

朱付保 段赵磊 李灿林 著

精通

C# 4.0

程序设计

- **实用：** 从实际应用角度出发，系统地介绍C#编程和调试的基础知识
- **全面：** 全面系统地介绍C#相关的知识，内容涵盖基础知识、核心技术及项目实战
- **实战：** 精心设计的综合案例，详细介绍基于.NET框架和C#进行项目开发的过程

清华大学出版社

精通 C#4.0 程序设计

朱付保 段赵磊 李灿林 著

清华大学出版社
北 京

内 容 简 介

C#语言是一种简单、现代、面向对象和平台独立的高效组件编程语言，作为微软的旗舰编程语言，深受程序员喜爱。本书以 Visual Studio 2010 为开发环境，比较系统地讲述了使用 C#语言进行程序开发从入门到实战应该掌握的各项技术。

全书共分 14 章，在介绍了 C#集成开发环境和 C#语法基础的同时，还详细介绍了面向对象程序设计方法、异常处理与程序调试、Windows 窗体应用程序设计、图形设计、文件操作、多线程编程、数据库访问编程和基于组件的编程等内容，最后通过干部信息管理系统和快餐 POS 系统两个实例阐述了应用 C#程序设计语言实战开发具体项目的过程，本书配有大量的图片和翔实的设计文档及代码，便于读者对系统的深入理解和自行上机练习。

本教程面向 C#程序开发人员，内容翔实、结构合理、由浅入深、示例丰富、语言简洁流畅。适合作为高等院校本/专科计算机及其相关专业的 C#程序设计教材，同时也适合作为各种 C#编程培训班的教材及 C#程序开发人员的参考资料。

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

精通 C#4.0 程序设计 / 朱付保，段赵磊，李灿林著. —北京：清华大学出版社，2014
ISBN 978-7-302-35141-2

I. ①精… II. ①朱… ②段… ③李… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 012434 号



责任编辑：袁金敏

封面设计：陈宇超

责任校对：徐俊伟

责任印制：王静怡

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 装 者：北京密云胶印厂

经 销：全国新华书店

开 本：185mm×260mm

印 张：24.75

字 数：587 千字

版 次：2014 年 6 月第 1 版

印 次：2014 年 6 月第 1 次印刷

印 数：1~3500

定 价：49.00 元

产品编号：054081-01

前 言

C#是微软公司发布的一种面向对象的、运行于.NET Framework 之上的高级程序设计语言。利用 C#语言和基于.NET 框架的 Visual Studio 2012 集成开发环境,程序员可以方便快捷地开发出各种安全可靠的 Windows 应用程序和 Web 应用程序。目前 C#在应用型软件开发和城市信息化等领域已得到了广泛应用。

本书从实际应用角度出发,系统地介绍了 C#编程和调试的基础知识,并通过案例介绍基于.NET 框架和 C#进行项目开发的详细过程。全书共分 14 章:

第 1 章讲述了 Visual Studio 2012 集成开发环境的使用及创建 C#应用程序的操作步骤。

第 2 章讲述了 C#编程所需要的基本工具:数据类型、常量和变量的声明和使用、运算符和表达式的使用、流程控制语句的使用等。

第 3 章介绍了面向对象编程的基本思想,讲述了类的声明方式及类中数据成员和方法成员的定义方法,类的构造函数和属性、索引等基本概念。

第 4 章讲述了面向对象编程中的继承和派生、多态、抽象类、接口、委托和事件、泛型等高级概念及其使用技巧。

第 5 章讲述了异常处理语句的用法和常用的程序调试方法。

第 6 章介绍了窗体的属性、方法、事件和常用控件,讲述了 Windows 窗体应用程序的设计步骤。

第 7 章介绍了如何使用 GDI+在 Windows 窗体上绘制图形和显示文字。

第 8 章讲述了 C#中的文件和文件夹操作及数据的读写访问。

第 9 章介绍了进程与线程的基本概念和操作,以及如何使用多线程技术构建应用程序。

第 10 章介绍了基于 ADO.NET 的数据库访问技术。

第 11 章介绍了组件和控件的相关概念和开发方法。

第 12 章和第 13 章介绍了干部信息管理系统、快餐 POS 系统两个 C#综合应用实例。

第 14 章介绍了 Windows 应用程序的部署工具 Windows Installer 及相关的部署流程。

本书内容翔实,图文并茂,条理清晰,通俗易懂,示例丰富,在讲解每个知识点时都配有相应的实例方便读者上机实践。本书在编写时融入了作者多年的开发经验,配有大量综合实例和练习,帮助读者在不断的实际操作中更加牢固地掌握书中讲解的内容。本书除了可用作高等院校本、专科学生的教材外,兼顾一般读者,可作为从事计算机应用开发人员在学习 C#编程时的参考书。

全书由朱付保老师统稿,其中第 1、2 章由黄艳编写,第 3、4 章由段赵磊编写,第 5、9 章由吴雪丽编写,第 6、8 章由李灿林编写,第 7、10 章由吴庆岗编写,第 11、12、13 章由朱付保编写。徐显景、杨鹏、钱慎一、王辉、姚妮参与了本书其他章节内容的编写,

并对全书资料的整理与内容校对付出了辛勤的汗水，在此一并表示感谢。最后感谢郑州轻工业学院教务处的的大力支持。

由于编写时间仓促，加之作者水平有限，书中难免会疏漏和错误之处，恳请广大读者不吝赐教并给予批评指正。

目 录

第 1 章 概述	1
1.1 .NET Framework 概述	1
1.1.1 .NET 平台简介	1
1.1.2 .NET Framework	4
1.1.3 .NET 程序的编译和执行	8
1.1.4 C#与.NET Framework	9
1.2 C#集成开发环境 VS 2010	11
1.2.1 启动 VS 2010 开发环境	11
1.2.2 新建项目	12
1.2.3 VS 2010 主窗口	13
1.2.4 帮助系统	16
1.3 创建简单的 C#应用程序	18
1.3.1 创建简单的 C#控制台应用程序	19
1.3.2 C#控制台应用程序的基本结构	21
1.3.3 创建简单的 Windows 窗体应用程序	23
1.3.4 Windows 窗体应用程序的基本结构	25
1.4 本章小结	27
习题	28
第 2 章 C#程序设计基础	29
2.1 C#数据类型	29
2.1.1 值类型	29
2.1.2 引用类型	35
2.1.3 数据类型转换	42
2.2 变量和常量	45
2.2.1 变量的声明和使用	46
2.2.2 变量的分类	46
2.2.3 常量	48
2.3 常用运算符和表达式	48
2.3.1 运算符	48
2.3.2 表达式	51
2.4 C#方法及其重载	51
2.4.1 方法的定义	52
2.4.2 方法的调用	54

2.4.3 方法的重载	54
2.5 C#流程控制语句	55
2.5.1 条件分支语句	55
2.5.2 循环控制语句	57
2.5.3 跳转语句	59
2.6 控制台的输入和输出	60
2.7 常见的预处理指令	62
2.8 本章小结	64
习题	64
第3章 面向对象编程基础	66
3.1 软件开发方法	66
3.1.1 结构化程序设计方法	66
3.1.2 面向对象的开发方法	66
3.1.3 面向对象程序设计方法	67
3.2 类	67
3.2.1 类的声明	67
3.2.2 创建对象	68
3.2.3 构造函数	69
3.3 类的成员	72
3.3.1 类的数据成员	72
3.3.2 类的方法成员	74
3.3.3 类的属性成员	77
3.3.4 索引指示器	81
3.3.5 析构函数	83
3.4 C#常用类操作	84
3.4.1 Convert 类	84
3.4.2 string 类和 StringBuilder 类	85
3.4.3 DateTime 类和 TimeSpan 类	90
3.4.4 Math 类	91
3.5 本章小结	91
习题	92
第4章 面向对象高级编程	94
4.1 继承与派生	94
4.1.1 继承现象	94
4.1.2 隐藏基类成员	96
4.1.3 派生类的构造函数	97
4.2 多态性	99
4.2.1 多态性的重要性	99
4.2.2 虚方法	100

4.2.3 多态的实例	101
4.3 抽象类与抽象方法	102
4.3.1 抽象类	102
4.3.2 抽象方法	103
4.4 密封类与密封方法	104
4.4.1 密封类	104
4.4.2 密封方法	104
4.5 接口	105
4.5.1 接口的声明	106
4.5.2 接口成员的声明	107
4.5.3 接口成员的访问	107
4.5.4 接口的实现	107
4.6 委托与事件	109
4.6.1 委托	109
4.6.1 事件	111
4.7 序列化与反序列化	112
4.7.1 二进制序列化	113
4.7.2 XML 序列化	114
4.8 泛型处理	115
4.8.1 泛型类的定义	115
4.8.2 泛型的引用	116
4.8.3 常用泛型	117
4.9 本章小结	118
习题	119
第 5 章 异常处理与程序调试	120
5.1 异常处理	120
5.1.1 为什么需要异常处理	120
5.1.2 异常处理机制	121
5.1.3 系统的异常类及其使用	122
5.1.4 try / catch / finally 语句块	124
5.1.5 抛出异常	127
5.2 程序调试	128
5.2.1 断点设置	128
5.2.2 启动、中断、继续和停止程序调试	130
5.2.3 监视调试状态	132
5.3 本章小结	133
习题	134
第 6 章 Windows 窗体应用程序设计	135
6.1 窗体与控件	135

6.1.1	窗体的常用属性	135
6.1.2	窗体的常用方法和事件	139
6.1.3	主要的窗体控件概述	140
6.2	基本控件	142
6.2.1	按钮与标签控件	142
6.2.2	文本控件	147
6.2.3	选择控件	152
6.2.4	列表控件	155
6.2.5	容器控件	161
6.2.6	高级列表选择控件	165
6.3	菜单、工具栏和状态栏	172
6.3.1	菜单	172
6.3.2	工具栏	176
6.3.3	状态栏	178
6.4	对话框控件及其设计	179
6.4.1	模态对话框与非模态对话框	179
6.4.2	通用对话框	179
6.4.3	消息对话框	187
6.5	键盘与鼠标事件处理	190
6.5.1	焦点处理	190
6.5.2	键盘事件处理	191
6.5.3	鼠标事件处理	191
6.6	本章小结	192
	习题	192
第 7 章	图形设计	194
7.1	绘图概述	194
7.1.1	绘图的基本知识	194
7.1.2	什么是 GDI+	195
7.1.3	Graphics 类	195
7.2	绘图流程	197
7.3	创建画图工具	197
7.3.1	创建画笔	197
7.3.2	创建画刷	198
7.4	绘制空心图形	200
7.4.1	绘制直线	200
7.4.2	绘制矩形	201
7.4.3	绘制多边形	202
7.4.4	绘制圆和椭圆	203
7.4.5	绘制圆弧	205

7.4.6 绘制饼形	206
7.4.7 绘制非闭合曲线	207
7.4.8 绘制闭合曲线	209
7.4.9 绘制贝塞尔曲线	210
7.5 绘制填充图形	212
7.5.1 填充矩形	212
7.5.2 填充多边形	213
7.5.3 填充圆和椭圆	214
7.5.4 填充饼形	215
7.5.5 填充闭合曲线	216
7.6 绘制文本	218
7.7 本章小结	219
习题	219
第 8 章 文件操作	221
8.1 文件	221
8.1.1 文件类型	221
8.1.2 文件的属性	222
8.1.3 文件访问方式	222
8.2 System.IO 模型	222
8.2.1 什么是 System.IO 模型	222
8.2.2 文件编码	223
8.2.3 C#的文件流	223
8.3 文件夹和文件操作	224
8.3.1 文件夹操作	224
8.3.2 文件操作	225
8.4 FileStream 类	228
8.5 文本文件的操作	229
8.5.1 StreamReader 类	229
8.5.2 StreamWriter 类	230
8.6 二进制文件操作	232
8.6.1 BinaryReader 类	232
8.6.2 BinaryWriter 类	233
8.6.3 二进制文件的随机查找	235
8.7 本章小结	235
习题	235
第 9 章 多线程技术	237
9.1 进程与线程	237
9.1.1 进程的基本概念	237
9.1.2 线程的基本概念	237

9.1.3	多线程	238
9.1.4	Thread 类	238
9.1.5	前台线程与后台线程	239
9.1.6	线程的状态	239
9.2	线程的基本操作	239
9.2.1	启动线程	240
9.2.2	暂停线程	241
9.2.3	合并线程	241
9.2.4	终止线程	241
9.2.5	Volatile 关键字	242
9.2.6	在一个线程中访问另一个线程的控件	244
9.3	线程优先级	247
9.4	线程同步	248
9.5	本章小结	251
	习题	251
第 10 章	ADO.NET 数据库编程	253
10.1	数据库基础知识	253
10.1.1	数据库常用术语	253
10.1.2	关系型数据库	254
10.2	SQL 基础知识	256
10.2.1	SQL 简介	256
10.2.2	插入语句 (INSERT)	257
10.2.3	查询语句 (SELECT)	258
10.2.4	删除语句 (DELETE)	260
10.2.5	更新语句 (UPDATE)	260
10.3	ADO.NET 概述	260
10.3.1	ADO.NET 简介	260
10.3.2	ADO.NET 对象模型	261
10.4	利用 ADO.NET 访问数据库	262
10.4.1	Connection 对象	262
10.4.2	Command 对象	265
10.4.3	DataReader 对象	267
10.4.4	DataAdapter 对象	271
10.4.5	DataSet 对象	273
10.4.6	数据绑定	276
10.5	本章小结	282
	习题	283
第 11 章	组件技术	284
11.1	概述	284

11.2 组件和控件	284
11.3 组件	285
11.3.1 组件的创建	285
11.3.2 组件的测试	287
11.4 用户控件	289
11.4.1 用户控件的创建	289
11.4.2 用户控件的测试	291
11.5 本章小结	292
习题	292
第 12 章 干部信息管理系统	293
12.1 系统分析	293
12.1.1 需求分析	293
12.1.2 功能模块的划分	296
12.2 数据库设计	296
12.2.1 数据库的需求分析	297
12.2.2 数据的逻辑设计	297
12.3 公共类设计	302
12.3.1 数据库连接及操作类	302
12.3.2 配置文件读写类	305
12.4 系统的实现	306
12.4.1 系统登录模块	306
12.4.2 单位注册模块	307
12.4.3 文件管理模块	311
12.4.4 综合管理模块	316
12.4.5 数据分析模块	326
12.4.6 打印管理模块	331
12.4.7 用户管理模块	333
12.5 本章小结	334
习题	334
第 13 章 快餐 POS 系统	335
13.1 系统分析	335
13.1.1 需求分析	335
13.1.2 功能模块的划分	336
13.2 数据库设计	337
13.2.1 数据库的需求分析	337
13.2.2 数据的逻辑设计	338
13.3 公共类设计	346
13.3.1 数据库连接及操作类	346
13.3.2 配置文件读写类	349

13.4 系统的实现	350
13.4.1 登录模块	351
13.4.2 上线模块	352
13.4.3 系统主窗体	353
13.4.4 滚动字幕模块	354
13.4.5 当前信息模块	354
13.4.6 点单模块	356
13.4.7 功能模块	357
13.4.8 数字模块	358
13.4.9 点餐模块	361
13.4.10 商品分类模块	361
13.4.11 商品模块	362
13.4.12 修改密码模块	364
13.4.13 查询模块	365
13.4.14 下线模块	366
13.5 本章小结	368
习题	368
第 14 章 部署 Windows 应用程序	369
14.1 应用程序部署概述	369
14.1.1 VS 2010 提供的应用程序部署功能	369
14.1.2 部署前的工作准备	369
14.2 使用 Windows Installer 部署 Windows 应用程序	369
14.2.1 创建安装程序	370
14.2.2 测试安装程序	380
14.3 本章小结	381
习题	381

第1章 概述

C#是微软公司推出的一种面向.NET平台的、类型安全的面向对象编程语言,利用C#语言和基于.NET框架的Visual Studio 2010集成开发环境,程序员可以方便快捷地开发出各种安全可靠的应用程序。本章将对.NET平台的相关内容作简单介绍,并通过图文并茂的方式介绍Visual Studio 2010集成开发环境及创建两种类型的C#应用程序的操作步骤。通过本章的学习,读者会对C#语言和Visual Studio 2010集成开发环境有一个初步了解,并能够顺利地创建简单的C#应用程序。

1.1 .NET Framework 概述

2000年6月22日,微软公司正式对外宣布了.NET战略。同年11月,微软在COMDEX计算机大展上发布了Visual Studio .NET软件,全面推进了.NET技术向市场进军的步伐。C#语言是微软公司针对.NET平台推出的主流语言,它不但继承了C++、Java等面向对象语言的强大功能特性,同时还继承了VB、Delphi等编程语言的可视化快速开发功能,是当前第一个完全面向组件的语言。作为.NET平台的第一语言,C#语言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。本节主要介绍与C#语言密切相关的.NET平台和Visual Studio 2010集成开发环境。

1.1.1 .NET 平台简介

微软总裁兼首席执行官 Steve Ballmer 给.NET下的定义为:“.NET 代表一个集合,一个环境,一个可以作为平台支持下一代 Internet 的可编程结构。”。即.NET = 新平台 + 标准协议 + 统一开发工具。作为微软的集成开发平台,.NET 技术提供迅速修改、部署、处理并且使用连接的能力,提高了WEB服务的高效性;同时,.NET 技术也使创建稳定、可靠而又安全的 Windows 桌面应用程序更为容易。下面简单了解.NET的发展历程。

1. .NET 平台的发展历程

2000年6月22日,比尔·盖茨向全球宣布其下一代软件和服务,即Microsoft .NET平台的构想和实施步骤。2000年微软的白皮书这样定义.NET:Microsoft .NET是Microsoft XML Web Services平台。XML Web Services允许应用程序通过Internet进行通信和共享数据,而不管所采用的是哪种操作系统、设备或编程语言。Microsoft.NET平台提供创建XML Web Services并将这些服务集成在一起之所需。

2002年2月13日微软正式发布了Visual Studio.NET 2002,其中包含了.NET Framework

1.0,除了引入一门全新的语言 C#外,同时还提供了对于 Java 的支持。C#大量借鉴了 Java 的语法,同时保留了 VB 方面的诸多便利性。ASP.NET 作为平台的关键组成部分,传承了微软一直以来的可视化设计风格,允许开发人员以拖放方式开发 Web 应用。然而.NET 1.0 作为全新的平台,许多类库仍然还不成熟。

2003 年 4 月 25 日,曾被命名为 Windows .NET Server 的操作系统 Windows Server 2003 正式发布,同日还发布了 Visual Studio .NET 2003,并将 .NET Framework 的版本升级到了 1.1.4322。Windows Server 2003 是微软发展史上一个非常重要的里程碑:一方面 Windows 操作系统在企业级应用的能力得到证实;另一方面 .NET 终于完成了和 Windows 操作系统的无缝集成,也真正意义上为开发人员提供了一套完整的 .NET 解决方案。

Visual Studio.NET 2003 为程序开发人员提供了统一的开发语言和开发界面,不管开发桌面应用,还是 Web 应用,或者是手机设备的应用,Visual Studio.NET 使开发人员能够在不同应用开发中自由切换。同时,随着 .NET Framework 的稳定,微软内部越来越多的产品采用 .NET 重新开发,或者提供了和 .NET 的无缝对接,例如,2003 年发布的 Exchange 2003、Office 2003 及 2004 年发布的 Biztalk Server 2004,都允许开发人员使用 .NET 开发应用,并且做到了无缝集成。

2005 年 10 月 27 日,微软将 Visual Studio.NET 重新命名为 Visual Studio 2005,同时将 .NET Framework 的版本升级到 2.0。另外,为了方便开发人员应用,内置了一个用于开发调试的 Web 服务器,使得开发人员在开发过程可以更加方便地测试与部署。同日发布的 SQL Server 2005 完全架构在 .NET 之上,并允许开发人员使用 .NET 编写存储过程、函数及用户自定义类型 (UDT)。

2006 年,微软将 WPF(Windows Presentation Foundation)、WCF(Windows Communication Foundation)、WWF(Windows Workflow Foundation)和 Windows Cardspace 整合成代号为“WinFX”的 .NET Framework 3.0,并于 2006 年 11 月 6 日发布。.NET Framework 3.0 的发布是对 .NET Framework 2.0 的一个重要补充,它弥补了微软在企业级开发的软肋。

2007 年 11 月 19 日,微软发布了 Visual Studio 2008,随同发布了 .NET Framework 3.5。.NET Framework 3.5 引入了 Linq 和 X.Linq 技术,Linq 和 X.Linq 为开发人员带来了激动人心的编程体验。开发人员可以混合对象与数据,然后用同样的查询方式进行数据处理,更重要的是允许开发人员在任意环节进行扩展,从而帮助开发人员以一致的方式进行数据处理。

2010 年 4 月 12 日,微软发布了 Visual Studio 2010 及 .NET Framework 4.0。.NET Framework 4.0 包括更好的多核心支持、后台垃圾回收和服务器的探查器附加,增加了新的内存映射文件和数字类型,并支持新的动态数据功能,包括新的查询筛选器、实体模板、对 Entity Framework 4 更丰富的支持及可轻松应用于现有 Web 窗体的验证和模板化功能等。

2012 年 8 月 16 日,微软发布了最新版本 Visual Studio 2012 及 .NET Framework 4.5。.NET Framework 4.5 是一个针对 .NET Framework 4.0 的高度兼容的就地更新。.NET Framework 4.5 包括针对 C#、Visual Basic 和 F# 的重大语言和框架改进(以便程序员能够更轻松地编写异步代码)、同步代码中的控制流混合、可响应 UI 和 Web 应用程序可扩展性,并提供比 .NET Framework 4.0 更高的性能、可靠性和安全性。

2. .NET 平台的组成

众所周知,微软的灵魂产品 Windows 操作系统是硬件设备和软件运行环境的平台,它消除了不同硬件设备之间的差别,使外部设备都变成了可以自由使用、无缝集成的一个整体。与 Windows 操作系统类似,微软推出的 .NET 平台能够消除互连环境中不同硬件、软件、服务的差别,使不同的设备、不同的操作系统都可以相互通信,使不同的程序和服务之间都可以相互调用。

.NET 平台几乎包含了微软正在研发或已经得到广泛应用的各种软件开发技术。对 .NET 程序员来说,应主要关心 .NET 平台的以下几个组成部分。

(1) .NET Framework: 微软推出的一种运行于各操作系统之上的新软件运行平台,提供了 .NET 程序运行时支持和功能强大的类库,是其他所有 .NET 技术产品的坚实基础。只要安装了 .NET Framework, 则从 Windows 98 到 Windows XP 都可以运行 .NET 程序。

(2) .NET 编程语言: .NET 平台支持 20 多种编程语言,传统的各种编程语言有许多都已经或正在被移植到 .NET 平台,目前 .NET 平台支持的编程语言种类仍在不断增加。目前,由微软公司提供的 .NET 编程语言主要有 Visual Basic.NET (改进过的 Visual Basic)、C++、C# 和 F#。

(3) Visual Studio.NET 集成开发环境: 用来开发、测试和部署应用程序。Visual Studio.NET 历经微软公司持续多年的完善,已经成为世界一流的“软件集成开发环境 (Integrated Development Environment, IDE)”。

(4) .NET 软件产品: 几乎微软公司所有主要软件产品都基于 .NET Framework 或包容 .NET 技术,包括 Windows 操作系统、SQL Server 数据库服务器、Office 商业应用开发与运行平台和 Azure 云计算平台等。

3. .NET 技术前景

从 2002 年发布 .NET 1.0, 历经 11 年发展, .NET 版本已经发展到了 4.5。 .NET 是一个庞大而复杂的软件开发与运行平台,包含了“一堆”的子技术领域。

(1) 桌面应用程序开发技术。

在很长一段时间内, Windows Form 成为 .NET 桌面领域的主流技术,而且有一大批各式各样的第三方控件,其功能可谓应有尽有,使用方便。然而 .NET 3.0 中出现的 WPF,在界面设计和用户体验上比 Windows Form 要强得多,比如,其强大的数据绑定、动画、依赖属性和路由事件机制等。WPF 的性能在 .NET 4.0 上有了进一步的改进。WPF 相对于 Windows 客户端的开发来说,向前跨出了巨大的一步,它提供了超丰富的 .NET UI 框架,集成了矢量图形,丰富的流动文字支持 flow text support, 3D 视觉效果和强大无比的控件模型框架。

(2) 数据存取技术。

.NET 平台融合了 ADO.NET、LINQ 和 WCF Data Service 等数据存取技术。ADO.NET 不仅提供了对 XML 的强大支持,还引入了一些新的对象,如驻于内存的数据缓冲区 DataSet、用来高效率读取数据并一个只读的记录集的 DataReader 等。LINQ (Language Integrated Query, 语言集成查询) 是 Visual Studio 2008 和 .NET Framework 3.5 版中引入的

一项创新功能，它在对象领域和数据领域之间架起了一座桥梁。LINQ 是编程语言的一个组成部分，在编写程序时可以得到很好的编译时语法检查、丰富的元数据、智能感知、静态类型等强类型语言的好处。同时，LINQ 还可以方便地对内存中的信息进行查询而不仅仅只是对外部数据源进行查询。WCF Data Service 原称 ADO.NET Data Service，体现了“数据是一种服务”的思想，让数据可以通过 HTTP 请求直接获取。WCF Data Service 设计了一套 URI 模式，可以完成投影、选择和分页等功能，用起来方便灵活。

(3) Web 开发技术。

.NET 平台底层使用 ADO.NET 实体框架或 LINQ to SQL 构造数据模型，通过提取数据模型中的元数据，动态选择合适的模板生成网页，避免了真实项目中不得不为每个数据存取任务设计不同网页的负担，而且提供了很多方式允许用户定制网站。.NET 平台的另一种 Web 应用架构代表技术 Silverlight 充分利用客户端的计算资源，大大地降低了对服务端的依赖，并且易于构造良好的用户体验。

(4) 插件技术。

.NET 4.0 引入了 Managed Extensibility Framework (MEF) 技术。MEF 通过简单地给代码附加 “[Import]” 和 “[Export]” 标记就可以清晰地表明组件之间的“服务消费”与“服务提供”关系，并在底层使用反射动态地完成组件识别、装配工作从而使得开发基于插件架构的应用系统变得简单。

(5) 函数式编程语言 F#。

F# 是微软 .NET 平台上一门新兴的函数式编程语言，通过 F#，开发人员可以轻松应对多核多并发时代的并行计算和分布问题。

1.1.2 .NET Framework

.NET Framework 是 .NET 平台的关键组件，提供了 .NET 程序运行时支持和功能强大的类库。从开发各种应用程序的程序员角度来看，.NET Framework 用易于理解与使用的面向对象方式调用 Windows 操作系统所提供的各种系统功能。.NET Framework 在整个软件体系结构中的地位如图 1-1 所示。.NET Framework 在应用程序和操作系统之间起到承上启下的作用，向内包容着操作系统内核，向外给运行于其上的 .NET 应用程序提供访问操作系统核心功能的服务。在 .NET Framework 下编程，程序员不再需要与各种复杂的 Windows API 函数打交道，只需使用现成的 .NET Framework 类库即可。

.NET Framework 的体系结构如图 1-2 所示，它主要由公共语言运行库 (CLR, Common Language Runtime) 和 .NET Framework 类库构成。CLR 是 .NET Framework 的核心执行环境，也称为 .NET 运行库。CLR 是一个技术规范，无论程序使用什么语言编写，只要能编译成微软中间语言 MSIL (Microsoft Intermediate Language)，就可以在它的支持下运行。这意味着在不久的将来，可以在 Windows 环境下运行传统的非 Windows 语言。而 .NET Framework 类库是一个由 Microsoft .NET Framework SDK 中包含的类、接口和值类型组成的库，提供对系统功能的访问，是建立 .NET Framework 应用程序、组件和控件的基础，该库可以完成以前要通过 Windows API 来完成的绝大多数任务。