

完全学习手册 ◀

ASP.NET 4.0 程序设计 完全学习手册



刘志勇 黄定光 编著

实用：每个知识点都配以实例进行讲解，让读者摒弃简单枯燥的学习

全面：全面系统地介绍ASP.NET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从实战出发，易学易懂



光盘包含主要知识点的
视频演示及源代码



清华大学出版社

完全学习手册

ASP.NET 4.0 程序设计 完全学习手册

刘志勇 黄定光 编著

清华大学出版社
北 京

内 容 简 介

本书共分 16 章,全面系统地介绍了 ASP.NET 相关的知识点,主要内容包括 C#与.NET Framework 简介、Visual Studio 2010 介绍、C#语言特色、ASP.NET 技术简介、ASP.NET 服务器控件、ASP.NET 应用程序状态管理、异常处理和日志追踪、ASP.NET 中的样式、主题和母版页、ASP.NET 中的站点导航技术、ASP.NET Ajax 开发技术、ADO.NET 数据访问技术、ASP.NET 数据绑定技术、数据绑定控件、成员和角色管理技术等内容。最后通过家教预约系统和仓库出入库管理系统两个案例的开发,对前面知识点进行了综合运用。

本书是针对 ASP.NET 初学者量身打造的一本实用教材,既可以作为广大 ASP.NET 爱好者自学使用,也可以作为计算机相关专业教材,对于 ASP.NET 开发人员也是一本非常不错的参考书。

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

ASP.NET 4.0 程序设计完全学习手册 / 刘志勇, 黄定光编著. —北京: 清华大学出版社, 2014
(完全学习手册)

ISBN 978-7-302-35293-8

I. ①A… II. ①刘… ②黄… III. ①网页制作工具—程序设计—手册 IV. ①TP393.092-62

中国版本图书馆 CIP 数据核字 (2014) 第 018870 号

责任编辑: 袁金敏

封面设计: 刘新新

责任校对: 胡伟民

责任印制: 沈 露

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: 北京富博印刷有限公司

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: 28.5

字 数: 715 千字

附光盘

版 次: 2014 年 7 月第 1 版

印 次: 2014 年 7 月第 1 次印刷

印 数: 1~3000

定 价: 69.00 元



前 言

在 .NET Framework 的基础上构建的 ASP.NET 开发 Web 平台，可提供构建基于企业级服务器的 Web 应用程序所必需的所有服务。在 ASP.NET 中，可使用与公共语言运行时（CLR）兼容的任何语言（包括 Microsoft Visual Basic 和 C#）编写应用程序。同时，ASP.NET 技术作为目前开发 Web 应用程序的最流行技术之一，具有简单易学、跨平台性、可管理性等特点。因此，备受广大开发者的喜爱，为了能够让广大 ASP.NET 爱好者迅速掌握这项技术，我们特意策划编写了本书。

本书共 16 章，各章主要内容如下。

第 1 章 C# 与 .NET Framework 简介：主要对 .NET Framework 基础及 C# 编程语言进行简单介绍。

第 2 章 Visual Studio 2010 简介：主要介绍了 Visual Studio 2010（简称 VS 2010）的基础操作及 Visual Studio 2010 的功能。

第 3 章 C# 语言特色：主要介绍了局部变量的隐式声明、自动属性、对象初始化器、扩展方法、匿名类型、局部方法和 Lambda 表达式等内容。

第 4 章 ASP.NET 技术简介：主要介绍了 ASP.NET 的特点、程序结构、配置、Web 窗体基础及应用程序事件等内容。

第 5 章 ASP.NET 服务器控件：主要介绍了 HTML 服务器控件、Web 服务器控件、验证控件、Calendar 日期控件、广告轮询控件及 MultiView 和 Wizard 控件等内容。

第 6 章 ASP.NET 应用程序状态管理：主要介绍了 ViewState 视图状态、跨页传递信息、使用 cookie 保存会话信息、会话状态和应用程序状态等内容。

第 7 章 异常处理和日志追踪：主要介绍了 ASP.NET 应用程序异常处理、异常的记录、错误页面及页面追踪等内容。

第 8 章 ASP.NET 中的样式、主题和母版页：主要介绍了在 ASP.NET 中如何应用样式、主题的创建与应用及母版页的创建与编辑等内容。

第 9 章 ASP.NET 中的站点导航技术：主要介绍了站点地图的定义与使用、TreeView 控件及 Menu 控件的使用。

第 10 章 ASP.NET AJAX 开发技术：主要介绍了 ASP.NET AJAX 基础、ASP.NET AJAX 服务器控件及 ASP.NET AJAX 服务器端控件扩展等内容。

第 11 章 ADO.NET 数据访问技术：主要介绍了 ADO.NET 基础、数据库的连接及如何与数据库交互等内容。

第 12 章 ASP.NET 数据绑定技术：主要介绍了如何对数据进行绑定及如何使用数据源控件。

第 13 章 数据绑定控件：主要介绍了 GridView、ListView 和 DataPager、DetailsView 和 FormView 等控件的使用。

第 14 章 成员和角色管理技术：主要介绍了如何使用 Membership API、如何使用登录

控件、使用 Membership 类及角色管理和授权等内容。

第 15 和 16 两章，通过家教预约系统和仓库出入库管理系统的开发，对前面的知识点进行了综合运用。

本书结构安排合理、信息量大，语言通俗易懂。内容涵盖了 ASP.NET 开发的许多方面，每个知识点都配有相应的示例，帮助读者充分地参与实践过程，并在实践中牢牢掌握这些知识点。另外，本书附赠的光盘包含了本书涉及的主要代码及主要案例的视频演示，可以帮助读者更好地掌握相关知识。

本书由刘志勇和黄定光两位老师共同编写，他们均有着丰富的项目开发经验，参加过数十个项目的策划与开发工作，在岗位中担任重要职位。参与本书编写工作的还有钱慎一、徐明华、王国胜、张敬伟、蒋燕燕、杨诚、张石磊、张丽、王梦迪、马陈、薛峰、贺金玲、任海峰、曹培培等老师，在此一一表示感谢。当然，尽管我们在编写过程中力求精益求精，仍难免有疏漏和不足之处，也恳请广大读者给予指正。

目 录

第 1 章 C#与.NET Framework 简介 ... 1

- 1.1 .NET Framework 基础 1
 - 1.1.1 什么是.NET Framework 1
 - 1.1.2 .NET 类库简介 2
 - 1.1.3 .NET Framework 4 的新功能 3
- 1.2 C#编程语言简介 5
 - 1.2.1 .NET 编程语言介绍 5
 - 1.2.2 C#程序设计语言 5
 - 1.2.3 Hello World 示例程序 6
- 1.3 本章小结 8

第 2 章 Visual Studio 2010 介绍 9

- 2.1 Visual Studio 2010 IDE 介绍 9
 - 2.1.1 解决方案资源管理器 9
 - 2.1.2 文档窗口 10
 - 2.1.3 错误列表和任务列表窗口 11
 - 2.1.4 服务器资源管理器 12
 - 2.1.5 对象浏览器 13
 - 2.1.6 代码编辑器 13
- 2.2 Visual Studio 2010 的功能特色 15
 - 2.2.1 多目标支持 15
 - 2.2.2 JavaScript 代码智能提示和调试功能 16
 - 2.2.3 内置的 ASP.NET AJAX 支持 17
 - 2.2.4 Web 站点与 Web 项目支持 18
 - 2.2.5 Web 设计器的改进 20
 - 2.2.6 嵌套的母版页支持 20
- 2.3 本章小结 21

第 3 章 C# 4.0 语言特色 22

- 3.1 局部变量的隐式声明 22
 - 3.1.1 如何声明隐式局部变量 22

- 3.1.2 在 foreach 循环中使用隐式声明 23
- 3.1.3 隐式类型变量的限制 23
- 3.1.4 隐式类型的局部数组 24
- 3.1.5 隐式类型变量的作用 24
- 3.2 自动属性 25
 - 3.2.1 理解自动属性 25
 - 3.2.2 访问自动属性 26
 - 3.2.3 自动属性的访问限制 26
 - 3.2.4 自动属性和默认值 27
- 3.3 对象初始化器 28
 - 3.3.1 理解对象初始化器语法 28
 - 3.3.2 调用使用初始化语法的自定义构造函数 29
 - 3.3.3 内部类型的初始化 29
 - 3.3.4 理解集合初始化语法 31
- 3.4 扩展方法 32
 - 3.4.1 定义扩展方法 32
 - 3.4.2 调用扩展方法 33
 - 3.4.3 扩展方法的使用范围 34
 - 3.4.4 使用扩展方法扩展接口类型 35
- 3.5 匿名类型 36
 - 3.5.1 理解匿名类型 36
 - 3.5.2 匿名类型的内部呈现 38
 - 3.5.3 匿名类型的相等语法 38
- 3.6 局部方法 39
 - 3.6.1 局部方法的规则 40
 - 3.6.2 使用局部方法 41
- 3.7 lambda 表达式 41
 - 3.7.1 理解 lambda 表达式 41
 - 3.7.2 剖析 lambda 表达式 42
 - 3.7.3 在 lambda 中使用多语句块 43
 - 3.7.4 多参数的 lambda 表达式 44

3.8 LINQ 简介	45	5.1.1 HTML 服务器控件简介	76
3.8.1 什么是 LINQ	45	5.1.2 HtmlContainerControl 和 HtmlInputControl 类	76
3.8.2 LINQ 示例	46	5.1.3 HTML 控件类	79
3.9 本章小结	47	5.1.4 编程创建 HTML 控件	80
3.10 上机练习	47	5.1.5 处理服务器端事件	81
第 4 章 ASP.NET 4.0 技术简介	48	5.2 Web 服务器控件	83
4.1 ASP.NET 介绍	48	5.2.1 基本 Web 服务器控件介绍	83
4.1.1 ASP.NET 4.0 的特点	48	5.2.2 常用控件的属性介绍	85
4.1.2 一个简单的 ASP.NET 应用 程序	49	5.2.3 列表控件介绍	86
4.2 ASP.NET 程序结构	51	5.2.4 可选择的列表控件	88
4.2.1 ASP.NET 文件类型	51	5.2.5 Bulletedlist 列表控件	90
4.2.2 ASP.NET 目录介绍	52	5.2.6 表格控件	92
4.3 ASP.NET 配置	54	5.3 验证控件	93
4.3.1 Web.config 配置文件	55	5.3.1 验证控件介绍	93
4.3.2 嵌套配置设置	56	5.3.2 验证控件的基类 BaseValidator	94
4.3.3 在 Web.config 中存储自定义 设置	57	5.3.3 使用 RequiredFieldValidator 强制用户输入	95
4.3.4 ASP.NET Web 站点管理 工具 WAT	58	5.3.4 使用 RangeValidator 控件 指定输入范围	96
4.3.5 编程读取和写入配置设置	61	5.3.5 使用 CompareValidator 比较值	97
4.3.6 加密配置块	64	5.3.6 RegularExpressionValidator 正则表达式验证	99
4.4 Web 窗体基础	66	5.3.7 VustomValidator 自定义验证	100
4.4.1 基于事件的编程模型	66	5.3.8 使用 ValidationSummary 显示验证信息	101
4.4.2 自动回发特性 (AutoPostBack)	67	5.3.9 使用 ValidationGroup 属性	102
4.4.3 Web 窗体处理流程	68	5.3.10 编程使用验证控件	105
4.4.4 ASP.NET 中的 page 类	70	5.4 Calendar 日期控件	106
4.4.5 页面重定向	71	5.4.1 日期控件简介	106
4.4.6 HTML 编码	72	5.4.2 日期控件格式化显示	107
4.5 应用程序事件	72	5.4.3 限制日期	108
4.5.1 Global.asax 全局文件	73	5.4.4 编程控制日期选择	108
4.5.2 基本应用程序事件	74	5.5 广告轮询控件 Adrotator	109
4.6 本章小结	74	5.5.1 广告文件设置	109
4.7 上机练习	75	5.5.2 Adrotator 类介绍	110
第 5 章 ASP.NET 服务器控件	76	5.6 MultiView 和 Wizard 控件	111
5.1 HTML 服务器控件	76		

5.6.1	MultiView 多视图控件	111	7.2.3	自定义日志	147
5.6.2	Wizard 向导控件	113	7.2.4	编程查看事件日志	149
5.6.3	WizardStep 向导步骤属性	114	7.3	错误页面	150
5.6.4	Wizard 事件	115	7.3.1	错误模式	151
5.7	本章小结	116	7.3.2	自定义错误页面	151
5.8	上机练习	117	7.4	页面追踪	152
第 6 章 ASP.NET 应用程序状态管理			7.4.1	启用页面追踪	152
6.1	ASP.NET 状态管理简介	118	7.4.2	页面追踪信息	152
6.2	ViewState 视图状态	119	7.4.3	写入页面追踪信息	153
6.2.1	视图状态原理	119	7.4.4	应用程序页面追踪信息	154
6.2.2	在视图状态中存储对象	120	7.5	本章小结	156
6.2.3	视图状态的安全	122	7.6	上机练习	156
6.3	跨页传递信息	123	第 8 章 ASP.NET 中的样式、主题和母版页		
6.3.1	查询字符串	123	8.1	在 ASP.NET 中应用 CSS 样式	157
6.3.2	跨页回送技术	125	8.1.1	创建样式	157
6.4	使用 Cookie 保存会话信息	126	8.1.2	应用样式	159
6.4.1	Cookie 技术简介	126	8.2	主题	160
6.4.2	使用 Cookie 保存信息	126	8.2.1	创建主题	161
6.5	会话状态	127	8.2.2	创建命名皮肤	162
6.5.1	会话状态简介	128	8.2.3	处理主题冲突	163
6.5.2	使用会话状态	129	8.2.4	为整个网站应用主题	164
6.5.3	配置会话状态	131	8.2.5	添加 CSS 样式	165
6.6	应用程序状态	134	8.2.6	动态应用主题	166
6.7	本章小结	136	8.3	母版页	167
6.8	上机练习	136	8.3.1	创建母版页	168
第 7 章 异常处理和日志追踪			8.3.2	默认内容	169
7.1	ASP.NET 应用程序异常处理	137	8.3.3	母版页和相对路径	170
7.1.1	异常处理基础	138	8.3.4	在 Web.config 中配置母版页	171
7.1.2	Exception 类	139	8.3.5	修改母版页	171
7.1.3	捕捉异常	140	8.3.6	动态加载母版页	173
7.1.4	异常的嵌套	141	8.3.7	母版页的嵌套	174
7.1.5	自定义异常	142	8.4	本章小结	176
7.1.6	finally 语句块	143	8.5	上机练习	176
7.2	记录异常	145	第 9 章 ASP.NET 中的站点导航技术		
7.2.1	查看 Windows 事件日志	145	9.1	站点地图	177
7.2.2	写入事件日志	146	9.1.1	定义站点地图文件	177

9.1.2	使用 SiteMapPath 控件	178	10.3.1	安装 ASP.NET AJAX Control Toolkit	214
9.1.3	SiteMap 类	180	10.3.2	使用 ASP.NET AJAX 扩展控件	218
9.1.4	URL 地址映射	180	10.4	本章小结	227
9.2	TreeView 控件	181	10.5	上机练习	228
9.2.1	为 TreeView 控件添加或 删除节点	182	第 11 章 ADO.NET 数据访问技术		229
9.2.2	设计 TreeView 控件的外观	183	11.1	ADO.NET 基础	229
9.2.3	使用 TreeView 控件用于 站点导航	185	11.1.1	ADO.NET 结构	229
9.2.4	绑定到 XML 文件	187	11.1.2	ADO.NET 数据提供者	230
9.3	Menu 控件	189	11.2	连接数据库	231
9.3.1	添加 Menu 控件	189	11.2.1	SqlConnection 数据库 连接类	231
9.3.2	为 Menu 控件定义项	190	11.2.2	使用 SqlConnectionStringBuilder 连接字符串	232
9.3.3	设计动态或静态菜单	192	11.2.3	关闭和释放连接	232
9.3.4	Menu 控件样式	194	11.2.4	在 Web.config 配置文件中 保存连接字符串	233
9.3.5	绑定到站点地图的菜单	196	11.2.5	连接池技术	234
9.4	本章小结	198	11.3	与数据库交互	235
9.5	上机练习	198	11.3.1	使用 Connection 打开与 关闭数据库	235
第 10 章 ASP.NET AJAX 开发技术		199	11.3.2	使用 Command 操作数据库	236
10.1	ASP.NET AJAX 基础	199	11.3.3	使用 DataReader 对象检索 数据	238
10.1.1	AJAX 技术简介	199	11.3.4	使用 DataAdapter 和 DataSet 对象	239
10.1.2	ASP.NET 中的 AJAX 结构	200	11.4	综合案例: 用户管理	241
10.2	ASP.NET AJAX 服务器控件	201	11.5	本章小结	249
10.2.1	声明 ScriptManager 控件	202	11.6	上机练习	250
10.2.2	使用 ScriptManager 注册 脚本	202	第 12 章 ASP.NET 数据绑定技术		251
10.2.3	在 ScriptManager 中注册 Web 服务	203	12.1	数据绑定概要	251
10.2.4	使用 UpdatePanel 控件	204	12.1.1	使用 <%#表达式%> 绑定 数据	251
10.2.5	使用嵌套的 UpdatePanel 控件	206	12.1.2	使用 DataSource 属性绑定 数据	253
10.2.6	编程控制 UpdatePanel 控件	208	12.1.3	使用 Eval 方法绑定数据源	254
10.2.7	使用 UpdateProgress 控件	210			
10.2.8	使用 Timer 控件更新 UpdatePanel	211			
10.3	ASP.NET AJAX 服务器端控件 扩展	214			

12.2	数据源控件	256
12.2.1	SqlDataSource 控件	256
12.2.2	使用 SqlDataSource 控件连接到 SQL Server 数据库	257
12.2.3	ObjectDataSource 控件	264
12.2.4	在 ObjectDataSource 中使用参数类查询数据	265
12.2.5	LinqDataSource 数据源简介	266
12.2.6	使用 LinqDataSource 更新数据	267
12.3	本章小结	272
12.4	上机练习	272

第 13 章 数据绑定控件

13.1	GridView 控件	273
13.1.1	使用 GridView 控件	274
13.1.2	GridView 布局和样式	276
13.1.3	使用模板列	277
13.1.4	GridView 控件事件	278
13.1.5	实现分页	279
13.1.6	实现排序效果	282
13.2	ListView 和 DataPager 控件	285
13.2.1	ListView 控件简介	286
13.2.2	对 ListView 项应用样式	286
13.2.3	使用 DataPager 分页	286
13.2.4	使用 ListView 控件对数据进行分页和排序	287
13.3	DetailsView 和 FormView 控件	297
13.3.1	DetailsView 控件简介	297
13.3.2	使用 DetailsView 绑定数据源	297
13.3.3	DetailsView 与 GridView 的联合使用	300
13.3.4	FormView 控件简介使用	302
13.3.5	利用模板显示数据	303
13.3.6	利用 FormView 编辑数据	304
13.4	本章小结	305
13.5	上机练习	305

第 14 章 成员和角色管理技术

14.1	ASP.NET Membership API 简介	306
14.2	使用 Membership API	308
14.3	使用登录控件	313
14.3.1	Login 控件	313
14.3.2	LoginStatus 控件	315
14.3.3	LoginView 控件	316
14.3.4	PasswordRecovery 控件	316
14.3.5	ChangePassword 控件	318
14.3.6	CreateUserWizard 控件	319
14.4	使用 Membership 类	320
14.4.1	创建用户	321
14.4.2	编辑和删除用户	321
14.5	角色管理和授权	322
14.5.1	URL 授权	323
14.5.2	文件和文件夹访问授权	324
14.5.3	角色管理器	324
14.5.4	使用 Roles API 配置基于角色的授权	325
14.6	本章小结	326
14.7	上机练习	326

第 15 章 家教预约系统

15.1	系统概述	327
15.2	需求分析	327
15.3	用例图	327
15.4	系统总体设计	328
15.5	开发环境设计	328
15.6	数据库设计	329
15.6.1	数据库结构设计	329
15.6.2	数据表的创建	329
15.7	项目及数据库搭建	330
15.8	数据库连接字符串	333
15.9	学员主要模块的实现	333
15.9.1	系统登录	333
15.9.2	用户注册	339
15.9.3	预约教员	347
15.9.4	修改信息	352



15.10	管理员的主要模块	360
15.10.1	教师审批	360
15.10.2	预约管理	368
15.11	教员的主要模块	375
15.12	本章小结	380
15.13	上机练习	380

第 16 章 仓库出入库管理系统 381

16.1	系统概述	381
16.2	需求分析	381
16.3	用例图	381
16.4	系统总体设计	382
16.5	开发环境	382
16.6	数据库设计	383
16.6.1	数据库的结构设计	383
16.6.2	数据表的创建	383
16.7	项目及数据库搭建	384

16.8	数据访问层实现	387
16.8.1	公共数据库访问类 SqlHelper 的实现	387
16.8.2	Web.config 文件配置	393
16.9	登录界面的实现	394
16.10	系统的主要模块	398
16.10.1	主页	398
16.10.2	入库登记页面	403
16.10.3	库存查询页面	408
16.10.4	库存盘点页面	413
16.10.5	出库登记页面	419
16.10.6	出入库报表页面	428
16.10.7	数据库备份页面	437
16.10.8	修改密码页面	440
16.11	本章小结	445
16.12	上机练习	445

第1章 C#与.NET Framework 简介

Framework 平台是微软公司继 ASP 之后推出的高度集成化的平台。它将服务器页面的控件进行了分类整合,将经常应用到的功能集成为公共的控件对其进行深入的开发,并且将所有的控件、服务等整合在一起,构成了一个综合的开发平台。使程序开发人员能够快速准确地为客户提供一个优质高效的整体解决方案。

本章主要内容:

- ☐ .NET Framework 基础
- ☐ C#编程语言简介
- ☐ Hello Word 示例

1.1 .NET Framework 基础

微软公司的.NET 框架经过了十多年的发展已经变得十分强大和成熟,在 2002 年微软推出了第一个版本 Visual Studio 2002,这个版本只是一个开发工具的模式。一年后推出了 Visual Studio 2003,该版本的开发工具已经具备了后期开发工具的雏形,但是没有后期编译器的使用方便,比如当用户将光标放在当前变量的时候,所有当前的这个变量全部加重颜色显示。这个在以前的编译器中是没有的,只有在 Visual Studio 2010 以后的编译器才存在,这样有利于识别当前变量和快速查找当前变量,在后续的编译器中这样的例子很多。

1.1.1 什么是.NET Framework

.NET Framework 的核心包括三部分,CLS 公共语言规范(Common Language Specification),CLR 公共语言运行库(Common Language Runtime)和 CTS 通用类型系统(Common Type System)。

1. CLS 公共语言规范(Common Language Specification)

CLS 是微软在开发 Framework 平台的同时提出的一个标准,遵循这个标准开发的类库能够被所有的语言所调用。在微软的 Framework 平台中几乎所有的类都是遵循这个标准开发的,所以这些类库都能够被所有开发语言的程序所调用。

所有类型都可以在 CTS 中声明。CTS 定义了一组语言编译器必须遵循的规则,以定义、引用、使用和存储引用类型和值类型。因此,遵循 CTS 在不同语言中编写的对象才能彼此交互,但并不是所有的类型都可以用于所有的语言。要建立可以在所有.NET 语言中访问的组件就要使用 CLS。有了 CLS,编译器就可以根据 CLS 规范检查代码是否有效。

任何支持.NET 的语言都不仅仅局限于 CLS 定义的公共功能的子集,利用.NET 所支持的语言比利用其他语言要简单得多。如果把自己限制在 CLS 之内,就可以保证组件能够在

所有的语言中使用。第三方编写的库很可能限制在 CLS 之内,以确保该库可以在所有的语言中使用。

.NET Framework 是为了支持多种语言而设计的。在设计 .NET 的阶段中,Microsoft 让许多编译器开发商建立它们自己的 .NET 语言。Microsoft 自己就发布了 VB.NET、Managed C++、C#、J# 和 JScript。另外,不同开发商开发了 40 多种语言,例如 COBOL、Smalltalk、Perl 和 Eiffel 等。每种语言都有其特有的优点和许多不同的功能,这些语言的编译器都进行了扩展以支持 .NET。

2. 公共语言运行库 (Common Language Runtime)

为了实现快速开发,微软在 Framework 中提供了公共语言运行库,负责把将要使用到的资源加载到系统内存中。除了系统资源外,还预先加载用于算数和逻辑运算、控制系统 IO、内存访问、异常处理、垃圾回收等可能会使用到的资源。CLR 具备跨语言应用、安全性高、版本化清晰等特点。

3. CTS 通用类型系统 (Common Type System)。

CTS 通用类型系统也是微软在 Framework 开发的同时提出的一个标准。它可以使不同语言编写的代码直接交互使用。CTS 可以分为值类型和应用类型两大类,同时两种类型之间可以进行强制转换,这种转换称为 Boxing (装箱) 和 UnBoxing (拆箱)。CTS 的每一种类型都是对象,并继承一个基类 System.Object。所有遵循 CTS 标准开发的类型都能够在所有开发语言中进行类型间的互相转换,这也是高级语言的一个明显的特征。毕竟一个高速、快捷、安全的类型转换是非常必要的。

1.1.2 .NET 类库简介

在 .NET 中有很多的基础类库用以进行快速开发。比如 System 命名空间,此命名空间是开发的所有命名空间是基础。当然,如果要引用其他的类库也能达到这个空间的效果,但是要实现快速开发的原则就不太可靠了。System 命名空间是基础,其中包括常用的值和对象引用类型、事件、接口、属性和异常处理等操作。下面简单介绍下几个经常使用到的类库。

Dispose 方法: 顶层接口中的静态方法。所以对象都拥有此方法。手动调用可以进行垃圾回收机制,释放由 ActivationContext 使用的资源。

Array 类: 提供数组的操作方法,所有数组操作的基类。

Console 类: 控制台应用程序的标准输入流、输出流和错误流。

Convert 类: 为基础数据类型转换提供操作。

Delegate 类: 实现委托功能。

Exception 类: 所有异常类的基类。

EventArgs 类: 所有事件类的基类。

Math 类: 所有计算函数类。

Random 类: 所有随机数的基类。

I/O 类: 所有 I/O 操作的基类。

此外,还有很多基类,在后面的学习中会接触到。

1.1.3 .NET Framework 4 的新功能

1. 应用程序兼容性和部署

.NET Framework 4 与早期版本生成的应用程序有很高的兼容性,除了提高安全性、标准遵从性、正确性、可靠性等性能所做的一些更改外。

.NET Framework 4 不能自动识别和运行公共语言运行时版本,如果要运行.NET Framework 早期版本生成的应用程序,则必须使用 Visual Studio 项目属性指定.NET 版本,或者使用应用程序配置文件中的<supportedruntime>元素指定所支持的运行版本。

2. 核心新增功能和改进

核心新增功能和改进包括如下几种。

- (1) 诊断和性能。
- (2) 全球化。
- (3) 垃圾回收。
- (4) 代码协定。
- (5) 仅用于设计时的互操作程序集。
- (6) 动态语言运行时。
- (7) 协变和逆变。
- (8) BigInteger 和复数。
- (9) 元组。
- (10) 文件系统枚举改进。
- (11) 内存映射文件。
- (12) 64 位操作系统和进程。

3. Managed Extensibility Framework

Managed Extensibility Framework (MEF) 是.NET Framework 4 中的一个新库,可帮助用户生成可扩展和可组合的应用程序。使用 MEF 可指定可以扩展应用程序的位置,公开要提供给其他可扩展应用程序的服务并创建供可扩展应用程序使用的部件。MEF 还可以基于元数据启用可用部件的便捷发现功能,而无须加载部件的程序集。

4. 并行计算

.NET Framework 4 引入了用于编写多线程和异步代码的新编程模型,极大地简化了应用程序和库开发人员的工作,该模型使开发人员可以通过固有方法编写高效、细化且可伸缩的并行代码,不必直接处理线程或线程池。新的 System.Threading.Tasks 命名空间和其他相关类型支持此新模型。并行 LINQ (PLINQ) 是 LINQ to Objects 的并行实现,能够通过声明性语法实现类似功能。

5. 网络

网络改进包含以下几个方面:

- 若干类中对于 Windows 身份验证的安全改进,包括 System.Net.HttpWebRequest、System.Net.HttpListener、System.Net.Mail.SmtpClient、System.Net.Security.SslStream 和 System.Net.Security.NegotiateStream。对于 Windows 7 和 Windows Server 2008 R2

上的应用程序提供了扩展保护。

- ❑ 支持使用 IPv6 或 Teredo 的网络地址转换 (NAT) 遍历。
- ❑ 提供有关 `HttpWebRequest` 对象信息新的网络性能计数器。
- ❑ 在 `System.Net.HttpWebRequest` 类中, 支持在 `AddRange` 方法的新重载中使用大字节范围标头 (64 位范围)。 `System.Net.HttpWebRequest` 类的新属性允许应用程序设置很多 HTTP 标头。可使用 `Host` 属性设置 HTTP 请求中独立于请求 URI 的 `Host` 标头值。
- ❑ 对于 `System.Net.Mail.SmtpClient` 和相关类的安全套接字层 (SSL) 支持。
- ❑ 改进了对 `System.Net.Mail.MailMessage` 类中的邮件标头的支持。
- ❑ 对在加密中使用 `null` 密码的支持。可以通过使用 `System.Net.ServicePointManager` 类和 `EncryptionPolicy` 属性指定加密策略。 `System.Net.Security.SslStream` 类的构造函数现在采用 `System.Net.Security.EncryptionPolicy` 类作为参数。
- ❑ `System.Net.NetworkCredential` 类中提供了用于基于密码的身份验证方案 (例如基本、摘要、NTLM 和 Kerberos 身份验证) 的凭据。为了提高安全性, 现在可将密码作为 `System.Security.SecureString` 实例而不是 `System.String` 实例进行处理。
- ❑ 在 `System.Uri` 和 `System.Net.HttpListener` 类中指定如何转换和规范化使用百分号编码值的 URI 能力。

6. Web

ASP.NET 4 在以下几个方面引入了新功能:

- ❑ 核心服务: 包括可用来扩展缓存的新 API、支持对会话状态的数据进行压缩以及新的应用程序预加载管理器 (自动启动功能)。
- ❑ Web 窗体: 包括对 ASP.NET 路由的更集中化支持、对 Web 标准的增强支持、更新的浏览器支持、数据控件的新功能以及视图状态管理的新功能。
- ❑ Web 窗体控件: 包括新的 `Chart` 控件。
- ❑ MVC: 包括视图的新帮助器方法、对分区的 MVC 应用程序的支持以及异步控制器。
- ❑ 动态数据: 包括对现有 Web 应用程序的支持、对多对多关系和继承的支持、新的字段模板和特性及增强的数据筛选。
- ❑ Microsoft AJAX: 包括对 Microsoft AJAX 库中的基于客户端的 AJAX 应用程序的附加支持。
- ❑ Visual Web Developer, 包括改进的 JScript IntelliSense、针对 HTML 和 ASP.NET 标记的新的自动完成代码段和增强的 CSS 兼容性。
- ❑ 部署: 包括用于自动化典型部署任务的新工具。
- ❑ 多目标: 包括针对目标版本的 .NET Framework 中未提供的功能的更好的筛选功能。

7. 客户端

Windows Presentation Foundation (WPF) 版本 4 包含以下方面的更改和改进。

- ❑ 新控件: 包括 `Calendar`、`DataGrid` 和 `DatePicker`。
- ❑ `VisualStateManager` 支持更改控件的状态。
- ❑ 利用触控和操作, 用户可以创建在 Windows 7 上同时接收来自多个触控的输入的应用程序。

- ❑ 图形和动画支持布局舍入、像素着色器版本 3.0、缓存合成和缓动函数。
- ❑ 文本改进了文本呈现，并支持在文本框中自定义插入符号的颜色和选定内容的颜色。
- ❑ InputBinding 的 Command 属性、动态对象和 Text 属性支持绑定。
- ❑ XAML 浏览器应用程序 (XBAP) 支持与网页通信，并且支持完全信任部署。
- ❑ 利用 System.Windows.Shell 命名空间中新增的类型，用户能够与 Windows 7 任务栏通信，还能将数据传递到 Windows shell。
- ❑ Visual Studio 2010 中的 WPF 和 Silverlight 设计器中提供了各种改进，有助于创建 WPF 或 Silverlight 应用程序。

1.2 C#编程语言简介

C#是专门为.NET 的应用而开发的语言。它吸收了 C++、Visual Basic、Delphi、Java 等语言的优点，体现了当今最新的程序设计技术的功能和精华。C#继承了 C 和 Java 语言的语法风格，同时又继承了 C++的面向对象特性。不同的是：C#的对象模型已经面向 Internet 进行了重新设计，使用的是.NET 框架的类库；C#不再提供对指针类型的支持，使得程序不能随便访问内存地址空间，从而更加健壮；C#不再支持多重继承，避免了以往类层次结构中由于多重继承带来的可怕后果。.NET 框架为 C#提供了一个强大的、易用的、逻辑结构一致的程序设计环境。同时，公共语言运行时 (Common Language Runtime) 为 C#程序语言提供了一个托管的运行时环境，使程序比以往更加稳定、安全。其特点有：

- ❑ 语言简洁。
- ❑ 保留了 C++的强大功能。
- ❑ 快速应用开发功能。
- ❑ 语言的自由性。
- ❑ 强大的 Web 服务器控件。
- ❑ 支持跨平台。
- ❑ 与 XML 相融合。

1.2.1 .NET 编程语言介绍

.NET 编程语言是一种面向对象的开发语言。由于是运行在公共语言运行库下的，它可以在多种语言中互相调用，可以根据程序开发人员的个人习惯来选择自己熟悉的语言来开发，然后可以被其他语言所调用。这些优越性都取决于它的语言特点，是一种高效的安全的语言。

1.2.2 C#程序设计语言

C#语言是微软针对一系列问题的综合解决方案，是一种全新的、面向对象的编程语言。

它使得程序开发人员可以快速地编写各种基于 Microsoft .NET 平台的应用程序，Microsoft .NET 提供了一系列的工具和服务来最大程度地开发利用计算与通讯领域。C#语言具有以下几点特性：

1. 效率与安全性

新兴的网络经济迫使企业必须更加迅速地应对竞争的威胁。开发者必须不断缩短开发周期，推出应用程序的新版本，而不仅仅是开发一个“标志性”的版本。

C#在设计时就考虑了这些问题。它使开发者用更少的代码做更多的事，同时也不易出错。

2. 支持现有的网络编程新标准

新的应用程序开发模型意味着越来越多的解决方案依赖于新出现的网络标准，例如 HTML、XML、SOAP 等。现存的开发工具往往都是早于 Internet 出现的，或者是在我们所熟知的网络还处于孕育期时出现的。所以，它们一般无法很好地支持最新的网络技术。

3. 消除大量程序错误

即使是专家级的 C++程序员也常会犯一些最简单的小错误，比如忘了初始化变量。但往往就是这些小错误造成难以预料的后果，有些甚至需要很长时间来寻找和解决。一旦一个程序作为产品来使用，即使最简单的错误纠正起来也可能要付出极其昂贵的代价。

4. 扩展交互性

作为一种自动管理的类型安全的环境，C#适合于大多数企业应用程序，但经验表明有些应用程序仍然需要一些底层的代码，要么是因为基于性能的考虑，要么是因为要与现有的应用程序接口兼容，这些情况可能会迫使开发者使用 C++，即使他们本身使用更高效的开发环境。

1.2.3 Hello World 示例程序

学习编程语言，通常采用“Hello, World”程序作为起步。下面我们就来演示 C#版的 Hello World 实现过程。

首先打开 Visual Studio 2010 开发工具，从开始菜单中打开 Visual Studio 2010 集成开发环境，在主菜单中点击“文件|新建项目”菜单，在弹出的对话框中左侧选择 Visual C#，在右侧的列表中选择“控制台应用程序”，在项目名称中输入 Hello World，点击“确定”按钮。完成后，编辑器将自动进入视图窗口。整体代码如下。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```