

清华
电脑学堂

超值多媒体光盘
完整实例源代码
12段教学视频文件

- ✓ 合理的知识结构
- ✓ 真实的案例教学
- ✓ 阶梯式实践环节
- ✓ 理论实践一体化

C# 程序设计

基础教程与实验指导

■ 孙晓非 牛小平 冯冠 李乃文

清华大学出版社

清华
电脑学堂

C# 程序设计

基础教程与实验指导

■ 孙晓非 牛小平 冯冠 李乃

清华大学出版社
北 京

内 容 简 介

本书由浅入深地介绍 C# 开发知识, 共分 12 章, 介绍 .NET 开发基础、C# 基础语法、函数、面向对象的编程、数组与集合、处理字符串、处理异常、Windows 窗体控件、可视化界面设计、文件存取、ADO.NET 数据库编程等内容。本书综合案例对超市管理系统需求和功能进行了分析, 并使用 C# 来实现超市商品管理、销售额统计和用户积分的计算等。本书附有配套光盘, 提供了书中实例的源代码和视频教学文件。

本书体现了作者在软件技术教学改革过程中形成的“项目驱动、案例教学、理论实践一体化”教学方法, 可以作为 C# 职业培训教材和各级院校 C# 授课培训教程的教材, 也适合作为 C# 自学资料和参考资料使用。

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# 程序设计基础教程与实验指导 / 孙晓非等编著. —北京: 清华大学出版社, 2012.3
(清华电脑学堂)

ISBN 978-7-302-26896-3

I. ①C… II. ①孙… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 193317 号

责任编辑: 夏兆彦

封面设计: 柳晓春

责任校对: 徐俊伟

责任印制: 杨 艳

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

印 刷 者: 清华大学印刷厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 24.25 字 数: 606 千字
(附光盘 1 张)

版 次: 2012 年 3 月第 1 版

印 次: 2012 年 3 月第 1 次印刷

印 数: 1~4000

定 价: 45.00 元

Visual C#是 Microsoft 公司开发的一种使用简单、功能强大、面向组件、表达力丰富的语言。它结合了 C++强大灵活和 Java 语言简洁等特性,还吸取了 Delphi 和 Visual Basic 所具有的易用性。C#在 Microsoft .NET Framework 4.0 框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技术。

.NET Framework 的基础是公共语言运行时。它是执行时管理代码的代理,提供内存管理、线程管理和远程处理等核心服务。本书以 Visual Studio 2010 为基础,结合 .NET Framework 4.0 类库以及 Visual Studio 2010 自带的 SQL Server 2008 R2 Express 数据库,介绍 C#语言的知识以及窗体应用程序的开发技巧,并配有完整的开发实例讲解。

1. 本书定位与特色

□ 面向职业技术教学

本书是作者在总结了多年开发经验与成果的基础上编写的,以实际项目为中心,全面、翔实地介绍 C#开发所需的各种知识和技能。通过本书的学习,读者可以快速、全面地掌握使用 C#进行面向对象开发的方法。本书体现了作者“项目驱动、案例教学、理论实践一体化”教学理念,是一本真正面向职业技术教学的教材。

□ 合理的知识结构

本书面向程序员职业培训市场,结合程序开发实践介绍 C#编程知识,突出职业实用性。本书各章都有实例分析,带领读者经历程序开发全过程,是一本真正的实训性案例教程。

□ 真实的案例教学

针对每个知识点,本书设计了针对性强的教学案例,这些小案例既相对独立,又具有一定的联系,是综合性开发实例的组成部分。读者在制作这些小案例的过程中可以掌握每个知识点。

□ 理论实践一体化

在每个案例中,本书都有机融合了知识点讲解和技能训练目标,融“教、学、练”于一体。每个案例的讲解都首先提出功能目标,然后是实例制作演示和学生模仿练习,让读者掌握案例的完成过程,体现“在练中学、学以致用”的教学理念。

□ 阶梯式实践环节

本书精心设置了 3 个教学环节:课堂练习、扩展练习、综合实训,让学生通过不断练习实践,实现编程技能的逐步推进,最终与职业能力的接轨。

2. 本书主要内容

本书由浅入深地介绍 C#开发知识,共分 12 章,介绍 .NET 开发基础、C#基础语法、

函数、面向对象的编程、数组与集合、处理字符串、处理异常、Windows 窗体控件、可视化界面设计、文件存取、ADO.NET 数据库编程等知识。

本书最后的综合案例对超市管理系统需求和功能进行分析，介绍系统总体结构和用例图，最后介绍数据库设计，并使用 C#来实现超市管理系统。本章详细剖析了实现这些模块的 C#代码。

本书附有配套光盘。光盘提供了书中实例的源代码，全部经过精心调试，在 Windows XP/Windows 2000/Windows 2003 Server/Vista/Windows 7/Windows 2008/R2 下全部通过，能够保证正常运行。

3. 读者对象

本书体现了作者在软件技术教学改革过程中形成的“项目驱动、案例教学、理论实践一体化”教学方法，读者通过本书可以快速、全面地掌握使用 C#应用于 Windows 编程的开发经验和技能。本书可以作为 C#职业培训教材和各级院校 C#授课培训教程的教材，也适合作为 C#自学资料和参考资料。

除了封面署名人员之外，参与本书编写的还有孙岩、马海军、张仕禹、夏小军、赵振江、李振山、李文采、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林等。由于编者水平有限，本书在编写过程中难免会有漏洞，欢迎读者通过清华大学出版社网站 www.tup.tsinghua.edu.cn 与我们联系，帮助我们改正提高。

编 者

2011 年 6 月

第 1 章 .NET 开发基础	1	2.3.2 赋值运算	40
1.1 Microsoft .NET 框架	1	2.3.3 关系运算	41
1.1.1 Microsoft .NET 简介	1	2.3.4 逻辑运算	43
1.1.2 C#编程语言	5	2.3.5 按位运算	43
1.2 公共语言运行时	6	2.3.6 特殊运算	46
1.2.1 公共类型系统	6	2.3.7 运算的优先级	48
1.2.2 公共语言规范	7	2.4 流程控制	49
1.2.3 中间语言和 JIT 编译器	8	2.4.1 条件分支控制	50
1.2.4 托管执行过程	9	2.4.2 迭代控制	52
1.2.5 自动内存管理	10	2.4.3 跳转与中断	55
1.3 .NET Framework 类库	11	2.5 扩展练习	56
1.4 程序集	13	第 3 章 函数	58
1.4.1 程序集概述	13	3.1 函数基础知识	58
1.4.2 程序集内容	15	3.1.1 计算机函数	58
1.4.3 程序集清单	16	3.1.2 自定义函数	59
1.5 .NET 开发工具	17	3.2 函数的返回值	62
1.5.1 Visual Studio 2010 简介	17	3.3 函数的参数	63
1.5.2 安装与配置 Visual Studio		3.3.1 形式参数与实际参数	64
2010	19	3.3.2 引用参数与输出参数	65
1.6 命名空间	22	3.3.3 不定参数	67
1.6.1 命名空间结构	22	3.4 函数重载	68
1.6.2 定义命名空间	23	3.5 入口函数	69
1.6.3 使用 using 关键字	25	3.6 扩展练习	71
第 2 章 C#基础语法	26	第 4 章 面向对象的编程	72
2.1 数据类型	26	4.1 面向对象的基本概念	72
2.1.1 值类型	26	4.1.1 面向对象与面向过程	72
2.1.2 引用类型	31	4.1.2 面向对象的程序设计	74
2.1.3 装箱和拆箱	32	4.1.3 面向对象的特点	76
2.2 变量和常量	32	4.2 类和成员	78
2.2.1 变量	33	4.2.1 定义类	78
2.2.2 常量	35	4.2.2 定义字段	81
2.2.3 作用域和修饰符	36	4.2.3 定义属性	82
2.3 数据运算	37	4.2.4 定义方法	83
2.3.1 算术运算	37	4.3 构造函数与析构函数	85

4.3.1	构造函数	85	6.2.4	其他处理方式	130
4.3.2	析构函数	87	6.3	字符串的进阶操作	133
4.4	抽象类与接口	88	6.3.1	追加字符串	133
4.4.1	抽象类	88	6.3.2	格式化字符串	134
4.4.2	接口	89	6.3.3	插入字符串	135
4.5	委托与类事件	91	6.3.4	替换字符串	136
4.5.1	委托	91	6.3.5	其他进阶处理方式	137
4.5.2	声明事件	92	6.4	正则表达式	139
4.5.3	为类定义事件	93	6.4.1	正则表达式概述	139
4.6	泛型	94	6.4.2	正则表达式语法	139
4.6.1	泛型的基本概念	95	6.4.3	使用正则表达式	147
4.6.2	定义与使用泛型类	95	6.5	扩展练习	148
4.7	扩展练习	97	第 7 章 处理异常		149
第 5 章 数组与集合		98	7.1	异常处理基础	149
5.1	使用数组	98	7.1.1	异常处理机制	149
5.1.1	定义数组	98	7.1.2	嵌套 Try 语句	151
5.1.2	数组的维度	100	7.1.3	Finally 块	152
5.1.3	数组的嵌套	100	7.2	异常类	153
5.2	操作数组元素	101	7.2.1	使用异常类	153
5.2.1	获取数组元素数与类型	101	7.2.2	基于类型的筛选异常	155
5.2.2	遍历数组	104	7.2.3	内部异常	157
5.2.3	检索数组元素索引	105	7.3	自定义异常类	158
5.2.4	翻转数组元素	109	7.4	扩展练习	159
5.3	定义集合	110	第 8 章 Windows 窗体控件		160
5.3.1	集合的概念	110	8.1	Windows 窗体概述	160
5.3.2	泛型集合	111	8.1.1	Windows 窗体界面技术	160
5.4	常用泛型集合	112	8.1.2	操作控件	162
5.4.1	列表集合 List	113	8.1.3	编辑窗体控件	163
5.4.2	队列集合 Queue	118	8.2	文本窗体控件	166
5.4.3	字典集合 Dictionary	120	8.2.1	TextBox 控件	166
5.5	扩展练习	122	8.2.2	RichTextBox 控件	167
第 6 章 处理字符串		123	8.2.3	MaskedTextBox 控件	169
6.1	字符串基础	123	8.3	显示信息窗体控件	170
6.1.1	字符串与数据的转换	123	8.3.1	Label 控件	170
6.1.2	字符串的属性	124	8.3.2	LinkLabel 控件	171
6.2	处理字符串	125	8.4	图形图像类控件	171
6.2.1	比较字符串	125	8.4.1	ImageList 控件	172
6.2.2	定位与查找字符	127	8.4.2	PictureBox 控件	173
6.2.3	提取和拆分字符串	129	8.5	按钮类控件	173

8.5.1 Button 控件	173	10.4.1 流式存取基础	218
8.5.2 RadioButton 控件	175	10.4.2 读写顺序文件	219
8.5.3 CheckBox 控件	175	10.4.3 读写随机文件	220
8.6 列表类控件	176	10.4.4 读写二进制流数据	221
8.6.1 ListBox 和 CheckedListBox 控件	176	10.5 扩展练习	223
8.6.2 ComboBox 控件	178	第 11 章 ADO.NET 数据库编程	224
8.6.3 ListView 控件	179	11.1 数据库基础	224
8.7 容器类控件	181	11.1.1 数据库基本知识	224
8.8 扩展练习	183	11.1.2 常用数据库系统	225
第 9 章 可视化界面设计	184	11.1.3 SQL 语句	226
9.1 基于对话框的应用程序	184	11.2 ADO.NET 技术	227
9.2 基于文档的应用程序	185	11.2.1 ADO.NET 简介	227
9.2.1 SDI 应用程序	185	11.2.2 ADO.NET 类库	228
9.2.2 MDI 应用程序	185	11.3 连接数据库	229
9.3 处理窗体组件	186	11.3.1 定义连接字符串	230
9.3.1 创建菜单栏	186	11.3.2 存储连接字符串	231
9.3.2 编辑工具栏	189	11.3.3 读取连接字符串	231
9.3.3 编辑状态栏	192	11.3.4 测试连接	232
9.4 GDI+设计	193	11.4 操作数据库	234
9.4.1 了解 GDI+	193	11.4.1 使用 DataAdapter 填充 DataSet 对象	234
9.4.2 Graphics 类	194	11.4.2 使用 Command 对象	235
9.4.3 标准坐标系	196	11.4.3 使用 DataReader 类	237
9.4.4 处理颜色	199	11.4.4 使用 DataTable 和 DataView 类	238
9.4.5 定义画笔样式	200	11.4.5 定义数据库关系	241
9.4.6 绘制几何图形	201	11.4.6 使用 Command Builder 生成 SQL 语句	241
9.5 扩展练习	203	11.5 DataSet 应用	243
第 10 章 文件存取	205	11.5.1 更新 DataSet	243
10.1 文件数据操作简介	205	11.5.2 向 DataSet 添加数据	244
10.1.1 文件系统基础	205	11.5.3 对 DataSet 排序和筛选	245
10.1.2 System.IO 命名空间	207	11.6 扩展练习	246
10.2 路径、目录和文件	208	第 12 章 超市管理系统	248
10.2.1 目录和目录信息类	208	12.1 系统设计分析	248
10.2.2 文件和文件信息类	211	12.1.1 需求分析	248
10.2.3 路径类	215	12.1.2 功能分析	249
10.3 目录和文件对话框	216	12.2 系统设计概要	250
10.3.1 打开文件对话框	216	12.2.1 系统总体结构	250
10.3.2 保存文件对话框	218		
10.4 文件流的操作	218		

12.2.2 系统用例图	251	练习 1-3 创建控制台 Hello World 程序	308
12.3 系统模块设计	253	练习 1-4 创建 Windows 窗体 HelloWorld 程序	310
12.3.1 登录	253	第 2 单元	311
12.3.2 商品类别管理	254	练习 2-1 打印正三角形图形	311
12.3.3 商品管理	254	练习 2-2 十进制数转换为 二进制数	312
12.3.4 员工管理	255	练习 2-3 打印杨辉三角	313
12.3.5 会员管理	255	练习 2-4 分解质因数	315
12.3.6 积分规则管理	255	第 3 单元	316
12.3.7 查看统计信息	256	练习 3-1 计算最大公约数和 最小公倍数	316
12.3.8 日常销售	256	练习 3-2 完全数	318
12.4 数据库设计	256	练习 3-3 亲密数	319
12.5 登录及系统主窗体模块	258	练习 3-4 计算圆周率 π 近似值	321
12.5.1 编写数据库操作类	258	第 4 单元	322
12.5.2 登录窗体	260	练习 4-1 创建商品信息类	322
12.5.3 注销和退出	263	练习 4-2 根据出生日期 计算年龄	324
12.6 商品类别管理模块	264	练习 4-3 计算员工工资	326
12.6.1 添加商品类别	264	第 5 单元	327
12.6.2 编辑商品类别	268	练习 5-1 学生成绩排序	327
12.6.3 删除商品类别	271	练习 5-2 求某日在当年天数	328
12.7 商品管理模块	276	练习 5-3 求平均值	330
12.7.1 添加商品	276	第 6 单元	331
12.7.2 编辑商品	279	练习 6-1 获取关键字出现次数	331
12.8 员工管理模块	282	练习 6-2 转换字符串大小写	332
12.8.1 添加员工	283	练习 6-3 判断是否为 E-mail 地址	333
12.8.2 编辑员工	284	第 7 单元	334
12.8.3 删除员工	286	练习 7-1 判断除数为 0 异常	334
12.9 会员管理模块	287	练习 7-2 判断用户输入	335
12.9.1 添加会员	287	练习 7-3 搜索产品内容	336
12.9.2 编辑会员	289	第 8 单元	338
12.9.3 删除会员	291	练习 8-1 制作简单文本 显示程序	338
12.10 积分规则管理	292	练习 8-2 制作员工信息 录入程序	340
12.11 查看统计信息模块	294		
12.12 日常销售模块	297		
附录 单元练习	302		
第 1 单元	302		
练习 1-1 设置 Visual Studio 2010 开发环境	302		
练习 1-2 新建 Visual Studio 项目	307		

练习 8-3 制作整数计算器	343
第 9 单元	347
练习 9-1 制作拼图游戏	347
练习 9-2 制作简单绘图板	350
第 10 单元	355

练习 10-1 制作柱状图表	355
练习 10-2 制作文本编辑器	357
第 11 单元	370
练习 11-1 员工信息登记	370
练习 11-2 个人收支管理	373

第1章 .NET 开发基础

Microsoft .NET 框架是由微软公司开发的软件开发系统平台，是一种主要用于 Windows 操作系统的托管代码编程模型。它提供大量的公共类库，为多种编程语言提供支持，实现本地应用、富互联网应用和服务器端应用。

在学习 C# 语言时，可先阅读本章内容，了解 .NET Framework 与 C# 之间的关系，以及开发 C# 时所需要理解的 .NET 框架基础知识。

本章学习目标

- 了解 .NET 框架
- 了解 C# 编程语言与 .NET 的关系
- 理解公共语言运行时
- 理解程序集
- 了解 .NET 命名空间
- 了解 Visual Studio 2010
- 掌握 Visual Studio 2010 的安装

1.1 Microsoft .NET 框架

Microsoft .NET 框架的出现，为开发者提供了一种类似虚拟机技术的平台，允许开发者以通用的代码实现多种硬件架构和操作系统的应用程序，降低了软件开发的成本，提高了工作效率。

1.1.1 Microsoft .NET 简介

Microsoft .NET 框架是微软公司面向下一代移动互联网、服务器应用和桌面应用的基础开发平台，是微软为开发者提供的基本开发工具，包含许多有助于互联网应用迅捷开发的新技术，如图 1-1 所示。

1. Microsoft .NET 的产生

在传统的软件开发工作中，开发者需要面对的是多种服务器和终端系统，包括用于个人计算机的 Windows 操作系统、用于服务器的 Windows 服务器系统、非 Windows 系统（如 FreeBSD、Linux 和 BSD）、用于平面设计的 Mac OS X 操作系统，以及各种移动终端系统（如 Windows Mobile、iOS、Android）等。

在开发基于以上这些系统的软件时，开发者往往需要针对不同的硬件和操作系统，编写大量实现兼容性的代码，并使用不同的方式对代码进行编译。这一系列的问题，都

给软件设计和开发带来很多困难。

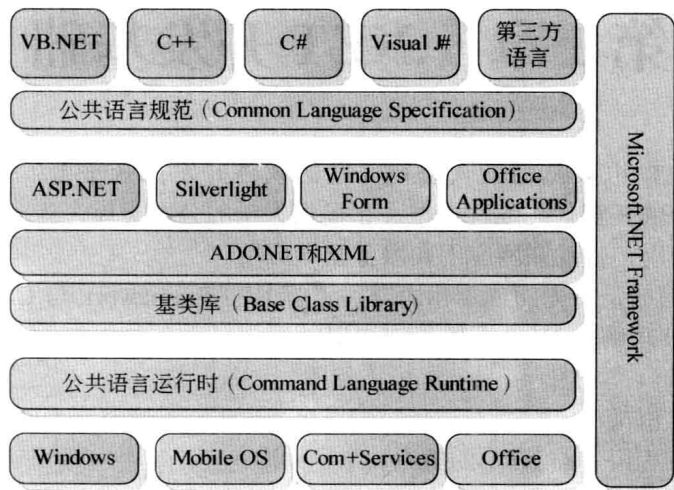


图 1-1 Microsoft .NET 开发平台

以 Windows 操作系统为例，目前主要使用的 Windows 操作系统内核包括 Windows 9X、NT4、NT 5.0/5.1、NT6.0/6.1、Windows CE、Windows Mobile 6.X 和 Windows Phone OS 等。在这些操作系统下进行软件开发，可使用的技术包括以下几种。

- ☐ 用于图形图像开发的 GDI、DirectX、OpenGL 等技术。
- ☐ 用于数据库操作的 ADO、DAO、RDO、ODBC 等技术。
- ☐ 用于 Web 应用开发的 ASP、JSP、PHP 等技术。
- ☐ 用于移动终端的 XNA、HTML 5 等技术。

以上这些技术各有各的标准和接口，相互并不兼容。若干软件开发必须学习和使用相同的技术才能实现协作，而企业在实施开发项目时，也需要聘用指定技术的开发人员，才能实现最终的产品。

基于以上问题，微软公司在 21 世纪初开发出了一种致力于敏捷而快速的软件开发框架。其更加注重平台无关化和网络透明化，以 CLR（Common Language Runtime，通用语言运行时）为基础，支持多种编程语言，这就是 Microsoft .NET 框架。

2. Microsoft .NET 的特点

Microsoft .NET 框架既是一个灵活、稳定的能运行服务器端程序、富互联网应用、移动终端程序和 Windows 桌面程序的软件解析工具（类似虚拟机程序），又是软件开发的基础资源包，其具有以下特点。

☐ 统一应用层接口

.NET 框架将 Windows 操作系统底层的 API（Application Programming Interface，应用程序接口）进行封装，为各种 Windows 操作系统提供统一的应用层接口，从而消除了不同 Windows 操作系统带来的不一致性，用户只需直接调用 API 进行开发，无需考虑

平台。

□ 面向对象的开发

.NET 框架使用面向对象的设计思想, 更加强调代码和组件的重用性, 其提供了大量的类库, 每个类库都是一个独立的模块, 供用户调用。同时, 开发者也可着手自行开发类库给其他开发者使用。

□ 支持多种语言

.NET 框架支持多种开发语言, 允许用户使用符合 CLR 规范的多种编程语言开发程序, 包括 C#、VB.NET、J#、C++ 等, 然后再将代码转换为中间语言存储到可执行程序。在执行程序时, 通过 .NET 组件对中间语言进行编译执行。

3. Microsoft .NET 的版本

Microsoft .NET 框架与 Windows 操作系统和 Microsoft Visual Studio 集成开发环境保持着紧密的联系, 其发布的版本也与这两者紧密相关, 如表 1-1 所示。

表 1-1 Microsoft .NET 框架版本

发布日期	版本	对应 Windows 版本	对应 Visual Studio 版本
2002 年 2 月 13 日	1.0	Windows XP	Visual Studio.NET
2003 年 4 月 24 日	1.1	Windows Server 2003	Visual Studio.NET 2003
2005 年 11 月 7 日	2.0		Visual Studio 2005
2006 年 11 月 6 日	3.0	Windows Vista/Windows Server 2008	
2007 年 11 月 19 日	3.5	Windows 7/Windows Server 2008 R2	Visual Studio 2008
2010 年 4 月 12 日	4.0		Visual Studio 2010

目前最新版本的 .NET 框架 Microsoft .NET Framework 4.0 具有以下几方面特性。

□ 图表控件

在开发 .NET Framework 4.0 的应用程序时, 开发者可以直接从 Visual Studio 2010 中调用之前必须从 Technet 下载的图表控件, 创建更具可视化效果的数据图表。

□ 托管扩展框架

托管扩展性框架 (MEF) 是 .NET Framework 4.0 中的一个新库, 其可以帮助开发者创建可扩展和组合的应用程序, 允许开发者指定应用程序中的扩展点, 为其他应用程序服务。

□ 并行计算

针对越来越多支持多线程技术的处理器, .NET Framework 4.0 中引入了一个新的编程模式, 简化了应用程序和库开发者的编程。此模式可以帮助开发者在不使用线程或线程池时编写高效、具有可扩展性的并行计算程序。

□ 垃圾收集

.NET Framework 4.0 改进了之前版本的并行垃圾收集机制, 支持从后台进行垃圾收集, 从而提供更好的系统性能。

4. Microsoft .NET 的应用

在微软公司发布 .NET 框架之初, 该技术仅仅是一种面向 Windows XP 和 Windows

Server 2003 桌面应用的实现方式。随着富互联网应用和移动计算技术的发展,.NET 框架不断得到增强,目前其已经可以作为一种综合的开发平台,应用到多种领域。

□ 桌面应用

桌面应用是.NET 框架最基本的应用,使用 Microsoft .NET 框架,开发者可以开发出基于 Windows 2000/NT5 以上版本桌面操作系统和服务器操作系统的桌面应用程序,并通过用户计算机的.NET 组件实现本地文档和数据的操作。

使用.NET 框架开发桌面程序,开发者只需将精力专注于程序算法和架构的本身,无需考虑这些桌面操作系统之间的差异,因此可以从繁杂的程序调试和兼容性测试工作中解放出来,极大地提高了工作效率。

□ 服务器应用

服务器应用也是.NET 框架的重要应用之一,使用.NET 框架开发出的服务器应用程序名为 ASP.NET 程序,相比传统的 ASP 程序,.NET 框架将网页分成前台页面和后台系统两个模块,将页面开发层和应用逻辑层完全隔离开,提高网页开发的效率和代码的重用性,增强了服务器应用程序的稳定性和安全性。

□ Office 增强功能

作为微软公司提供的开发工具,Microsoft .NET 框架可以与微软公司开发的 Office 系列办公软件紧密地结合,开发应用于该软件的宏、加载项等,增强 Office 系列办公软件的功能,提高办公效率。

□ 富互联网应用

为抗衡 Adobe 公司开发的 AIR (Adobe Integrated Runtime, Adobe 集成运行时)等富互联网应用技术,微软公司提出了 Silverlight 计划,通过.NET 框架编写基于 Web 的多媒体应用程序,通过丰富的可视化元素实现用户体验。

□ 移动应用

Microsoft .NET 框架不仅可以应用到个人计算机、工作站等平台上,还可以为一些移动计算设备提供支持,例如,使用 Windows CE 操作系统的 PDA、使用 Windows Mobile 和 Windows Phone 7 等操作系统的智能手机等。开发者开发的.NET 程序同样可以在这些设备上执行。

5. 其他平台中的.NET 框架

除了微软公司开发的桌面、服务器和移动设备操作系统外,.NET 框架还可以应用在其他几种操作系统中,通过以下几种技术实现跨平台应用。

□ SSCLI 技术

SSCLI (Microsoft Shared Source Common Language Infrastructure, 微软共享源公共语言平台)是由微软公司提供的代码共享实现的,可以在 Windows XP、FreeBSD、Mac OS X 等操作系统上执行.NET 框架。

□ Mono

Mono 是一个开源的.NET 框架运行时与开发库实现,由 Novell Ximian 和开源软件社区负责开发维护,目前已经实现了对 ASP.NET 和 ADO.NET 的支持,同时支持部分 Windows Forms 库,允许在 Linux 等类 UNIX 系统下开发和执行.NET 程序。

1.1.2 C#编程语言

C#编程语言是微软公司推出的一款基于.NET 框架的、面向对象的高级编程语言，是专门为.NET 框架设计的语言。C#语言由 Java、C 和 C++语言派生而来，继承了这 3 种语言的绝大多数语法和特点，是.NET 框架中最常用的编程语言。

1. C#语言简介

C#语言的语法相对 C 和 C++简单一些，因此使用 C#开发应用程序效率更高、成本更低。C#是一种类型安全语言，其代码并不能被编译为直接在计算机上执行的二进制代码。其与 Java 类似，都会被编译为中间语言代码。因此，虽然编译出的结果与传统的可执行程序相同，都具有“.exe”的扩展名，但 C#并不能被 Windows 操作系统直接执行。

在执行 C#编写的程序时，必须通过.NET 框架组件进行二次编译，将中间语言代码翻译为二进制代码。最终翻译的二进制代码可以存储在一个操作系统的缓冲区中，因此一旦其他程序使用了相同的代码，则 Windows 会直接调用缓冲区中的版本快速执行。

微软公司以 C#为基础，向多家国际标准化组织（包括 ECMA 国际、ISO 等）进行了国际标准申请。在 2001 年，C#成为 ECMA 语言规范（ECMA-334 C#）。在 2003 年成为 ISO 标准（ISO/IEC 23270）。

2. C#语言特点

相比 Java、C 和 C++等编程语言，C#语言具有以下特点。

□ 指针限制

在 C 和 C++等传统的编程语言中，指针（Pointer）被广泛的应用。然而，这种广泛应用往往带来不安全隐患。在 C#中，绝大多数对象访问必须通过安全的引用实现，仅允许在不安全模式下使用指针，以防止无效的调用。同时，即使在不安全模式下，指针也只能够调用值类型和受垃圾收集控制的托管对象。

□ 垃圾回收机制

在 C#中，不允许以显式的方式释放对象，只能在对象未被引用时通过垃圾收集机制回收。因此，使用 C#编写的代码更加健壮、更加安全。

□ 继承与接口

与 C++不同，C#只允许对象的单一继承（Single Inheritance），但是一个类可以实现多个接口。

□ 类型安全

C#更加强调类型安全，其默认的安全转换为隐式转换。然而，接口布尔型与整形、枚举型同整形等数据不允许隐式转换。不同于 C++的复制构造函数，非空指针（通过引用相似对象）同用户定义类型的隐含转换必须被显式确定。

□ 泛型

C#中没有模版（Template），但是 C# 2.0 中引入了泛型（Generic Programming）的概念，支持一些 C++模版不支持的特性。例如，泛型参数中的类型约束。另一方面，表达

式不能像 C++ 模版中被用于类型参数。

1.2 公共语言运行时

公共语言运行时 (Common Language Runtime, CLR) 是 .NET Framework 的基础, 其作用是提供内存管理、线程管理和远程处理等核心服务。

首先, 公共语言运行时分别通过公共类型系统 (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。然后再通过 Just-In-Time 编辑器在运行应用程序之前将中间语言 (Intermediate Language, IL) 代码转换为可执行代码。除此之外, 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, 这些功能是公共语言运行时在运行托管代码时的固有模块, 如图 1-2 所示。

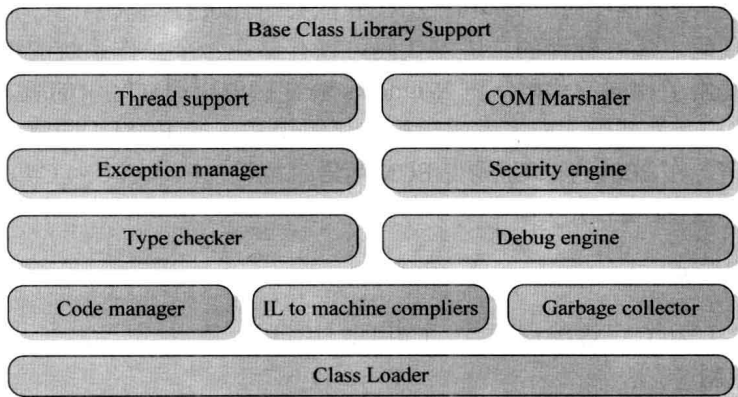


图 1-2 公共语言运行时

1.2.1 公共类型系统

公共类型系统 (CTS) 定义了声明和使用类型的标准, 使得 CLR 可以在不同语言开发的应用程序之间管理这些标准化的类型, 并且在不同计算机之间以标准化的格式进行数据通信。其具有以下功能。

- ☐ 定义了所有应用程序使用的主要数据类型, 以及这些类型的内部格式。例如, CTS 定义了整型是 32 位大小, 还指定了整型值的内部格式。
- ☐ 指定了如何为结构和类分配内存。
- ☐ 允许不同语言开发的组件可以互操作。
- ☐ 实施类型安全性, 禁止一个应用程序使用为另一个应用程序分配的内存。

公共类型系统构成了 .NET 架构的公共语言运行时基础, 其中一个重要体现就是为 .NET 平台的多种语言提供支持。基于公共类型系统的每种语言为了维护自己的语法特色, 往往使用别名来替代基础数据类型。例如, C# 中的 `int` 类型和 Visual J# 中的 `int` 类型都是基础数据类型 `System.Int32` 的别名。

公共类型系统不仅定义了所有数据类型，还提供了面向对象的模型以及各编程语言需要共同遵守的标准。其定义类型可分为两大类，即值类型和引用类型，其结构如图 1-3 所示。

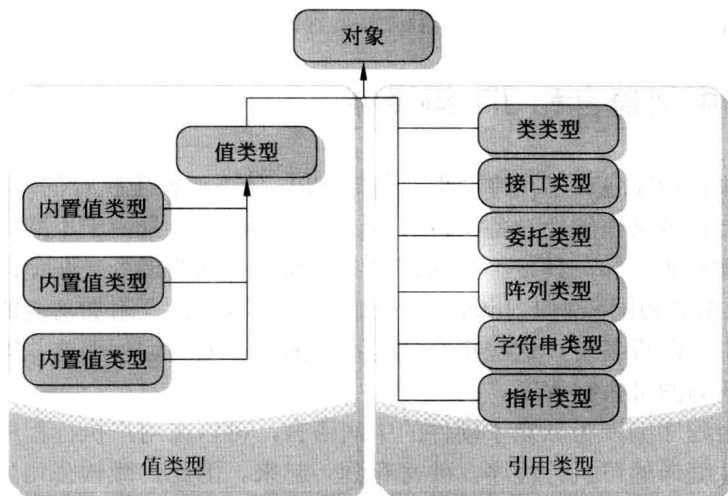


图 1-3 公共类型系统的基本结构

.NET 框架允许值类型和引用类型之间的强制转换，这种转换被称作装箱（Boxing）和拆箱（UnBoxing）。CTS 的每一种类型都是对象，并继承自一个基类——System.Object。

1.2.2 公共语言规范

公共语言规范（CLS）是一组结构和限制条件，其可为库编写者和编译器编写者提供指南，使任何支持 CLS 的语言都完全使用库，并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，这两个集合一起定义了所有 .NET 编程语言的标准集，允许这些编程语言编写的应用程序相互通信和操作。

CLS 和 .NET 自身都依赖于 Windows API 提供的底层服务，例如，菜单、按钮、列表框和标签等基本的 Windows 窗体控件类，以及基本的 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。其规则可以概括如下。

- ❑ 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。
- ❑ 定义了原始数据类型，如 Int32、Int64、Single、Double 和 Boolean。
- ❑ 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。
- ❑ 定义了对支持基于 0 的数组的支持。
- ❑ 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。
- ❑ 定义了事件名和参数传递给事件的规则。