



21世纪高等学校计算机
专业实用规划教材

Visual C#.NET 基础与应用教程 (第2版)

◎ 夏敏捷 罗 菁 主编
李国伟 陈海蕊 副主编
郑秋生 主审

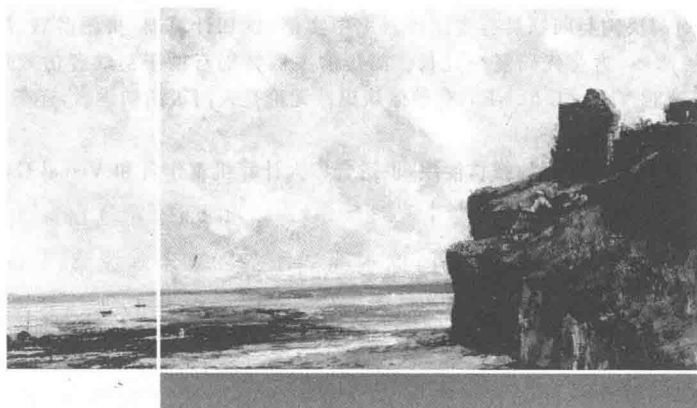


清华大学出版社





21世纪高等学校计算机
专业实用规划教材



Visual C#.NET 基础与应用教程 (第2版)

◎ 夏敏捷 罗 菁 主编
李国伟 陈海蕊 副主编

清华大学出版社
北京

内 容 简 介

本书全面介绍了 Visual C#.NET 基础知识、控件、文件处理和系统操作、多媒体编程、网络编程、数据库编程以及 Web 应用程序开发和 XML 技术,并以实例的形式向读者展示了 Visual C#.NET 的编程精髓,将 Visual C#.NET 编程中的语法、技巧等方面的知识融入其中,由浅入深,从易到难。这些实例典型简洁,所涉及的技术对解决同类问题具有实用性。书中实例(例如计算器、拼图游戏、网络象棋开发、五子棋等)贴近读者、讲解清晰、力避代码复杂冗长。简短的案例特别有助于初学者仿效理解、把握问题的精髓;能够帮助读者快速对 Visual C#.NET 有整体认识。无论是入门级的初学者,还是有一定基础的读者,都可以在本书中得到有用的东西。

本书可作为高等院校相关课程的教材使用,也适合广大计算机工作者和 Visual C#.NET 用户编程爱好者、软件开发参考。

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

Visual C#.NET 基础与应用教程/夏敏捷,罗菁主编.—2 版.—北京:清华大学出版社,2017
(21 世纪高等学校计算机专业实用规划教材)
ISBN 978-7-302-45412-0

I. ①V… II. ①夏… ②罗… 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(2016)第 260153 号

责任编辑:刘 星

封面设计:刘 键

责任校对:李建庄

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:28.25 字 数:686 千字

版 次:2014 年 5 月第 1 版 2017 年 1 月第 2 版 印 次:2017 年 1 月第 1 次印刷

印 数:1~2000

定 价:49.50 元

产品编号:069952-01

前 言

为什么学习 Visual C#.NET?

DotNET(.NET)是微软未来的技术发展方向,其强大的技术优势为人们所推崇,并且在全世界掀起了学习 DotNET 技术的高潮,掌握该技术,无疑在目前激烈的就业竞争中把握了有力武器。作为微软 DotNET 框架下的核心技术之一,Visual C#.NET(简称 C# 语言)经过几年的发展,已经成为主流开发语言。

C 和 C++一直是最有生命力的程序设计语言。这两种语言为程序员提供了丰富的功能、高度的灵活性和强大的底层控制能力,而这一切都不得不以牺牲效率作为代价。例如与 Visual C#.NET 相比,Visual C++程序员为实现同样的功能就要花费更长的开发周期。C 和 C++既为我们带来了高度的灵活性,又使我们必须要忍受学习的艰苦和开发的长期性,特别对 Visual C++来说,大部分的程序结构都被封装在 MFC 中,对于初学者来说,程序结构显得十分混乱,学习将变得十分艰苦。

Visual C#.NET 程序结构十分清晰,较易学习和使用,同时又不失灵活性和强大的功能,它在开发能力和效率之间取得较好的平衡。它不仅具有快速开发应用程序的能力,而且具有 C++的基本特征——面向对象,Visual C#.NET 已成为功能强大的面向对象的编程语言。

本书作者长期从事 Visual C#.NET 教学与应用开发,在长期的工作与学习中,积累了丰富的经验和教训,能够了解在学习编程的时候需要什么样的知识才能提高 C# 开发能力,以最少的时间投入得到最快的实际应用。

本书内容共 10 章,各章内容如下:

第 1 章主要介绍 .NET 框架和 Visual Studio 2013.NET 集成开发环境,同时介绍了 Visual Studio.NET 集成开发环境中如何创建 C# 三种应用程序等。

第 2 章主要介绍 Visual C#.NET 语言数据类型、流程控制语句。

第 3 章介绍面向对象的基本概念,包括类和对象以及需要重点掌握的面向对象的继承性、多态性思想和具体体现。

第 4 章主要介绍常用控件,同时展示用 Windows 窗体来编写程序的特点以及技巧。

第 5 章介绍利用 .NET 框架提供的一整套图形类库,绘制各种图形、处理位图图像和视频,从而建立图形游戏程序。

第 6 章主要介绍 Visual C#.NET 语言提供的用于文件操作的类,以及如何利用它们

实现对文件的存储管理、对文件的读写等各种操作。

第7章主要介绍利用.NET框架类库中提供的应用层类 TcpClient、TcpListener 和 UdpClient 类来实现网络编程的知识。本章最后通过应用层类开发出基于 UDP 的网络中国象棋。

第8章在 ADO.NET模型的基础上介绍如何操作数据库,读者可以熟悉掌握 ADO.NET 中各种对象的操作方法以及常用 SQL 语句,并能够读、写、检索数据库。

第9章主要介绍开发 Web 应用程序的 ASP.NET 工作原理和 ASP.NET 常用控件,在 Web 应用程序中访问数据库等。本章最后通过母版技术创建网络游戏网站。

第10章介绍.NET框架中与 XML 相关的命名空间和其中的重要类及 DOM 技术,并用实例使读者更进一步了解 XML 文件的 C# 读写操作的具体方法。

需要说明的是,学习编程是一个实践的过程,而不仅仅是看书、看资料的过程,亲自动手编写、调试程序才是至关重要的。通过实际的编程以及积极的思考,读者可以很快掌握很多的编程技术,而且,在编程中读者会积累许多宝贵的编程经验。在当前的软件开发环境下,这种编程经验对开发者尤其显得不可或缺。

本书获得中原工学院教材建设立项并资助,由夏敏捷(中原工学院)、罗菁(中原工学院)主持编写,陈海蕊(中原工学院)、李娟(中原工学院)编写第2章和第8章,张锦歌(河南工业大学)编写第4章,李国伟(中原工学院)编写第6章,罗菁(中原工学院)编写第9章,其余章节由夏敏捷编写。在本书的编写过程中,为确保内容的正确性,参阅了很多资料,并且得到了中原工学院计算机学院郑秋生教授的指导和一些资深 C# 程序员的支持。在此谨向他们表示衷心的感谢。

本书提供电子课件和程序代码,可以到清华大学出版社网站本书页面下载。由于编者水平有限,书中难免有错,敬请广大读者批评指正,在此表示感谢。电子邮件地址: xmj@zut.edu.cn。

夏敏捷

2016年10月

目 录

第 1 章 Visual C#.NET 概述	1
1.1 Visual C#.NET 简介	1
1.1.1 Visual C#.NET 产生	1
1.1.2 Visual C#.NET 的特点	2
1.1.3 .NET 框架	3
1.2 Visual Studio 2013.NET 集成开发环境	4
1.2.1 Visual Studio 2013.NET 的安装	5
1.2.2 Visual Studio 2013.NET 简介	7
1.2.3 Visual Studio 2013.NET 中的其他窗口	12
1.3 Visual C#.NET 的三种应用程序结构	14
1.3.1 Visual C#.NET 编写控制台应用程序	14
1.3.2 Visual C#.NET 编写 Windows 应用程序	16
1.3.3 Visual C#.NET 编写 Web 应用程序	17
1.4 命名空间	19
1.4.1 定义命名空间	19
1.4.2 导入命名空间	20
1.4.3 常用命名空间	20
1.5 Visual C#.NET 应用程序的开发步骤	21
习题	22
第 2 章 Visual C#.NET 编程基础	23
2.1 数据类型	23
2.2 不同数据类型之间的转换	27
2.2.1 显式转换与隐式转换	27
2.2.2 装箱和拆箱	28
2.3 常量和变量	29
2.3.1 常量	29
2.3.2 变量	30
2.3.3 变量的作用范围(作用域)	30
2.4 运算符与表达式	31

2.4.1	运算符	31
2.4.2	运算符优先级	35
2.4.3	表达式	35
2.4.4	C# 4.0 引入动态关键字 dynamic	36
2.5	控制台应用程序与格式化输出	37
2.5.1	控制台输出	37
2.5.2	控制台输入	38
2.5.3	字符串的格式化输出	38
2.6	C# 流程控制语句	39
2.6.1	选择语句	39
2.6.2	循环语句	42
2.6.3	跳转语句	46
2.6.4	异常处理语句	48
2.7	数组	51
2.7.1	数组的声明与初始化	51
2.7.2	创建数组实例	53
2.7.3	一维数组	53
2.7.4	多维数组	58
2.7.5	交错数组	62
2.7.6	数组的方法和属性	63
习题		65
第3章	面向对象的编程基础	67
3.1	类	67
3.1.1	C# 类的声明和对象的创建	67
3.1.2	类的成员	69
3.1.3	类的构造函数和析构函数	69
3.1.4	静态成员和实例成员	71
3.1.5	方法	72
3.1.6	属性与索引器	78
3.1.7	分部类	83
3.2	结构类型	83
3.2.1	结构类型的声明	83
3.2.2	结构变量	84
3.3	类的继承	85
3.3.1	继承	86
3.3.2	抽象类和密封类	95
3.4	多态	97
3.4.1	隐藏基类方法	97

3.4.2	声明虚方法	98
3.4.3	实现多态性	100
3.5	接口	102
3.5.1	定义接口	102
3.5.2	实现接口	102
3.5.3	显式接口成员实现	103
3.6	委托与事件	104
3.6.1	委托	104
3.6.2	事件	107
3.7	反射	110
3.7.1	System.Reflection 命名空间	111
3.7.2	如何使用反射获取类型	111
3.7.3	获取程序集元数据	112
3.8	序列化与反序列化	113
3.8.1	二进制序列化与反序列化	113
3.8.2	XML 序列化与反序列化	116
3.9	.NET 泛型编程	117
3.9.1	为什么要使用泛型	117
3.9.2	定义泛型方法	119
3.9.3	定义泛型类	120
3.9.4	使用泛型集合类	122
3.10	Visual C#.NET 常用类	127
3.10.1	Console 类	127
3.10.2	String 类和 StringBuilder 类	128
3.10.3	DateTime 类和 TimeSpan 类	131
3.10.4	Math 类	131
3.10.5	Convert(转换)类	132
3.10.6	Random 类	133
3.10.7	与窗体应用程序相关的类	134
3.11	集合	135
3.11.1	ArrayList 数组列表	135
3.11.2	Stack 堆栈	139
3.11.3	Queue 队列	141
3.11.4	Hashtable 哈希表和 SortedList 排序列表	142
3.11.5	BitArray 位数组	143
	习题	144

第 4 章	Visual C#.NET 控件及其应用	145
4.1	特殊功能文本框和标签	145

4.1.1	常用属性和事件	145
4.1.2	只能输入数字文本框	146
4.1.3	文本框焦点转移	147
4.1.4	创建口令文本框	147
4.1.5	代码设置文本框的字体	147
4.1.6	只读文本框	147
4.1.7	标签控件	147
4.2	单选按钮应用——模拟单项选择题测试	148
4.2.1	常用属性和事件	148
4.2.2	实例开发	148
4.3	复选框应用——模拟多项选择题测试	151
4.3.1	常用属性和事件	151
4.3.2	实例开发	151
4.3.3	窗体中多页显示效果实现技巧	153
4.4	列表框应用——小学生做加减法的算术练习程序	153
4.4.1	常用属性和事件	153
4.4.2	实例开发	154
4.5	组合框应用——国家名选择	156
4.5.1	常用属性和事件	156
4.5.2	实例开发	157
4.6	Timer 控件用法——飘动窗体	158
4.6.1	常用属性和事件	158
4.6.2	实例开发	159
4.7	图片框应用——图片自动浏览器	160
4.7.1	常用属性和事件	160
4.7.2	实例开发	161
4.7.3	图片的缩放技巧	163
4.8	利用滚动条控件调配颜色	164
4.8.1	滚动条的属性和事件	164
4.8.2	实例开发	164
4.9	TreeView 控件和 ListView 控件——学校系部分层列表	166
4.9.1	TreeView 控件	166
4.9.2	实例开发	168
4.9.3	ListView 控件	169
4.9.4	实例开发	170
4.10	菜单使用	172
4.10.1	创建主菜单	172
4.10.2	实例开发	172
4.10.3	上下文菜单	173

4.10.4	实例开发	173
4.11	对话框控件应用——自己的记事本编辑器程序	174
4.11.1	打开文件对话框控件	174
4.11.2	保存文件对话框控件	175
4.11.3	颜色对话框控件	175
4.11.4	字体对话框控件	176
4.11.5	PrintDialog 控件和 PrintDocument 控件	176
4.11.6	对话框控件应用实例开发	176
4.12	实现控件数组的功能——计算器设计	179
4.12.1	控件数组的建立	179
4.12.2	实例开发	180
	习题	183
第5章	图形图像和多媒体编程	184
5.1	GDI+图形图像绘制	184
5.1.1	GDI+概述	184
5.1.2	坐标	186
5.1.3	Graphics 类	186
5.1.4	画笔 Pen 类和画刷 Brush 类	189
5.1.5	可擦写图形轮廓的实现	192
5.2	图像处理	194
5.2.1	显示图像	194
5.2.2	保存图像	195
5.2.3	图像的平移、旋转和缩放	196
5.2.4	生成数字字符验证码图片	197
5.3	播放声音与视频的文件	200
5.3.1	通过 API 函数播放声音文件	200
5.3.2	ActiveX 控件	201
5.3.3	Windows Media Player 控件播放声音和视频文件	201
5.3.4	无声动画控件(Animation)	204
5.4	特殊形状的窗体界面	205
5.4.1	Region 类和 GraphicsPath 类	205
5.4.2	程序设计的步骤	207
5.5	拼图游戏设计	208
5.5.1	Graphics 类的常用方法	208
5.5.2	程序设计的思路	209
5.5.3	程序设计的步骤	209
5.6	五子棋游戏	213
5.6.1	程序设计的思路	213

5.6.2 程序设计的步骤	214
习题	217
第6章 文件处理和键盘操作	220
6.1 C#目录(文件夹)和文件管理	220
6.1.1 System.IO命名空间	220
6.1.2 目录(文件夹)管理	220
6.1.3 文件管理	223
6.1.4 文件夹浏览器实现	224
6.2 文件的读写	228
6.2.1 FileStream类读写文件	228
6.2.2 文本文件的读写	232
6.2.3 读写二进制文件	237
6.3 处理鼠标和键盘事件	245
6.3.1 处理鼠标相关的事件	245
6.3.2 处理键盘相关的事件	246
习题	249
第7章 网络程序开发	251
7.1 网络通信编程基础	251
7.1.1 Socket套接字简介	251
7.1.2 TCP协议和UDP协议	251
7.1.3 Socket编程原理	252
7.1.4 套接字Socket类编程	253
7.1.5 .NET框架中网络通信的应用层类	257
7.2 使用TcpClient类和TcpListener类实现TCP协议通信	257
7.2.1 TcpClient类和TcpListener类	257
7.2.2 实现的基于TCP协议的局域网通信程序	262
7.3 使用UdpClient类实现UDP协议编程	269
7.3.1 UdpClient类	269
7.3.2 UdpClient类开发UDP程序的过程	271
7.4 基于UDP的网络中国象棋	273
7.4.1 网络中国象棋设计思路	274
7.4.2 网络象棋游戏窗体实现的步骤	280
习题	303
第8章 数据库编程	304
8.1 数据库的基本概念	304
8.1.1 关系数据库与二维表	304

8.1.2	关系数据库的有关概念	305
8.1.3	关系数据库的操作	306
8.2	ADO.NET 数据库访问技术	308
8.2.1	ADO.NET 简介	308
8.2.2	ADO.NET 的核心组件	308
8.2.3	ADO.NET 的联机与脱机数据存取模式	310
8.3	ADO.NET 对象及其编程	311
8.3.1	使用 Connection 对象连接数据源	312
8.3.2	使用 Command 对象执行数据库操作	313
8.3.3	DataReader 对象	314
8.3.4	DataSet 对象	317
8.3.5	DataView 对象	321
8.3.6	DataAdapter 对象	322
8.4	使用 ADO.NET 对数据库进行操作	324
8.4.1	在保持连接的方式下进行数据操作	324
8.4.2	在无状态(脱机)方式下进行数据操作	326
8.4.3	数据绑定	329
8.5	数据库中的图像存取	335
8.5.1	关键技术	335
8.5.2	程序设计的步骤	337
8.6	LINQ 技术及应用	340
8.6.1	什么是 LINQ	341
8.6.2	LINQ 基础	342
8.6.3	LINQ 查询子句	344
8.6.4	操作关系型数据——LINQ to SQL	346
8.6.5	使用 LINQ 操作 DataSet——LINQ to DataSet	351
	习题	352
第 9 章	Web 应用程序开发	354
9.1	Web 窗体与 ASP.NET 内置对象	354
9.1.1	ASP.NET 工作原理	354
9.1.2	Web 窗体页面	354
9.1.3	ASP.NET 常用内置对象	354
9.1.4	统计网站在线人数	358
9.2	ASP.NET 控件	361
9.2.1	ASP.NET 控件概述	361
9.2.2	标签控件 Label	364
9.2.3	Button、ImageButton 和 LinkButton 控件	365
9.2.4	DropDownList 控件和 ListBox 控件	366

9.2.5	Image 控件和 ImageMap 控件	368
9.2.6	文本输入控件	370
9.2.7	复选框和单选钮	372
9.2.8	AdRotator 控件	375
9.2.9	Calendar 控件	376
9.2.10	视图控件	378
9.3	Web 表单验证控件应用	381
9.3.1	RequiredFieldValidator 必须字段验证控件	381
9.3.2	RangeValidator 范围验证控件	382
9.3.3	CompareValidator 比较验证控件	382
9.3.4	RegularExpressionValidator 正则表达式控件	382
9.3.5	CustomValidator 自定义验证控件	383
9.4	数据库的操作——读取、修改表信息	385
9.4.1	连接两种数据库	385
9.4.2	读取数据库	385
9.4.3	数据的添加、删除、修改	386
9.4.4	数据库操作的应用实例	387
9.5	Web 数据显示控件应用——显示表信息	392
9.5.1	Repeater 控件	392
9.5.2	DataList 控件	393
9.5.3	GridView 控件	395
9.5.4	Web 数据显示控件应用	398
9.6	母版页创建游戏网站	401
9.6.1	关键技术	401
9.6.2	程序设计的思路	405
9.6.3	程序设计的步骤	405
9.7	网页间数据的传递	408
9.7.1	用 QueryString 来传送相应的值	408
9.7.2	利用 Session 对象传递或共享数据	409
	习题	410
第 10 章	XML 技术	411
10.1	XML 概念	411
10.1.1	使用 XML 的原因	411
10.1.2	与 XML 有关的命名空间和相关类	413
10.2	使用 ADO.NET 中 DataSet 创建 XML 文件	414
10.3	使用 ADO.NET 中 DataSet 读取 XML 文件	415
10.4	C# 通过 DOM 操作 XML 文档	418
10.4.1	.NET 中处理 XML 文档的方式	418

10.4.2	.NET 中使用 DOM 加载及保存 XML 数据	420
10.4.3	使用 DOM 访问 XML 文件	421
10.4.4	使用 DOM 添加新节点	425
10.4.5	使用 DOM 修改删除节点	426
10.5	基于 XML 的游戏网站留言板	428
10.5.1	程序设计的思路	428
10.5.2	程序设计的步骤	429
	习题	434
	参考文献	435

第 1 章

Visual C#.NET 概述

.NET 是微软公司开发的一种面向网络、支持各种用户终端的开发平台。Visual C#.NET(简称 C# 语言)是微软公司针对 .NET 平台推出的一门新语言,作为 .NET 平台的第一语言,也是微软公司推出的下一代主流程序开发语言。Visual C#.NET 经过不断地发展和更新,极大地扩充了原有的功能,开发速度也进一步提高。微软发布了基于 .NET 框架的可视化应用程序开发工具 Visual Studio .NET 简体中文版,集程序设计、程序编译以及程序调试于一体,并将多种程序设计语言紧密地集成在一起,共同使用一个集成开发环境,大大简化了应用程序的开发过程。

本章主要介绍 .NET 框架的概念以及开发 .NET 应用程序的运行环境 Visual Studio .NET,最后介绍 C# 三种应用程序结构。

1.1 Visual C#.NET 简介

1.1.1 Visual C#.NET 产生

首先,来了解一下 C# 的诞生。C 和 C++ 一直是最有生命力的编程语言,这两种语言提供了强大的功能、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,学习和掌握这两种语言比较困难。而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# 是专门为 .NET 应用而开发的语言,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。C# 语言是微软公司在 2000 年 6 月发布的一种新的编程语言,主要由安德斯·海尔斯伯格(Anders Hejlsberg)主持开发,C# 与 Java 非常相似,它包括了诸如单一继承、界面、与 Java 几乎同样的语法,和编译成中间代码再运行的过程。但是 C# 与 Java 有着明显的不同,它借鉴了 Delphi 的一个特点,与 COM(组件对象模型)是直接集成的,而且它是微软公司 .NET 框架的主角。

Visual C#.NET 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。如面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理等。它在设计、开发程序界面的时候和以前的某些程序开发语言有所不同。它既有 Visual Basic 快速开发的特点,又不乏 C++ 语言强大的功能。所以 C# 将成为最主要的软件开发语言。至今已发展到 6.0 版本,C# 的发展如表 1-1 所示。

表 1-1 C#的发展

版 本	日 期	.NET 框架的版本	Visual Studio 的版本
C# 1.0	2002 年 1 月	.NET Framework 1.0	Visual Studio .NET 2002
C# 1.2	2003 年 4 月	.NET Framework 1.1	Visual Studio .NET 2003
C# 2.0	2005 年 11 月	.NET Framework 2.0	Visual Studio .NET 2005
C# 3.0	2006 年 11 月	.NET Framework 3.5	Visual Studio .NET 2008
C# 4.0	2010 年 4 月	.NET Framework 4.0	Visual Studio .NET 2010
C# 5.0	2012 年 4 月	.NET Framework 4.5	Visual Studio .NET 2012
C# 5.0	2013 年 11 月	.NET Framework 4.5.1	Visual Studio .NET 2013
C# 6.0	2015 年 7 月	.NET Framework 4.6	Visual Studio .NET 2015

1.1.2 Visual C#.NET 的特点

1. 可视化的程序设计

Visual C#.NET 采用可视化的编程方式。所谓“可视化”，指程序设计者利用系统提供的良好的集成开发环境(IDE)，不需要编写大量代码去描述界面上各元素的外观和位置，利用系统提供的大量可视化控件(如文本框、按钮等)，通过直接拖动的方式把控件拖动到界面上相应的位置，所见即所得，非常方便，而且用户界面良好。

2. 面向对象的程序设计思想

Visual C#.NET 采用了面向对象的程序设计思想，它将复杂的设计问题分解为一个一个相对简单的独立问题，分别由不同的对象来完成。构成图形界面的可视化控件可以看作是一个个的对象，如一个按钮、一个文本框、一个列表框等。

在面向对象的程序设计中，对象是一个可操作的实体，每个对象具有各自的属性和方法。为了实现每个对象各自的功能，可以分别针对不同对象编写程序代码。

3. 事件驱动的编程机制

Visual C#.NET 采用了事件驱动的编程方式。系统为每个对象设定了若干特定的事件，每个事件都能驱动执行一段特定的程序代码(事件过程)，这一段代码就是针对该对象功能编写的程序代码。

例如，用鼠标单击命令按钮对象，产生一个 Click(单击)事件，同时调用执行该按钮的 Click 事件过程，实现该按钮的功能。事件过程的执行与否和执行顺序取决于用户的操作。

4. 支持大型数据库的管理和开发

Visual C#.NET 提供了强大的数据管理和存取操作能力，能够开发和管理大型的数据库。从 .NET 开始，数据访问技术在原有 ADO 的基础上发展为 ADO.NET，这是对 ADO 的重新设计和功能扩展，大大提高了数据访问和处理的灵活性。同时，ADO.NET 还可以使用 XML 在应用程序之间以及 Web 网页之间交换数据。

5. 强大的 Web 应用程序开发功能

.NET 框架强调网络编程和网络服务，因此在开发 Web 应用程序方面功能更强大，开发更容易。特别是基于 .NET 框架的 Visual C#.NET，在方便易用的 Web 应用程序开发环境下，可以通过直接编辑 ASP.NET 来开发 Web 应用程序以及 Web 服务。

1.1.3 .NET 框架

微软开发的 .NET 平台的核心思想即体现在 .NET Framework(.NET 框架)上,“它代表了一个集合、一个环境和一个可以作为平台支持下一代 Internet 的可编程结构”。

.NET 框架集为各种应用程序的开发提供了一个有利快捷的平台,目的就是让用户在任何地方、任何时间以及利用任何设备都能访问他们所需要的各种信息、文件和程序。系统对访问过程中的后台处理操作对用户来说是透明的,即用户只需提出请求,就可以直接得到处理结果,而不必关心信息的存储位置以及处理过程。到目前为止 .NET 框架先后经历了 .NET Framework 1.0、.NET Framework 2.0、.NET Framework 3.5、.NET Framework 4.0 和 .NET Framework 4.5 等多个版本。.NET 框架的体系结构如图 1-1 所示。

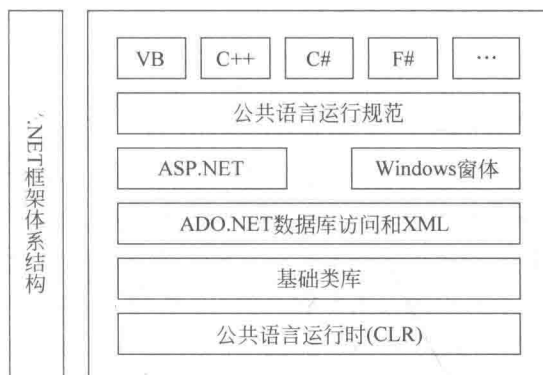


图 1-1 .NET 框架的整体结构

Visual C#.NET 提出了很多新的功能、概念和观点,掌握 Visual C#.NET 不仅要掌握语法,还需要理解并运用这些新的功能、概念和观点。当然也需要掌握 .NET 框架,理解 CLR(Common Language Runtime,公共语言运行时)。

.NET 框架(Framework)是一组用于帮助开发应用程序的类库集。Visual Studio.NET 开发平台需要此类库集的支持,用 Visual Studio.NET 开发的程序也需要此类库集的支持。

1. 公共语言运行时(CLR)

公共语言运行时是 .NET 框架的基础,提供所有的核心服务,如内存的管理和分配,线程和进程的启动、管理和删除等,并且强制实行安全性策略,确保代码运行的安全性和可靠性。与 COM 相比,运行时的自动化程度比较高,如在映射功能上,显著减少了将业务逻辑程序转化为可复用组建的代码编写量,这使得开发人员的工作量大幅减小,开发工作变得相对轻松。

2. 基础类库

.NET Framework 包含了一个综合性的面向对象的可重用的类库集(API),其中含有大量常用功能预先写好的代码,如访问 Windows 基本服务、访问网络、访问数据源等。用户根据需要通过继承可以使用该类库中的所有代码,开发出各种常用的 Windows 应用程序以及基于 ASP.NET 的 Web 应用程序和 Web 服务。

3. ADO.NET 和 XML

为 .NET 提供了统一的数据访问技术。ADO.NET 来源于 ADO,是对 ADO 对象模型