

■ 大学软件学院软件开发系列教材

C#

程序开发实用教程

马晓波 主 编
王国胜 张石磊 副主编

- ◆ 注重基础知识及案例的讲解。
- ◆ 教材示例源文件丰富，实例具有普遍性和实用性。
- ◆ 每章都有实训测试题和课后习题，便于巩固学习成果。
- ◆ 教材有配套的电子教案及习题答案，方便老师教学时使用。

免费赠送
PPT电子课件
及习题答案

清华大学出版社

大学软件学院软件开发系列教材

C#程序开发实用教程

马晓波 主 编

王国胜 张石磊 副主编

清华大学出版社

北 京

内 容 简 介

本书共分 14 章,由浅入深地介绍了 Visual C# .NET 基础、面向对象基础编程技术、Visual C# .NET 控件及其应用、图形图像和多媒体编程、网络程序开发、文件处理和系统操作、数据库编程、Web 应用程序开发、WPF 应用程序开发、XML 技术等内容,最后通过两个实例(图像处理程序实例和 B/S 型网上考试系统)对前面的知识进行综合应用。

本书结构合理,内容全面,非常适合各大院校作为教材使用,同时也是 C#初学者一本不可多得的学习用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C#程序开发实用教程/马晓波主编;王国胜,张石磊副主编. --北京:清华大学出版社,2013
(大学软件学院软件开发系列教材)

ISBN 978-7-302-31766-1

I. ①C… II. ①马… ②王… ③张… III. ①C 语言—程序教材—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 057826 号

责任编辑:杨作梅

装帧设计:杨玉兰

责任校对:宋延清

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>, 010-62791865

印 刷 者:清华大学印刷厂

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:185mm×260mm

印 张:25.75

字 数:619 千字

版 次:2013 年 4 月第 1 版

印 次:2013 年 4 月第 1 次印刷

印 数:1~4000

定 价:49.00 元

产品编号:045182-01

前 言

微软的 C#(Visual Studio 2010 版)是从 C/C++演变而来的一种新的面向对象的程序设计语言,它结合 ASP.NET 或 Windows Phone 平台开发应用程序,具有简洁、灵活、安全、面向对象、兼容性强等特点。C#代表了当前的编程方向,因此,受到了广大开发者的喜爱和追捧。

近年来,随着高校课程改革的逐渐推进和教学模式的不断优化,有些学校率先面向非计算机专业学生开设了 Visual C#教学课程,把 C#作为大学生程序设计的入门语言。这其中还有一个更重要的原因就是:从程序设计基础教学角度来讲,Visual C#语言具有简单易学和能快速开发等优点,非常适合学生学习。

在编写本书时,考虑大部分读者为初学者,所以在结构体系上采用由浅入深、循序渐进的方式。在讲解基础理论时,以实际应用为主,力求帮助读者不断地巩固和检验所学知识,提高编程能力。

本书共分为 14 章,从 Visual C# .NET 基础知识讲起,逐一面对面向对象基础编程技术、Visual C# .NET 控件及其应用、图形图像和多媒体编程、网络程序开发、文件处理和系统操作、数据库编程、Web 应用程序开发、WPF 应用程序开发、XML 技术等内容做详细的讲解。最后通过两个综合开发实例(图像处理程序实例和 B/S 型网上考试系统)对前面的知识进行巩固和应用。

本书一改以往图书单纯理论讲解的局面,在书中列举了大量的实例进行介绍。同时,书中还对程序代码做了全面的注释,旨在帮助读者更好地掌握编程的要领,并进入实际开发角色。书中所有实例及上机实训,均是多年来理论与实践教学中归纳总结出来的,是针对每章教程的重点、难点而编写的。通过学习本书,可以加深对理论基础的理解,并在短时间内轻松掌握编程技术。

本书不仅适合应用型高校计算机专业学生的专业课程教学,还适用于其他各专业学生的公共课程教学。为了满足教学需求,官方网站(www.wenyuan.com.cn)专门为广大教师提供了 PPT 课件、书中实例的源文件以及习题答案等内容。

本书由马晓波、王国胜、张石磊三位老师编写,其中第 1~7 章由马晓波老师编写;第 8~11 章由王国胜老师编写,第 12~14 章由张石磊老师编写。此外,陈丽丽、张丽、刘攀攀、付中举、王程、王赞赞、王亚坤、周杰、胡文华、胡娜、尼春雨、尼朋、李风云、蒋军军等人也参与了本书的校对和代码调试工作,在此一并表示感谢。

由于编者水平有限,书中难免有错误和不足之处,敬请批评指正。

编 者
2013 年 4 月

目 录

第 1 章 Visual C# .NET 简介	1	3.1.3 运算符和表达式	39
1.1 .NET 框架概述	2	3.1.4 结构化控制语句	40
1.1.1 公共语言运行库	3	3.1.5 数组	41
1.1.2 类库	4	3.1.6 结构体	42
1.1.3 .NET 程序的编译与运行	5	3.1.7 函数	42
1.1.4 .NET 框架与 J2EE	7	3.2 类	43
1.2 Visual Studio 2010 .NET 集成环境	7	3.2.1 声明一个类	43
1.2.1 Visual Studio 2010 概述	7	3.2.2 类成员的声明和访问	45
1.2.2 Visual Studio 2010 的新特性	8	3.3 对象	46
1.2.3 Visual Studio 2010 的窗口	9	3.3.1 对象的声明和访问	46
1.3 Visual C# .NET 的三种应用程序 结构	10	3.3.2 对象和类的关系	47
1.3.1 以 Visual C# .NET 编写 控制台应用程序	10	3.4 方法	47
1.3.2 以 Visual C# .NET 编写 Windows 应用程序	13	3.4.1 方法的声明和参数	47
1.3.3 以 Visual C# .NET 编写 ASP.NET Web 应用程序	18	3.4.2 静态方法和非静态方法	49
1.4 上机实训	22	3.4.3 重载方法	50
1.5 本章习题	23	3.4.4 析构函数	52
		3.4.5 构造函数	52
第 2 章 我的第一个 C# 程序	25	3.5 属性	54
2.1 Hello World 程序	26	3.6 上机实训	56
2.2 编译和运行程序	27	3.7 本章习题	56
2.2.1 在 Visual Studio 2010 中 编译和运行	28	第 4 章 面向对象高级编程技术	59
2.2.2 通过命令行编译和运行	29	4.1 封装技术	60
2.3 程序详解	30	4.1.1 封装的作用	60
2.4 上机实训	31	4.1.2 如何实现封装	60
2.5 本章习题	32	4.2 多态技术	62
第 3 章 面向对象基础编程知识	33	4.2.1 多态的作用	62
3.1 基础编程知识	34	4.2.2 如何实现多态	63
3.1.1 关键字及标识符	34	4.3 继承技术	64
3.1.2 数据类型	36	4.3.1 继承的作用	64
		4.3.2 如何实现继承	65
		4.4 抽象类和抽象方法	66
		4.4.1 抽象类与抽象方法概述	66
		4.4.2 使用抽象类和抽象方法	66

4.5 异常处理.....	67	5.8 对话框控件应用.....	106
4.5.1 何为异常处理.....	67	5.8.1 打开文件对话框控件.....	106
4.5.2 抛出和捕获异常.....	68	5.8.2 保存文件对话框控件.....	107
4.5.3 预定义异常.....	69	5.8.3 颜色对话框控件.....	108
4.5.4 使用 throw 语句处理异常.....	69	5.8.4 字体对话框控件.....	108
4.5.5 使用 try-catch-finally 语句 处理异常.....	71	5.8.5 PrintDialog 控件和 PrintDocument 控件.....	109
4.5.6 自定义异常.....	72	5.8.6 对话框控件应用实例开发.....	110
4.6 域.....	73	5.9 多窗体设计.....	120
4.7 上机实训.....	74	5.9.1 添加窗体的方法.....	121
4.8 本章习题.....	74	5.9.2 多窗体应用实例.....	122
第 5 章 Visual C# .NET 控件及其应用.....	75	5.10 上机实训.....	124
5.1 文本框应用.....	76	5.11 本章习题.....	125
5.1.1 常用属性和事件.....	76	第 6 章 图形图像和多媒体编程.....	127
5.1.2 文本控件应用实例.....	77	6.1 GDI+编程.....	128
5.1.3 按钮控件.....	81	6.1.1 Graphics 对象和画笔画刷.....	128
5.1.4 标签控件.....	81	6.1.2 在窗体上显示直线、矩形、 椭圆.....	134
5.2 单选按钮应用.....	82	6.1.3 图像进行平移、旋转与 缩放.....	137
5.2.1 常用属性和事件.....	82	6.1.4 绘制象棋棋盘.....	138
5.2.2 单选按钮控件应用实例.....	83	6.1.5 可擦写图形轮廓的实现.....	141
5.3 复选框应用.....	86	6.2 截图操作.....	143
5.3.1 常用属性和事件.....	86	6.2.1 固定大小的截图.....	143
5.3.2 复选框控件应用实例.....	87	6.2.2 任意大小的截图.....	144
5.4 列表框应用.....	90	6.2.3 图片的拖动.....	146
5.4.1 常用属性和事件.....	90	6.3 图像显示与保存.....	148
5.4.2 列表框控件应用实例.....	91	6.3.1 利用 Bitmap 类的对象显示 图像文件.....	148
5.4.3 Random 类的使用.....	95	6.3.2 保存到图像文件.....	149
5.4.4 关于随机 System.Random 类的 应用技巧.....	96	6.3.3 实例开发——图片的显示和 保存.....	149
5.5 组合框应用.....	96	6.4 托盘程序.....	152
5.5.1 常用属性和事件.....	96	6.4.1 托盘程序的关键技术.....	152
5.5.2 组合框控件应用实例.....	97	6.4.2 开发托盘程序.....	152
5.6 Timer 控件用法.....	100	6.5 定时抓取当前程序窗口.....	154
5.6.1 常用属性和事件.....	100	6.6 特殊形状的窗体界面.....	156
5.6.2 Timer 控件应用实例.....	101	6.6.1 定制不规则窗体时使用的 类库.....	156
5.7 图片框应用.....	103		
5.7.1 常用属性和事件.....	103		
5.7.2 PictureBox 控件应用实例.....	103		



6.6.2 使用类库定制不规则窗体.....	157	8.1.1 目录管理.....	190
6.6.3 使用位图的不规则窗体.....	158	8.1.2 文件管理.....	191
6.7 生成数字字符验证码图片.....	161	8.1.3 文件夹浏览器实现.....	194
6.7.1 设计原理.....	161	8.2 文件的读写.....	201
6.7.2 功能实现.....	162	8.2.1 FileStream 类读写文件.....	201
6.8 播放声音与视频的文件.....	164	8.2.2 文本文件的读写.....	203
6.8.1 通过 API 函数播放声音 文件.....	164	8.2.3 二进制文件的读写.....	205
6.8.2 ActiveX 控件.....	165	8.3 Visual C# .NET 调用 Windows API 函数.....	207
6.8.3 Windows Media Player 控件 播放声音和视频文件.....	165	8.3.1 Windows API 函数.....	207
6.8.4 无声动画控件(Animation).....	166	8.3.2 调用 Windows API 的 基本过程.....	208
6.9 上机实训.....	167	8.4 注册表编程.....	209
6.10 本章习题.....	167	8.5 处理鼠标和键盘事件.....	210
第 7 章 网络程序开发.....	169	8.5.1 处理鼠标相关的事件.....	210
7.1 网络通信编程基础.....	170	8.5.2 处理键盘相关的事件.....	211
7.1.1 Socket 套接字简介.....	170	8.6 上机实训.....	212
7.1.2 TCP 协议和 UDP 协议.....	170	8.7 本章习题.....	212
7.1.3 Socket 编程原理.....	171	第 9 章 数据库编程.....	215
7.1.4 套接字 Socket 类编程.....	172	9.1 ADO.NET 数据库访问技术.....	216
7.1.5 .NET 框架中网络通信的 应用层类.....	178	9.1.1 Connection 对象.....	219
7.2 使用 TcpClient 类和 TcpListener 类 实现 TCP 协议通信.....	178	9.1.2 Command 对象.....	220
7.2.1 TcpClient 类和 TcpListener 类.....	178	9.1.3 DataReader 对象.....	220
7.2.2 实现基于 TCP 协议的 局域网通信程序.....	181	9.1.4 DataAdapter 对象.....	221
7.3 使用 UdpClient 类实现 UDP 协议 编程.....	185	9.1.5 DataSet 对象.....	222
7.3.1 UdpClient 类.....	185	9.2 SQL Server 2008 使用基础.....	223
7.3.2 UdpClient 类开发 UDP 程序的 过程.....	187	9.3 简单数据表操作方法.....	233
7.4 上机实训.....	187	9.3.1 利用向导进行数据处理.....	233
7.5 本章习题.....	188	9.3.2 在保持连接的方式下进行 数据操作.....	237
第 8 章 文件处理和系统操作.....	189	9.3.3 在无状态方式下进行 数据操作.....	239
8.1 文件夹浏览器设计.....	190	9.4 使用 DataReader 顺序读取.....	240
		9.5 数据库中的图像存取.....	242
		9.6 上机实训.....	245
		9.7 本章习题.....	245

第 10 章 Web 应用程序开发	247
10.1 Web 窗体与 ASP.NET 内置对象	248
10.1.1 ASP.NET 工作原理	248
10.1.2 Web 窗体页面的结构	248
10.1.3 ASP.NET 的常用内置对象	249
10.2 ASP.NET 常用控件	254
10.2.1 ASP.NET 控件分类	255
10.2.2 文本输入控件	256
10.2.3 复选框和单选按钮	256
10.2.4 Button、ImageButton 和 LinkButton 控件	257
10.2.5 Image 控件和 ImageMap 控件	258
10.2.6 DropDownList 控件	258
10.2.7 Calendar 控件	259
10.2.8 视图控件	260
10.2.9 验证控件	262
10.3 ADO.NET 对数据库的操作	264
10.3.1 连接两种数据库	264
10.3.2 读取数据库	264
10.3.3 数据的添加、删除和修改	265
10.4 数据库操作控件	265
10.4.1 数据源控件	265
10.4.2 数据显示控件 GridView	268
10.5 上机实训	271
10.6 本章习题	272
第 11 章 WPF 应用程序开发	273
11.1 WPF 简介	274
11.2 创建 WPF 应用程序	275
11.3 WPF 应用程序实例	280
11.4 上机实训	289
11.5 本章习题	290
第 12 章 XML 技术	291
12.1 XML 的概念	292
12.1.1 使用 XML 的原因	292
12.1.2 XML 的文件规范	293
12.1.3 XML 的显示技术	296
12.1.4 使用 DOM 访问 XML 文档	297
12.2 用 .NET 框架中的类操作 XML 文件	299
12.2.1 与 XML 有关的命名空间和相关类	299
12.2.2 对 XML 文件编辑操作	303
12.3 使用 ADO.NET 中的 DataSet 进行 XML 文件的创建和读写	309
12.3.1 使用 DataSet 创建 XML 文件	309
12.3.2 使用 DataSet 读取 XML 文件	311
12.4 基于 XML 的 Web 留言板设计	313
12.5 上机实训	320
12.6 本章习题	320
第 13 章 图像处理程序实例	323
13.1 图像处理基础	324
13.2 图像处理技术	333
13.2.1 黑白处理	333
13.2.2 逆反处理	334
13.2.3 平滑处理	336
13.2.4 霓虹处理	337
13.2.5 锐化处理	338
13.2.6 浮雕处理	340
13.2.7 图像雾化	341
13.2.8 底片处理	342
13.2.9 平移变换	343
13.2.10 旋转变换	344
13.2.11 比例变换	345
13.2.12 查看缩略图	346
第 14 章 B/S 型网上考试系统设计	349
14.1 功能需求分析	350
14.2 系统设计	351
14.3 功能模块实现	356
14.3.1 公共模块	356

附录 课后习题参考答案 399

C#
面向对象

C# 面向对象
面向对象

第 1 章



Visual C# .NET 简介

学习目的与要求：

C#(C Sharp)是由美国微软公司开发的一种面向对象的、运行于.NET 框架之上的高级程序设计语言。本章着重讲解 C#语言和.NET 框架以及命名空间的作用，此外还对 C#开发工具进行简单介绍。本章是读者学习 C#程序设计语言的基础，也是学习 C#程序设计语言的关键所在。希望读者通过本章的学习，可以对.NET 框架、Visual Studio 2010 .NET 集成环境以及 Visual C# .NET 应用程序结构有一定的了解。

1.1 .NET 框架概述

Microsoft .NET 框架是一种采用系统虚拟机运行的编程平台，它以公共语言运行库(Common Language Runtime, CLR)为基础，支持多种编程语言(C#、F#、VB.NET、C++、Python 等)的开发。它是一个致力于敏捷软件开发(Agile Software Development)、快速应用开发(Rapid Application Development)、平台无关性和网络透明化的软件开发平台。

Microsoft .NET 框架由 Microsoft .NET 框架类库、应用程序开发技术、基类库和公共语言运行库(CLR)这几个部分组成，如图 1.1 所示。其中每个较高的层都使用一个或多个较低的层。

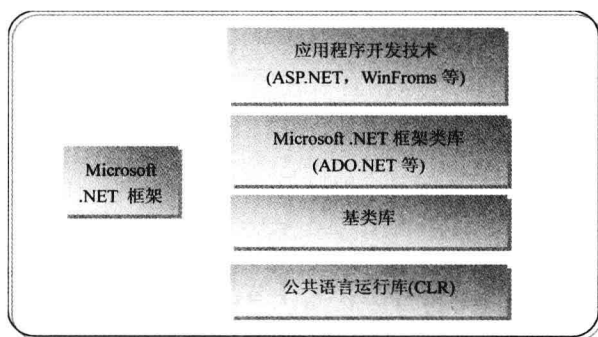


图 1.1 Microsoft .NET 框架的组成

1. 应用程序开发技术

应用程序开发技术位于框架的最上方，是应用程序开发人员关注的主要对象。它包括 ASP.NET 技术和 WinForms 技术等高级编程技术。

2. Microsoft .NET 框架类库

.NET 框架类库是一个与公共语言运行库紧密集成的可重用的类型集合，用于应用程序开发的一些支持性的通用功能。开发人员可以使用它来开发多种模式的应用程序，可以是命令行形式的应用，也可以是图形界面形式的应用。Microsoft .NET 框架包括可加快和优化开发过程并提供对系统功能访问的类、接口和值类型。

3. 基类库

基类库提供了支持底层操作的一系列通用功能。Microsoft .NET 框架主要覆盖了集合操作、线程支持、代码生成、输入输出(I/O)、映射和安全等领域的内容。

4. 公共语言运行库(CLR)

公共语言运行库是 Microsoft .NET 框架的基础内容，也是 Microsoft .NET 程序的运行环境，用于执行和管理任何一种针对 Microsoft .NET 平台的所有代码。CLR 可以为应用程序提供很多核心服务，如内存管理、线程管理和远程处理等，并且还强制实施代码的安全性和可靠性管理。



下面简要介绍 Microsoft .NET 框架的一些核心技术特点,以便读者能够更深入地了解 C#程序的后台运行原理。

(1) 多语言支持:在 Microsoft .NET 平台上,所有的语言都是等价的,它们都是基于公共语言运行库(CLR)的运行环境进行编译运行。所有 Microsoft .NET 支持的语言,不管是 Visual Basic .NET、Visual C++、Visual C#还是 JScript .NET,都是平等的。用这种语言编写的代码都被编译成一种中间代码,在公共语言运行库中运行。在技术上,这种语言与其他语言相比没有很大的区别,用户可以根据自己熟悉的编程语言进行操作。

在本书中,我们使用 C#进行编程,因为 C#是一种优秀的程序开发语言,它简洁、高效且便于使用,主要用于 Microsoft .NET 框架中面向组件的领域。

(2) 多平台支持:Microsoft .NET 框架的另一个重要特点就是多平台支持。不过相对于 Java 技术能够跨越 Unix、Linux 和 Windows 等众多平台而言,目前 Microsoft .NET 的跨平台性仅限于各种 Windows 操作系统。

(3) 性能:Microsoft .NET 的基本设计目标之一就是具有强大的性能和可伸缩性。对于 Microsoft .NET 来说,要具备很好的性能,要靠公共语言运行库来执行中间代码。为了确保最佳性能,在某种意义上,公共语言运行库将所有引用程序代码都编译成本机代码。这种转换既可以在应用程序运行时完成,也可以在应用程序首次安装时完成。

1.1.1 公共语言运行库

在.NET 框架概述中已经简单地提到了公共语言运行库的有关知识,本小节将对该知识做进一步介绍。公共语言运行库(Common Language Runtime, CLR)是托管代码执行核心中的引擎。

公共语言运行库的功能通过编译器和工具公开,利用托管,可以编写执行环境的代码。使用基于公共语言运行库的语言编译器开发的代码称为托管代码。托管代码具有许多优点,如跨语言集成、跨语言异常处理、增强的安全性、版本控制和部署支持、简化的组件交互模型、调试和分析服务等。

公共语言运行库自动处理对象布局并管理对象引用,当不再使用对象时就释放它们。按这种方式实现生存期管理的对象称为托管数据。垃圾回收则消除了内存泄漏以及其他一些常见的编程错误。如果编写的代码是托管代码,则可以在 Microsoft .NET 框架应用程序中使用托管数据、非托管数据或者同时使用这两种数据。由于语言编译器会提供自己的类型(如基元类型),因此并不总是知道(或需要知道)这些数据是否是托管的。

有了公共语言运行库,就可以很容易地设计出对象能够跨语言交互的组件和应用程序。也就是说,用不同语言编写的对象可以互相通信,并且它们的行为可以紧密集成。

例如,可以定义一个类,然后使用不同的语言从原始类派生出另一个类或调用原始类的方法。

还可以将一个类的实例传递到用不同语言编写的另一个类的方法。

这种跨语言集成之所以成为可能,是因为基于公共语言运行库的语言编译器和工具使用由公共语言运行库定义的通用类型系统,而且它们遵循公共语言运行库关于定义新类型以及创建、使用、保持和绑定到类型的规则。

图 1.2 列出了各种程序与 CLR 之间的关系。

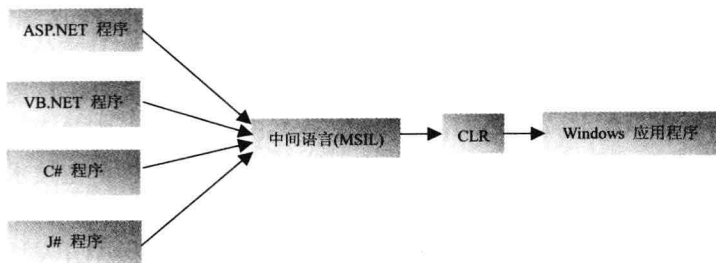


图 1.2 各程序与 CLR 之间的关系

(1) 公共语言运行库具有以下一些优点：

- 改进了性能。
- 能够轻松地使用其他语言开发的组件。
- 类库提供了可扩展类型。

(2) 公共语言运行库的执行模型包括以下几种：

- 将源代码(VB、C#)编译成托管代码块。托管代码块由中间语言和元数据组成。
- 将托管代码合并成程序集。
- 加载公共语言运行库。
- 执行程序集的代码。
- 生成本地代码。

1.1.2 类库

.NET 框架类库是一个由 Microsoft .NET 框架 SDK 中包含的类、接口和值类型组成的库，也是一个与公共语言运行库紧密集成的可重用的类型集合。该库提供对系统功能的访问，是建立 .NET 框架应用程序、组件和控件的基础。

该类库是面向对象的，并提供开发者自己的托管代码可从中导出功能的类型。这不但使 .NET 框架类型易于使用，而且还减少了学习 .NET 框架新功能所需要的时间。此外，第三方组件可与 .NET 框架中的类无缝集成。

.NET 框架类库提供了几乎所有应用程序所需要的公共代码。与 Windows 中 SDK 的库一样，类库使得开发者能将精力集中于编写他们的应用领域所特有的应用程序的代码，而不必一再重复编写类似于“读写文件”这样经常使用的功能的代码。.NET 框架类库包括的类型可执行下列功能：

- 表示基础数据类型和异常。
- 封装数据结构。
- 执行 I/O。
- 访问关于加载类型的信息。
- 调用 .NET 框架安全检查。
- 提供数据访问、多客户端 GUI 和服务端控制的客户端 GUI。

.NET 框架类库在域名空间“System”之下。为了方便查找开发者所需要的类，该域名



空间按照功能区的分级制度进行排列，如图 1.3 所示。

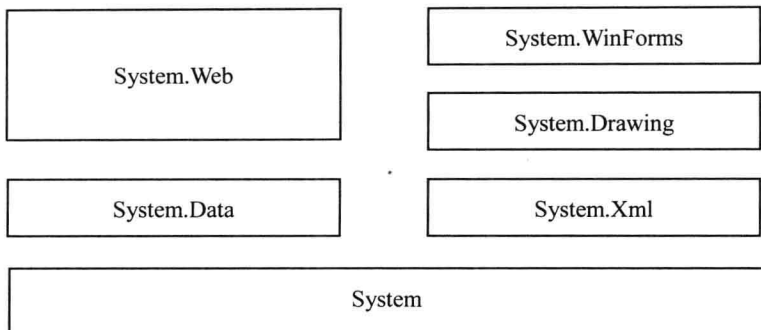


图 1.3 .NET 框架类库域名空间

1.1.3 .NET 程序的编译与运行

C#语言被设计为一种简单的、通用的以及面向对象的编程语言，它位于公共语言运行库之上，也就是说，C#无法离开公共语言运行库而单独运行。C#语言在.NET 框架上从编译到运行经历以下几个过程。

(1) 在编译期，公共语言运行库对 C#代码进行第一次编译，生成中间代码的 DLL 或者 EXE 文件。

(2) 在运行期，公共语言运行库将针对特定的硬件环境使用 JIT(即时编译引擎)，将中间代码编译成二进制代码并执行。

(3) 编译好的二进制代码将放到缓存区缓存，下次再调用相同代码时直接从缓存区调用，相同代码的编译只会执行一次。

下面详细介绍编译运行时的一些概念。

1. MSIL 和 JIT

在编译使用.NET 框架创建的代码时，不是立即创建操作系统特定的本机代码，而是把代码编译为微软中间语言(Microsoft Intermediate Language, MSIL)代码，这些 MSIL 代码不专用于任何一种操作系统，也不专用于任何一种语言，有些类似于 Java 的字节码。C#及其他.NET 语言，如 VB.NET 在编译阶段都编译为这种语言。

因为代码在编译阶段没有直接编译成本机代码，所以在执行应用程序时，必须完成更多的工作，这就是 Just-In-Time(JIT)编译器的任务。

JIT 把 MSIL 编译为专用于某种操作系统和目标机器结构的本机代码，只有这样，操作系统才能执行应用程序。这里编译器的名称 Just-In-Time 反映了 MSIL 仅在需要时才编译的特性。

过去，常常需要把代码编译为几个应用程序，每个应用程序用于特定的操作系统和 CPU 结构，这通常是一种优化形式(例如，为了让代码在 AMD 芯片上运行得更快)，但更多时候是必需的(例如分别运行在 Windows 和 Linux 操作系统上)。现在就不必要了，顾名思义，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

译器来创建所需的本机代码。这样，用户需要做的工作就比较少了。实际上，用户不必考虑与系统相关的细节，只需要把注意力放在代码的功能上就足够了。

2. 程序集

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，其扩展名为.exe)和其他应用程序使用的库(其扩展名是.dll)。

除了包含 MSIL 外，程序集还包含元数据(即程序集中包含的数据信息)和可选的资源(MSIL 使用的其他数据，例如声音和图片文件)。在程序集包含的所有文件中，有一个文件用于保存清单(清单是元数据部分中一组数据表的集合，其中包含了程序集中一部分文件的名称，描述了程序集的版本、语言文化、发布者、共有导出类型，以及组成该程序集的所有文件)。需要注意的是一个程序集只能有一个程序清单。

元数据允许程序集是完全自我描述的，不需要其他信息就可以使用程序集。也就是说，我们不再需要把应用程序所需要的数据添加到系统注册表中，因此，部署应用程序就非常简单了，只需把文件复制到远程计算机中的目录下即可。

当然，不必把运行应用程序所需要的所有信息都安装到一个地方。我们可以编写一些程序集，执行多个应用程序所要求的任务。此时，通常把这些可重用的程序集放在所有应用程序都可以访问的地方。在.NET 框架中，这个地方是“全局程序集高速缓冲存储器”(Global Assembly Cache)，有相应的工具可以帮助把程序集放在高速缓冲存储器中。

3. 托管代码

把代码编译为 MSIL，再用 JIT 编译器把它编译为本机代码后，CLR 的任务还没有完全完成。用.NET 框架编写的代码在执行时是托管的，即 CLR 管理着应用程序，其方式是管理内存、处理安全性，以及允许进行跨语言调试等。

而不在 CLR 控制之下运行的应用程序是非托管的，某些语言，如 C++可以用于编写这类应用程序，例如访问操作系统的低级功能。

使用 C#主要编写在托管环境下运行的代码，它们使用 CLR 的托管功能，让.NET 自己与操作系统进行交互，当然也可以编写在非托管环境下运行的代码，但需要特别标注。

4. 垃圾回收

托管代码最重要的一个功能是垃圾回收(Garbage Collection)，它可确保应用程序不再使用某些内存时，这些内存就会被完全释放。在.NET 推出以前，这项工作主要由程序员负责，代码中的几个简单错误，会把大块内存分配到错误的地方，使这些内存神秘失踪。这通常意味着计算机的运行速度会逐渐减慢，最终导致系统崩溃。

.NET 的无用存储单元收集器会频繁检查计算机内存，从中删除不再需要的内容。它可能一秒钟内会进行上千次的检查，也可能几秒钟检查一次，或者随时进行检查，但可以肯定的是“进行了检查”。

5. 应用程序域

在以前传统的开发中我们都知道，一个应用程序对应一个进程，并为该进程指定虚拟内存，由操作系统来映射实际的物理内存，有效地维护了进程之间的安全性。但另一方



面, 每一个进程都会消耗一定的系统资源, 降低了性能, 并且进程间的通信也比较麻烦。在.NET 中推出了一个新的概念: 应用程序域(AppDomain)。可以理解成很多应用程序域都可以运行在同一个.NET 的进程中, 可以降低系统消耗, 同时不同的域之间互相隔离, 在安全性方面有保障。另外对于同一个进程内不同域之间的通信也相对简单一点。

1.1.4 .NET 框架与 J2EE

作为当今企业应用的两大体系结构, J2EE 和.NET 开发平台在目标和体系结构上极其相似, 但在实现上又完全不同。

对于.NET 框架的每一部分, J2EE 体系结构多少都有所体现。它们都是企业级开发平台, 并且广泛应用于 Web 开发领域。.NET 框架和 Java 都使用了一种托管的运行环境, 都将源代码转换为一种中间语言, 然后将其编译为本地的可执行代码。这两种体系结构都提供了无用存储单元收集器、动态类加载和异常处理等。

J2EE 和.NET 这两个平台在底层的执行引擎都源于托管的虚拟机概念, 但.NET 的 CLR 沿着 Java 虚拟机(JVM)走得更远, CLR 在借鉴 JVM 的自动垃圾收集、异常处理等机制的同时, 又为.NET 平台添加了多语言支持、组件自描述等新的特性。

J2EE 最大的优势就是跨平台, .NET 最大的优势是入门容易和性能较高。对于没有.NET 和 J2EE 平台基础的开发团队来说, 如果开发的应用程序没有跨平台的要求, 只需要运行在微软平台之上, 则选择.NET 平台比较好; 如果要求跨平台, 则只能使用 J2EE。

1.2 Visual Studio 2010 .NET 集成环境

Visual Studio 2010 是目前最为流行的 Windows 平台应用程序开发环境, 它与 Microsoft .NET 开发框架紧密结合, 是构建下一代互联网应用的优秀工具。Visual Studio 2010 的集成开发环境(IDE)界面被重新设计和组织, 变得更加简单明了, 大大地提高了开发效率; 同时带来了.NET Framework 4.0、Microsoft Visual Studio 2010 CTP(Community Technology Preview, CTP), 并且支持开发面向 Windows 7 的应用程序; 除了 Microsoft SQL Server, 它还支持 IBM DB2 和 Oracle 数据库。

1.2.1 Visual Studio 2010 概述

Visual Studio 2010 是用于开发和维护托管的、本机的和混合模式的应用程序的集成开发环境(Integrated Development Environment, IDE)。它提供了用于创建不同类别应用程序的多种项目模板, 这些模板包括 Microsoft Windows 窗体、控制台、ASP.NET 网站、ASP.NET Web 服务、Windows Phone OS 7.0 以及其他类型的应用程序。此外, 它还提供了针对特定设备(如移动设备)的模板。而且, 开发人员还可以根据需要进行不同的编程语言, 包括 C#、Visual Basic .NET、Visual F#和托管的 C++等。

Visual Studio 2010 IDE 具有很多应用程序的快速开发(Rapid Application Development, RAD)工具。Visual Studio 代码编辑器、Microsoft IntelliSense、解决方案资源管理器、类视图、对象浏览器和类关系图都是用户界面的核心部分, 有助于提高开发人员的精确性和工

作效率。

1.2.2 Visual Studio 2010 的新特性

Visual Studio 2010 作为微软着力打造的下一代开发工具平台，跟 Visual Studio 2008 相比，新增了很多便于程序开发的新特性。

在 Visual Studio 2010 中，微软采用了全新的 WPF 技术重新打造了它的编辑器，新的编辑器以及 Visual F# 获得了更加强大的功能，成为更好的开发工具。比如代码的无级缩放，多窗口即时更新，代码的自动产生等，这些新的 IDE 特性都会极大地提高程序员的开发效率。下面详细介绍一下 Visual Studio 2010 的新增功能。

1. 代码编辑器

新的代码编辑器使代码更易于阅读。可以通过按 Ctrl 键并滚动鼠标滚轮来放大文本。此外，单击 Visual C# 或 Visual Basic 中的符号时，该符号的所有实例都将自动地突出显示出来。

2. Visual F#

Visual Studio 2010 中已正式包含 Visual F#，作为一种常规的开发工具发布。作为 .NET 框架家族的新成员，F# 可提供类型安全、性能以及类似脚本语言的工作能力，所有这些都是 .NET 环境的一部分。

F# 为 Visual Studio 2010 带来了很大的改观。因为替代了局部程序中的元组，工作变得简单了。在异步编程和强类型及对于浮点编码方面也提供了不错的解决方式。可以说 F# 是 Visual Studio 2010 中结合了众多优点的编程语言。

3. C# 4.0

C# 4.0 主要改进了动态编程模式的互操作性和 Office 编程，动态查询是 C# 4.0 中的一个新功能，允许我们从 IronPython、IronRuby、JScript、HTML DOM 或一个标准的 .NET 库以同样的方式使用和操作一个对象，不管它们来自哪里。

C# 4.0 中的动态关键字功能允许一个类型在运行时动态解析，而不是在编译时静态的方式，这可以让动态语言将它们的对象暴露给 C#。

4. 为开发者提供更好的工具和框架

Visual Studio 2010 使得开发者可以针对众多的应用程序进行开发，诸如 Windows 7、Office、Web、智能手机、云应用框架等。

微软除了支持其基本平台外，也准备提供对新兴应用趋势的工具支持，即云计算和并行设计，以充分利用高性能的多核系统。例如，为并行编程提供了更多支持，包括增加调试窗口，增加对 .NET 框架 PLINQ、本地 Parallel Pattern Library 和 Concurrency Runtime 并行扩展编码的支持。

第三方产品和框架也在该 IDE 中有了用武之地。Visual Studio 2010 中有两个值得注意的内嵌工具：一个是 Quest Software 公司开发的可选数据库架构，提供了对 Oracle 数据库的支持，另一个是本地 jQuery，开发人员会很容易熟悉而能立即上手利用新的功能。