



普通高等教育“十一五”国家级规划教材

高等学校计算机系列

C#应用程序设计 教程

耿肇英 耿 炎 编著

普通高等教育“十一五”国家级规划教材

高等学校计算机系列

C#应用程序设计教程

耿肇英 耿 焱 编著

 人民邮电出版社
POSTS & TELECOM PRESS

北 京

图书在版编目 (CIP) 数据

C#应用程序设计教程/耿肇英, 耿焱编著. —北京: 人民邮电出版社, 2007.9
普通高等教育“十一五”国家级规划教材. 高等学校计算机系列
ISBN 978-7-115-16335-6

I. C... II. ①耿...②耿... III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 081150 号

内 容 提 要

本书介绍 C#应用程序设计技术, 内容包括: C#语言基础、Windows 和 ASP.NET 应用程序开发、2D 和 3D 图形图像编程、文件读写和管理、多线程应用、ADO.NET 数据库应用程序设计、XML 标记语言、Web 服务等。本书使用微软公司免费的速成版集成开发环境, 它们包括了正式版的大部分功能, 非常适合教学。本书所有例子在 .NET Framework 2.0 下调试通过。

本书可作为大学本专科计算机导论和 C 语言的后续课程“面向对象 Windows 程序设计”的教材, 或作为学习使用 C#语言开发 Windows 和 ASP.NET 应用程序培训班教材, 也适合使用 C#语言开发项目的程序员参考。

普通高等教育“十一五”国家级规划教材

高等学校计算机系列

C#应用程序设计教程

-
- ◆ 编 著 耿肇英 耿 焱
责任编辑 张孟玮
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京华正印刷有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 21
字数: 501 千字 2007 年 9 月第 1 版
印数: 1—3 000 册 2007 年 9 月北京第 1 次印刷

ISBN 978-7-115-16335-6/TP

定价: 32.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

编 者 的 话

计算机专业开设的第一门高级程序设计语言课程一般是 C 或 C++, 学习高级程序设计语言基本知识, 编写控制台或 DOS 应用程序。在此之后应进一步学习面向对象 Windows 应用程序设计, 一方面为操作系统、数据库、计算机网络、计算机图形学等后续课程提供程序设计基础, 另一方面也尽早为学生课外实践提供一个实用工具。本课程是为上述目的而设置的, 是计算机导论和 C 或 C++ 课程的后续课程。单纯讲授面向对象理论的学习效果并不好, 结合某个面向对象语言的方法会有更好的学习效果。本书用 C# 语言介绍面向对象程序设计概念。

学习面向对象 Windows 应用程序设计应采用广泛使用又适合教学的工具和语言。考虑到 VC++ 学习难度较大, VB 虽然易学, 还要耗费很紧张的学时学习一门新语言, 是得不偿失的。而 C# 语言是为 .NET 架框量身定做的新一代面向对象的语言, 语法和 C++ 非常相似, 但比 C++ 简单, 使用组件编程, 和 VB 一样容易使用。在第一门高级程序设计语言课程是 C 或 C++ 情况下, 学习 C# 语言可节约很多学时, 对于根本没有接触过 VB 的大学生, 把 C# 作为学习面向对象 Windows 应用程序设计语言是合适的。当前用 C# 语言开发 .NET 系统的商用软件较多, 这也为毕业后工作打下较好基础。

以往 Windows 和 ASP.NET 应用程序的设计方法有很大不同, 一般作为两门课程开设。在 .NET 架构中, 两者的设计方法十分类似, 都使用组件的属性、事件和方法等面向对象的概念, 两种设计方法联系紧密, 例如文件系统、ADO.NET 是两种方法都要使用的。一个复杂的系统往往要包含 C/S 和 B/S 结构。在这种情况下, 把 Windows 和 ASP.NET 应用程序的设计作为一门课程开设是合适的。本书前半部分讲述 Windows 应用程序设计方法, 后半部分讲述 ASP.NET 和 Web 服务的设计方法。

本书采用实例教学法, 在讲清基本知识点的基礎上, 尽量使用实例加以说明, 因此书中包含了大量实用例子, 在例子中尽量避免不相关的知识点和无关的代码, 使例子代码短小精悍, 容易理解。书中绝大部分例子都给出了详细的设计步骤,

并对每一步所实现的代码给出详细的解释，读者可按照书中步骤完成书中例子，因此本书也可看作是学生的实验指导手册。本书所有例子代码及教学课件都可以从人民邮电出版社网站（www.ptpress.com.cn/download/）下载。

本课程的教学参考学时为讲课 54 学时，上机实验 36 学时，本书中的大部分习题也需要上机完成，因此如有可能也可适当增加上机实验学时。如学时紧张，讲课也可缩短为 36 学时，此时要将一些完整的例子，例如计算器、文本编辑器、留言板等，安排为上机实验，由于这些例子都有详细的步骤，学生应能完成这些例子。

本书编著工作主要由耿肇英完成，耿焱完成了有关多媒体以及网页设计的相应章节。杜建文、李燕杰、朱晓莲、周真真、姜志敏、韩忠政、冯东等也参加了本书部分代码的编写、调试、整理以及录入校对等工作，在此一并表示感谢。

由于时间仓促，加之水平有限，书中的缺点和不足之处在所难免，敬请读者批评指正。联系方法：gengzhaoying@sina.com。

编者

2007 年 4 月

目 录

第 1 章 C#语言基础	1
1.1 C#语言特点	1
1.2 编写控制台应用程序	2
1.3 类的基本概念	4
1.3.1 类的基本概念	4
1.3.2 类成员的存取控制	4
1.3.3 类的对象	5
1.3.4 类的构造函数、构造函数重载和析构函数	5
1.3.5 使用 Person 类的完整的例子	6
1.4 C#的数据类型	7
1.4.1 值类型和引用类型的区别	7
1.4.2 值类型变量分类	8
1.4.3 结构类型	8
1.4.4 简单类型	8
1.4.5 枚举类型	9
1.4.6 值类型的初值和默认构造函数	10
1.4.7 可空类型	10
1.4.8 引用类型分类	10
1.4.9 object 类	11
1.4.10 数组类	11
1.4.11 字符串类 (string 类)	12
1.4.12 类型转换	13
1.4.13 泛型	15
1.5 运算符	16
1.5.1 运算符分类	16
1.5.2 溢出检查操作符 checked 和 unchecked	16
1.5.3 new 操作符	17
1.6 程序控制语句	17

1.6.1 C#和C语言语句的不同点	17
1.6.2 foreach 语句	18
1.6.3 异常语句	19
1.7 类的继承	20
1.7.1 派生类的声明格式	20
1.7.2 隐藏基类方法	21
1.7.3 base 关键字	21
1.7.4 C#语言类继承特点	22
1.8 类的成员	22
1.8.1 类的成员类型	22
1.8.2 类成员访问修饰符	22
1.9 类的字段和属性	23
1.9.1 静态字段、实例字段、常量和只读字段	23
1.9.2 属性	24
1.10 类的方法	25
1.10.1 方法的声明	25
1.10.2 方法中参数的传递	25
1.10.3 静态方法和实例方法	28
1.10.4 方法的重载	29
1.10.5 操作符重载	29
1.10.6 this 关键字	30
1.11 类的多态性	31
1.12 抽象类和抽象方法	33
1.13 密封类和密封方法	34
1.14 静态类和静态类成员	34
1.15 C#2.0 中的分部类	35
1.16 使自定义类支持 foreach 语句	35
1.17 接口	35
1.17.1 接口声明	36
1.17.2 接口的继承	36
1.17.3 类对接口的实现	36
1.18 代表类型	38
1.19 事件	39
1.19.1 事件驱动	39
1.19.2 事件的声明	39
1.19.3 事件的预订和撤销	40
1.20 索引指示器	41
1.21 命名空间	41
1.21.1 命名空间的声明	42

1.21.2 命名空间使用	42
习题一	43
第2章 Windows 编程的基础知识	44
2.1 Windows 编程接口和类库	44
2.2 Windows 应用程序的基本结构	45
2.2.1 最简单的 Windows 应用程序	45
2.2.2 用 VS2005 创建 Windows 应用程序	47
2.2.3 方案 (Solution) 和项目 (Project)	48
习题二	49
第3章 常用控件和类的使用	50
3.1 控件通用属性	50
3.2 Form 类	51
3.3 标签(Label)控件	51
3.4 按钮(Button)控件	52
3.5 事件处理函数的参数	53
3.6 文本框(TextBox)控件	53
3.7 Convert 类	55
3.8 RadioButton 和 GroupBox 控件	55
3.9 Font 类	56
3.10 复选框(CheckBox)控件	56
3.11 列表框(ListBox)控件	58
3.12 下拉列表组合框(ComboBox)控件	59
3.13 ToolTip 控件	60
3.14 超级链接(LinkLabel)控件	60
3.15 定时(Timer)组件	61
3.16 DateTime 类	62
3.17 菜单	62
3.18 工具条	64
3.19 状态栏控件	65
3.20 鼠标事件和 Panel 控件	66
3.21 快捷菜单	67
3.22 综合例子: 计算器	68
3.23 类库、组件和控件	71
习题三	76
第4章 文本编辑器的实现	78

4.1	RichTextBox 控件	78
4.2	剪贴板功能	79
4.3	存取文件	80
4.3.1	OpenFileDialog 和 SaveFileDialog 控件	80
4.3.2	存取文件功能的实现	81
4.4	修改字体	82
4.4.1	FontDialog 控件	83
4.4.2	修改字体实现方法	83
4.5	About 对话框	83
4.6	文本编辑器查找替换功能	84
4.6.1	模式对话框和非模式对话框	85
4.6.2	查找替换功能的实现	85
4.7	提示用户保存已被修改的文件	87
4.7.1	MessageBox 类	87
4.7.2	提示保存已被修改的文件	88
4.8	打印和打印预览	90
4.8.1	PrintDocument 类	90
4.8.2	页面设置对话框 (PageSetupDialog)	92
4.8.3	打印预览	93
4.8.4	用打印对话框 (PrintDialog) 实现打印	93
4.9	编写多文档界面应用程序	94
4.9.1	多文档文本编辑器	94
4.9.2	多选项卡页的文本编辑器	96
	习题四	98
第 5 章	图形图像编程	100
5.1	GDI+ (图形设备环境接口)	100
5.2	Graphics 类	100
5.2.1	使用 Graphics 类绘图的基本步骤	101
5.2.2	窗体的 Paint 事件	101
5.3	GDI+中常用的结构	102
5.4	画笔 Pen 类	103
5.5	创建画刷	104
5.5.1	单色画刷 (SolidBrush)	104
5.5.2	阴影画刷 (HatchBrush)	104
5.5.3	纹理 (图像) 画刷 (TextureBrush)	105
5.5.4	颜色渐变画刷 (LinearGradientBrush)	106
5.5.5	画刷 PathGradientBrush 和 GraphicsPath 类	107

5.6 基本图形的绘制和填充	108
5.6.1 用 DrawLine 方法绘制线段	108
5.6.2 泛型 List 类	111
5.6.3 DrawEllipse 方法画椭圆 (圆) 及键盘事件	111
5.6.4 DrawRectangle 方法画矩形	113
5.6.5 DrawArc 方法绘制圆弧	113
5.6.6 DrawPie 方法绘制饼图	113
5.6.7 DrawPolygon 方法绘制多边形	114
5.6.8 DrawPath 方法	114
5.6.9 DrawString 方法写字符串	114
5.6.10 DrawImage 和 DrawIcon 方法	115
5.6.11 FillEllipse 方法填充内切椭圆 (圆)	116
5.6.12 FillRectangle 方法填充矩形	116
5.6.13 FillPie 方法填充饼图	116
5.6.14 FillRegion 方法和 Region 类	117
5.7 Matrix 类和图形的平移、变形和旋转	118
5.7.1 Matrix 类	118
5.7.2 图形的平移、变形和旋转	119
5.7.3 仿射矩阵	120
5.8 图形框 PictureBox 控件	121
5.9 Bitmap 类	122
5.9.1 Bitmap 类支持的图像类型	122
5.9.2 Bitmap 类的方法	122
5.9.3 SetPixel 方法画点	122
5.9.4 在 PictureBox 中拖动鼠标画曲线	123
5.9.5 存取位图文件	124
5.9.6 用拖动鼠标方法画椭圆或圆	125
5.10 图像剪贴板功能	126
5.10.1 剪切、复制区域的选定	126
5.10.2 剪贴板复制功能的实现	127
5.10.3 剪贴板剪切功能的实现	127
5.10.4 剪贴板粘贴功能的实现	128
5.11 图像的处理	129
5.11.1 降低图像的分辨率	129
5.11.2 彩色图像变换为灰度图像	130
5.11.3 旋转和翻转图像	131
5.11.4 调整图像的颜色、亮度和对比度	132
5.11.5 音频和视频的播放	133
5.12 DirectX 程序设计入门	134

5.12.1	DirectX SDK 9.0C 安装及类库介绍	134
5.12.2	实现视频播放	135
5.12.3	3D 图形的数学基础	137
5.12.4	图形卡和 GPU	137
5.12.5	Device 类	138
5.12.6	Mesh 类	140
5.12.7	灯光和材质	142
5.12.8	变换和 Matrix 结构	142
5.12.9	改变观察点和旋转茶壶	145
5.12.10	增加多个茶壶	146
5.12.11	纹理	146
5.12.12	用 Mesh 显示 X 文件中的 3D 图形	148
习题五		150
第 6 章	文件和流	152
6.1	用流读写文件	152
6.1.1	FileStream 类读写字节	152
6.1.2	BinaryReader、BinaryWriter 类读写基本数据类型	154
6.1.3	StreamReader 和 StreamWriter 类读写字符串	155
6.1.4	序列化	156
6.1.5	Stream 类的其他派生类	158
6.2	File 类和 FileInfo 类	159
6.2.1	File 类常用的方法	159
6.2.2	判断文件是否存在	160
6.2.3	删除文件	160
6.2.4	复制文件	160
6.2.5	移动文件	161
6.2.6	设置文件属性	162
6.2.7	得到文件的属性	162
6.3	Directory 类和 DirectoryInfo 类	163
6.3.1	Directory 类常用的方法	163
6.3.2	判断目录是否存在	163
6.3.3	创建目录	164
6.3.4	删除目录	164
6.3.5	移动目录	165
6.3.6	获取当前目录下所有子目录	165
6.3.7	获取当前目录下的所有文件	166
6.3.8	设置目录属性	167

6.4 例子: 查找文件	167
6.4.1 ListView 和 FolderBrowserDialog 控件	167
6.4.2 在指定文件夹中查找文件	168
6.5 例子: 鼠标拖放打开文件	169
6.6 例子: 拆分和合并文件	171
习题六	172
 第7章 多线程程序设计	 173
7.1 创建线程	173
7.1.1 线程 (Thread) 类的属性和方法	173
7.1.2 创建线程	174
7.1.3 进度条 (ProgressBar) 控件	175
7.1.4 用线程控制进度条	175
7.1.5 BackgroundWorker 组件	177
7.2 多个线程互斥	179
7.2.1 多个线程同时修改共享数据可能发生错误	179
7.2.2 用 Lock 语句实现互斥	180
7.2.3 用 Mutex 类实现互斥	181
7.2.4 用 Monitor 类实现互斥	182
7.3 生产者线程和消费者线程的同步	182
7.3.1 生产者线程和消费者线程不同步可能发生错误	182
7.3.2 生产者线程和消费者线程同步的实现	183
习题七	184
 第8章 数据库与 ADO.NET	 185
8.1 设计数据库应用程序的基本步骤	185
8.2 用 SQL Server Express 创建数据库	186
8.3 结构化查询语言 (SQL)	187
8.4 连接数据库 (SqlConnection 和 OleDbConnection 类)	188
8.5 OleDbCommand 和 SqlCommand 类	189
8.6 OleDbAdapter 和 SqlDataAdapter 类	190
8.7 DataSet 类对象	190
8.7.1 使用 DataSet 类的必要性	190
8.7.2 使用 DataSet 类对象	191
8.7.3 为 DataSet 对象中的表指定主键、建立关系	192
8.8 数据绑定概念和 BindingSource 组件	192
8.9 DataGridView 控件	193
8.10 BindingNavigator 控件	196

8.11 建立主从关系表	197
8.12 OleDbDataReader 和 SqlDataReader 类	198
8.13 存储过程	199
8.14 DataTable 类	200
8.15 在 SQL Server 数据库中存储图像	201
习题八	202
第 9 章 ASP.NET 编程基础知识	203
9.1 网络基础	203
9.1.1 局域网、广域网和 Internet	203
9.1.2 TCP/IP 和 IP 地址	203
9.1.3 域名和域名服务器	204
9.1.4 Web 网站工作方式和宿主目录	204
9.1.5 URL、超级链接、主页和端口号	205
9.1.6 HTML、HTTP 和网页	205
9.1.7 静态网页和动态网页	206
9.1.8 修改宿主目录及建立虚拟目录	206
9.2 HTML 标记语言	208
9.2.1 HTML 标记和网页文件结构	208
9.2.2 超级链接	210
9.2.3 网页中控件的概念	210
9.2.4 使用 VS2005 可视化创建网页	211
9.2.5 用表格进行页面布局	211
9.2.6 用层进行页面布局	212
9.2.7 CSS 样式文件	213
9.2.8 母版页	214
9.3 ASP.NET 技术基础	216
9.3.1 Web 服务器控件	216
9.3.2 记事本编写动态网页	216
9.3.3 ASP.NET 的事件处理	218
9.3.4 用 VS 2005 创建动态网页	218
9.3.5 Code Behind 技术	219
9.3.6 发布网站	220
9.3.7 复制网站	221
习题九	222
第 10 章 Web 服务器控件	223
10.1 标准 Web 服务器控件	223

10.1.1	Label 控件	223
10.1.2	TextBox 控件	224
10.1.3	Button、LinkButton 和 ImageButton 控件	224
10.1.4	CheckBox 和 CheckBoxList 控件	225
10.1.5	RadioButton 和 RadioButtonList 控件	226
10.1.6	Image 控件	227
10.1.7	HyperLink 控件	228
10.1.8	Table、TableCell 和 TableRow 控件	228
10.1.9	DropDownList 控件	229
10.1.10	ListBox 控件	230
10.2	数据验证控件	230
10.2.1	数据验证概述	230
10.2.2	RequiredFieldValidator 控件	231
10.2.3	ValidationSummary 控件	232
10.2.4	自定义数据验证控件 CustomValidator	233
10.2.5	CompareValidator 控件	235
10.2.6	RangeValidator 控件	235
10.2.7	RegularExpressionValidator 控件	236
10.3	Web 服务器控件数据绑定	239
10.3.1	绑定到其他控件属性	239
10.3.2	绑定到函数返回值	240
10.3.3	绑定到集合类对象	240
10.3.4	数据源控件	241
10.4	GridView 控件	243
10.5	DetailsView 控件	246
10.6	FormView 控件	246
10.7	Repeater 控件	247
10.7.1	Repeater 控件概述	247
10.7.2	用 Repeater 控件显示数据库表	249
10.7.3	Repeater 控件事件 ItemCommand	250
10.8	DataList 控件	251
10.8.1	DataList 控件概述	251
10.8.2	使用 VS2005 编辑 DataList	253
10.9	AdRotator 控件	255
10.10	Calender 控件	257
10.11	登录和注册控件	259
10.11.1	LoginView、Loginname 和 LoginStatus 控件	259
10.11.2	Loging 控件	260
10.11.3	使用网站配置工具增加注册用户	260

10.11.4 注册控件 CreateUserWizard	262
10.11.5 访问规则和角色	263
10.12 VS2005 实现留言板网站	264
10.13 导航控件	266
10.13.1 Web.sitemap 站点地图文件	266
10.13.2 SiteMapPath 控件	267
10.13.3 Menu 控件	267
10.13.4 TreeView 控件	267
习题十	268
第 11 章 ASP.NET 内建对象	270
11.1 Response 和 Request 对象	270
11.2 文件的上传和下载	272
11.3 Cookie 对象	273
11.3.1 Cookie 对象的用法	273
11.3.2 用 Cookie 对象记录访问网站的次数	274
11.3.3 网上商店购物筐实现	275
11.4 Application 对象	277
11.4.1 Application 对象方法和事件	277
11.4.2 Global.asax 文件	278
11.4.3 显示访问网站总人数的例子	278
11.5 Session 对象	280
11.5.1 Session 对象的属性、方法和事件	280
11.5.2 用 Session 对象实现网上商店购物筐	281
11.6 Server 对象	282
11.6.1 Server 对象属性和方法	282
11.6.2 使用 Server.Transfer 在网页之间传递数据	283
11.7 Cache 对象	284
11.8 Web.config 配置文件	285
习题十一	286
第 12 章 可扩展标记语言	288
12.1 XML 可扩展标记语言的基本概念	288
12.1.1 HTML 及其缺点	288
12.1.2 SGML (标准通用标记语言)	288
12.1.3 XML (可扩展标记语言)	289
12.1.4 XML 的文档格式	289
12.1.5 用 DTD 和 XML Schema 定义 XML 架构	290

12.1.6 较复杂的 XML 文档	291
12.2 XML 文档显示	292
12.2.1 用 XSL 文件显示 XML 文档	292
12.2.2 使用 XML 控件显示 XML 文档	294
12.2.3 使用数据绑定方法显示 XML 文档	294
12.2.4 使用 XmlDataSource	296
12.2.5 将 XML 文件转换为 HTML 文件	297
12.3 对 XML 文档的处理	297
12.3.1 使用 XmlTextReader 类读 XML 文件	297
12.3.2 使用 XmlTextReader 类读 XML 文档标记	299
12.3.3 使用 XmlTextWriter 类写 XML 文档	300
12.3.4 文档对象模型 (DOM) 使用	301
12.3.5 用 XML Schema 验证 XML 架构	304
12.4 数据库和 XML	305
习题十二	306
 第 13 章 Web 服务	 308
13.1 Web 服务的概念和用途	308
13.2 建立 Web 服务	309
13.2.1 用记事本建立 Web 服务	310
13.2.2 用 VS 2005 建立 Web 服务	312
13.2.3 服务描述语言 (WSDL)	312
13.3 基于 .NET 的 Web 服务客户端程序	313
13.3.1 调用 Web 服务方法采用的协议	313
13.3.2 VS 2005 建立 Web 服务客户端程序	313
13.3.3 WSDL 程序	315
13.3.4 同步和异步 Web 服务	316
13.3.5 不同系统 Web 服务的互操作	316
13.4 使用 Web 服务的例子	316
13.4.1 .NET 系统 Web 服务支持的数据类型	316
13.4.2 使用 Web 服务返回数据库表	316
13.4.3 用 Web 服务传送图形文件	317
习题十三	318
 参考文献	 319

第 1 章 C#语言基础

本章介绍 C#语言的基础知识, 具有 C 语言基础的读者通过本章的学习, 能够基本掌握 C#语言, 并以此为基础, 能够进一步学习用 C#语言编写 Window 应用程序和 Web 应用程序。当然仅靠一章的内容就完全掌握 C#语言是不可能的, 如需进一步学习 C#语言, 还需要阅读 C#语言的专著。

C 1.1 C#语言特点

Microsoft.NET Framework (微软.NET 架框, 以下简称.NET Framework) 是微软公司提出的新一代软件开发模型, C#语言是.NET Framework 中新一代的开发工具。C#语言是一种现代的、面向对象的语言, 它简化了 C++语言在类、命名空间、方法重载和异常处理等方面的操作, 它摒弃了 C++的复杂性, 更易使用, 更少出错。它使用组件编程, 和 VB 一样容易使用。C#语法和 C++、Java 语法非常相似, 如果用过 C++和 Java, 学习 C#语言会比较轻松。

用 C#语言编写的源程序, 被 C#语言编译器编译为中间语言 (Microsoft Intermediate Language, MSIL) 代码, 形成扩展名为 .exe 或 .dll 文件。中间语言代码不是 CPU 可执行的机器码, 在程序运行时, 必须由通用语言运行环境 (common language runtime, CLR) 中的即时编译器 (Just In Time, JIT) 将中间语言代码翻译为 CPU 可执行的机器码, 由 CPU 执行。CLR 为 C#语言中间语言代码运行提供了一种运行环境, C#语言的 CLR 和 Java 语言的虚拟机类似。这种执行方法使运行速度变慢, 但带来其他一些好处, 主要有以下几方面。

- 通用语言规范 (common language specification, CLS): .NET 系统包括 C#、C++、VB 和 J#语言, 它们都遵守通用语言规范。任何程序设计语言只要遵守通用语言规范, 其源程序都可编译为相同的中间语言代码, 由 CLR 负责执行, 这样的代码叫托管代码。只要为其他操作系统编制相应的 CLR, 中间语言代码也可在其他系统中运行。

- 自动内存管理: CLR 内建垃圾收集器, 当变量实例的生命周期结束时, 垃圾收集器负责收回不被使用的实例占用的内存空间。不必像 C 和 C++语言, 用语句在堆中建立的实例, 必须用语句释放实例占用的内存空间。CLR 具有自动内存管理功能。

- 交叉语言处理: 由于任何遵守通用语言规范的程序设计语言源程序, 都可编译为相同的中间语言代码, 不同语言设计的组件, 可以互相通用, 可以从其他语言定义类派生出本语言的新类。由于中间语言代码由 CLR 负责执行, 因此异常处理方法是一致的, 这在调试