安全的状态下 C#也可以使用指针，不过，一般情况下 C#程序员没必要这样做。
- 字符串处理：在 C#语言中，字符串是作为一种基本数据类型来对待的，因此 C#语言比 C++语言中对字符串的处理要简单得多。
- C++语言允许类的多继承，而 C#语言只允许类的单继承，但可以通过接口实现多继承。

在后面的学习中我们会发现，C#语言与C++语言相比还有很多不同和改进之处，包括一些细节上的差别，这里就不一一列举了。

2. C#语言与Java语言的比较

从语法上讲，C#语言和Java语言非常相似，只是在细节上有一些差别。实际上，C#语言和Java语言的主要差别不是在语言本身，而是在内部功能实现上以及性能上不一样。

Java程序需要一个运行环境JRE(Java Runtime Environment)来执行代码，但JRE只限于在Java这一门语言中使用。C#语言也需要一个运行环境CLR（Common Language Runtime），但是CLR提供了对支持.NET框架的所有语言的支持。

Java源代码可以被编译成字节代码的一种中间状态，然后由已提供的虚拟机来执行这些字节代码。C#代码也被编译成一种中间状态，称为中间语言（IL），但是IL代码则被传输到由CLR管理的执行进程上，然后通过CLR的JIT编译器编译成本地代码执行。

C#语言与Java语言相比也有很多不同和改进之处。例如，C#语言的文件名不受文件中类名的限制，而在Java语言中则有此限制。另外，C#语言也提供了一些在Java语言中没有的功能，如运算符重载、装箱和拆箱、结构以及方法隐藏等。

表1-1所示为C#语言和Java语言的比较。

表1-1　　C#语言和Java语言的比较

项　目	C#	Java
运行环境	可在具有CLR的平台上运行	可在具有JVM的平台上运行
完全面向对象	是	是
多重继承	不支持	不支持
内存管理	使用垃圾回收机制管理内存	使用垃圾回收机制管理内存
异常处理	try-catch-finally	try-catch-finally
指针	只能在非托管的代码段内使用	不支持
类型安全性验证	强制类型验证	强制类型验证
命名空间	支持	支持
布尔值	只能为true或者false	只能为true或者false
变量初始化	不能使用未初始化的变量	不能使用未初始化的变量
中间语言处理	将MSIL转换为JIT机器码	Java字节码
访问修饰符	public、protected、private、internal、partial	public、protected、private、friendly
Web Service	支持	支持

3. C#语言与VB.NET语言的比较

和C#语言一样，VB.NET语言也是基于.NET Framework和CLR的高级语言。但是C#语言有一些VB.NET语言所不具备的独有的特性，如C#语言可以使用非托管代码、移位操作符、内嵌的文档（XML）和运算符重载等。在发展前景上，由于C#语言一开始就是完全按照面向对象的思想来设计的，而且它使用的全部是.NET框架定义的语法格式，不存在考虑与.NET之前版本兼容的问题，因此给人的感觉是结构清晰，语法简洁、优雅。另外，C#语言作为一种高级语言标准，其基本的内部实现形式是公开的，因此更容易被多种平台（例如UNIX、Linux等）接受和广泛应用。当然，VB.NET语言也有其自身的优点，如以前学习过VB的人既可以使用.NET之前的语法格式及函数，也可以使用.NET规定的语法格式和面向对象的设计思想。

1.2　Visual Studio 开发环境的功能和特点

基于.NET 的 Microsoft Visual Studio 是.NET 平台下非常强大的软件开发工具，无论是软件服务商，还是企业级应用程序的部署与发布，Visual Studio 开发平台都可以提供近乎完美的解决方案。

在 Visual Studio 开发环境下，提供了一整套的软件开发和测试工具，包括设计、编码、编译调试、与数据库的互操作等基本功能和基于开放架构的服务器组件开发平台、企业开发工具和应用程序重新发布工具以及性能评测报告等高级功能。

表 1-2 所示为基于.NET 的 Visual Studio 系列开发平台提供的编程工具及特点。

表 1-2　　基于.NET 的 Visual Studio 系列开发平台提供的编程工具及特点

功　能	描　述
Windows 窗体设计器	提供图形化设计界面，通过对窗体中的控件进行拖放，可以快速创建应用程序的用户界面
Windows 窗体工具	提供了 Windows 窗体设计器、Windows 应用程序模板、基本的项目引用和初始代码，以帮助用户创建标准 Windows 窗体应用程序
Web 窗体工具	提供了 Web 窗体设计器、ASP .NET Web 应用程序模板、基本的项目引用和初始代码，以帮助用户创建以浏览器作为主界面的 Web 窗体应用程序
XML Web Services 工具	提供了一个 ASP .NET Web Service 模板，可以用来构成 Web 应用程序的基本结构。此时，Web 应用程序的基本架构将构建在 Web 服务器和本地解决方案文件之上
多语言支持	开发环境集成了所有的.NET 平台编程语言，包括 Visual C#语言
数据访问	包括用于创建数据库应用程序的组件、可视化数据库工具以及一个可靠的 ADO.NET 类集，使开发人员可以方便地操作所有类型的数据
错误处理	包括支持跨语言调试的调试工具以及结构化异常类
向导	可以用来快速创建复杂的通用任务

基于.NET 的 Visual Studio 开发平台主要经历了以下阶段。

2003 年，微软公司发布了 Visual Studio.NET 2003（简称 VS.NET 2003）和.NET Framework 1.1。

2005 年，微软公司发布了 Visual Studio 2005（简称 VS2005）和.NET Framework 2.0。

2006 年，微软公司发布了.NET Framework 3.0。

2007 年，微软公司发布了 Visual Studio 2008（简称 VS2008）以及.NET Framework 3.5。

2008 年，微软公司发布了 Visual Studio 2008 SP1 以及.NET Framework 3.5 SP1。

对 Visual Studio 的每一个版本来说，与它的前一个版本相比，都有非常多的改进。下面列出 Visual Studio2008 与 Visual Studio2005 相比，我们感兴趣的几个方面。

- Visual Studio2005 不支持 SQL Server 2008，而 Visual Studio2008 SP1 开始全面支持 SQL Server 2008。
- Visual Studio2008 在 Web 应用程序开发方面与 Visual Studio2005 相比变化很大，如新的设计视图和 CSS 设计工具、JavaScript 的单步调试和智能提示、内置的 AJAX 等。
- Visual Studio2008 在 Windows 应用程序开发方面提供了更多的功能，如专为 P2P 提供的类、支持 XNA3.0、智能客户端发布更简单等。

Visual Studio2008 主要有以下版本。

（1）Visual Studio 2008 速成版（Express Edition）：该版本为免费版本，功能有限，适合初学者学习使用。

（2）Visual Studio 2008 标准版（Standard Edition）：标准版适合一般软件公司。

（3）Visual Studio 2008 专业版（Professional Edition）：该版本功能完善，适合专业软件公司。

（4）Visual Studio 2008 团队版（Team Edition）：团队版在 Visual Studio 2008 Professional Edition 的基础上提供了更高级的开发工具。

除了上面几种版本外，还有团队套装版（Team Suite）、团队开发服务器版等。

本书的所有例子均在安装了 SP1 的 Visual Studio 2008 Professional Edition 开发环境下调试通过。

1.2.1 Microsoft.NET Framework

Microsoft.NET Framework（简称.NET 框架）是生成、运行.NET 应用程序和 Web Service 的组件库，它包括两个主要组件，一个是公共语言运行库（简称运行库），另一个是类库。运行库提供.NET 应用程序所需要的核心服务，类库为开发和运行.NET 应用程序提供了各种支持。

图 1-1 所示为公共语言运行库和类库与应用程序及其他系统之间的关系。

.NET 框架实现的目标如下。

- 提供一个对各种语言都一致的面向对象的编程环境。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可提高代码执行安全性的代码执行环境（包括托管代码和非托管代码）。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的开发经验在 Windows 应用程序和 Web 应用程序中保持一致。
- 按照工业标准生成所有通信，以确保基于.NET 框架的代码可与任何其他代码集成。

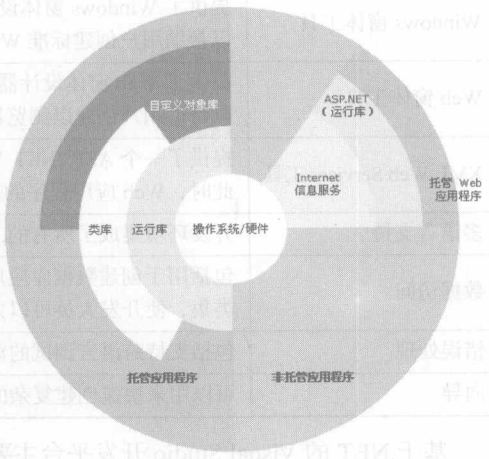

图 1-1 .NET Framework 环境

用基于.NET 框架开发的应用程序，不论使用的是哪种高级语言，均必须在安装了.NET 框架的计算机上才能运行。这种架构与 Java 应用程序必须由 Java 虚拟机支持相似。

目前，除了基于.NET 框架的 C#语言、VB.NET 语言、C++.NET 语言以及和 Java 语言语法完全相同的 J#语言外，还有基于.NET 框架的 FORTRAN 语言、Pascal 语言、COBOL 语言、PERL 语言、Python 语言、Eiffel 语言等其他高级语言。

1. 运行库

运行库是.NET 框架的基础，可被看作是一个在执行时管理代码的代理，提供诸如内存管理、线程管理、远程处理等核心服务，而且还强制实施严格的类型安全以及确保安全性和可靠性的其他形式的代码的准确性。

以运行库为目标的代码称为托管代码，不以运行库为目标的代码称为非托管代码。

使用.NET 框架提供的编译器可以直接将源程序编译为.EXE 或者.DLL 文件，但是需要注意的是，此时编译出来的程序代码并不是 CPU 能直接执行的机器代码，而是一种中间语言 IL（Intermediate Language）代码。