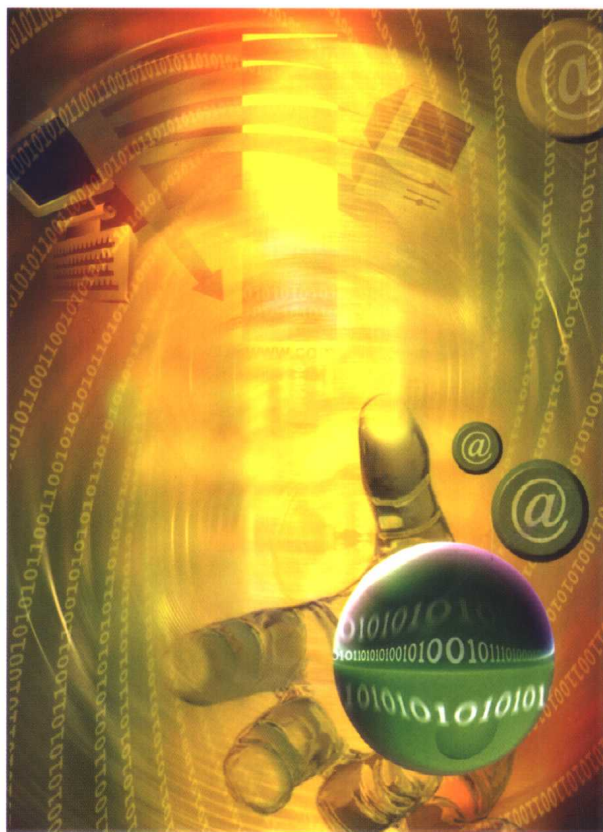


Visual C# 程序设计教程

- ◆ C# 语言基础
- ◆ 结构化程序设计方法
- ◆ 面向对象程序设计方法
- ◆ 常用数据结构与算法
- ◆ 异常处理机制
- ◆ Windows 应用程序开发
- ◆ GDI+ 与图形编程
- ◆ 多线程程序设计
- ◆ 访问文件与注册表
- ◆ 数据库应用程序设计
- ◆ Web 应用程序与 Web 服务开发



王昊亮 李 刚 等编著



清华大学出版社

高等院校计算机应用技术系列教材

Visual C#程序设计教程

王昊亮 李刚 等编著

清华大学出版社

北 京

内 容 简 介

C#是微软公司推出的新一代的编程语言。它功能强大、编程过程简捷、明快。除具有 Visual Basic 语言易学、易用、适合快速程序开发的特性外,同时也继承了 Visual C++的强大功能,是微软公司推行的.NET 战略的重要组成部分。C#借助.NET 框架的强大功能,非常适用于各种应用程序的开发。在当今的越来越复杂的程序设计世界中,使用 C#这件利器无疑会使您的工作事半功倍。全书共分 14 章,从语法、面向对象的编程思想到文件操作、网络编程和数据库应用,深入浅出地、全面地介绍了如何使用 C#语言在.NET 框架下开发各种应用程序。书中含有大量精心设计的代码实例,通过研究这些代码,读者可以深刻地理解和掌握 C#语言的语法和程序设计方法。在这些示例的基础上,读者可以快速而高效地开发出高质量的应用程序。

本书内容丰富、实用、可操作性强、语言生动流畅,没有晦涩的专业术语,能够使读者在轻松愉快的环境下迅速掌握使用 C#语言进行程序设计的方法和技巧。本书特别适用于 C#的初学者,同时也适用于有一定编程经验并想使用 C#语言开发应用程序的专业人员。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Visual C#程序设计教程/王昊亮等编著. —北京:清华大学出版社, 2003.9

ISBN 7-302-07254-X

I. V… II. 王… III. C 语言—程序设计—教材 IV.TP312

中国版本图书馆 CIP 数据核字(2003)第 082864 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:孟毅新

文稿编辑:孟毅新

封面设计:王伟

版式设计:康博

印 刷 者:北京昌平环球印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:24.5 字数:581千字

版 次:2003 年 10 月第 1 版 2003 年 10 月第 1 次印刷

书 号:ISBN 7-302-07254-X/TP·5270

印 数:1~5000

定 价:35.00 元

前 言

很多人曾说过：“Java 是最卓越的程序设计语言！”。不过从 C#诞生的那一刻开始，这已经成了历史。可以说，C#是比尔·盖茨和他的微软把 Java 集成到 Windows 环境之中后的产物。也许我们叫他 Win-Java 更合适一些。而此时的 Windows 已经不再局限在单机之上了，借着 .NET Framework 提供的强大翅膀，Windows 将充满网络无处不在。就在这无处不在的 .NET 之上开发应用程序，C#无疑是最好的选择。

简单地说，Microsoft .NET 是微软以服务方式递交软件的一种策略。它是微软公司的新战略，所有微软的产品都将围绕这个战略开发。Microsoft .NET 能使用户通过 Web 与众多的智能设备交互，同时确保用户而不是应用程序控制这个交互。Microsoft .NET 能使得用户对应用程序、服务、个性化设备的体验简单、一致而安全。

微软为了推行 .NET 战略，特别为 .NET 平台设计了一种新语言——C#。C#是由 C 和 C++发展而来的一种“简单、高效、面向对象、类型安全”的程序设计语言，其综合了 Visual Basic 的高效率和 C++的强大功能。C#是 .NET 的关键语言，是整个 .NET 平台的基础。与 C#相比，.NET 所支持的其他语言显然是配角身份，包括 VC++ .NET 在内。但是微软并没有打算放弃 VC++ .NET，相反微软对 VC++ .NET 有着另一番独特的打算，VC++ .NET 的定位与 C#不完全重合，VC++ .NET 应用范围仍强于 C#。这一点无论对微软公司还是软件业应用现状都非常重要。

可见 VC++ .NET 在 .NET 框架中已经退化成了底层的语言，就像 C++之前的标准 C 语言一样，以后将只有开发底层或桌面程序的程序员将使用它了。它虽然仍旧强大，但是已经不是主流。而主流就是 C#语言。

本书的阅读对象是所有对 C#和 .NET Framework 技术感兴趣的读者，特别是希望能把握互联网所带来机遇的应用开发者。

本书的读者可以是编程的入门者，甚至可以完全没有编程经验。本书从基本的语法开始，一步一步、深入浅出地介绍如何使用 C#语言进行各种应用程序的设计。如果读者以前有 Java 或 C、C++语言的基础，那么阅读本书可以使您事半功倍，并迅速掌握 C#语言与其他语言的不同之处和编程方法。通过学习本书的实例并借助 C#语言的强大功能，您可以很快学习到如何进行数据库、网络等复杂程序设计，进入到中级程序员的境界。

本书的特点是由浅入深，从最基本的语法开始讲起，逐步深入到面向对象、Windows 程序设计、数据库、网络等高级编程方法。在介绍语法时，本书并没有像一些语法书一样教条而死板地讲定义，而是试图利用示例代码生动地让读者在实践中体会一个个知识点。通过认真分析示例代码和书中的讲解，读者可以很快掌握 C#语言的精髓。

本书的每一章后面都有上机练习和习题。这些练习都是作者精心编排的，认真思考这些题目是掌握各重点的关键。

本书内容安排如下:

第1章对 Visual Studio .NET、C#的历史和意义、特点进行了简单而扼要的介绍。第2章介绍 C#了语言的基本数据类型、运算符和表达式。第3章介绍了 C#语言的基本流程控制方法。第4章介绍了面向对象的编程思想和 C#语言对面向对象的支持。第5章深入介绍了面向对象的其他知识点,例如委托、事件、索引等。第6章主要介绍 C#语言提供的各种数据结构,并简要介绍如何使用 C#语言实现算法。第7章介绍组件编程的关键技术——接口,它是 C#直接提供的一种专门面向协议的类型,是 C#的一大特点。第8章介绍了 C#的异常处理方法。第9章介绍如何在 Visual Studio .NET 中用 C#设计 Windows 应用程序,说明了 Visual Studio .NET 提供的各种控件的用法。第10章介绍了如何使用 GDI+图形接口绘制图形。第11章介绍了设计多线程应用程序的方法。第12章介绍了文件、目录的操作方法,以及在程序中修改注册表的技术。第13章介绍了数据库编程的方法,并以一个数据访问实例充分体现了 ADO.NET 在数据库访问上的便利与优势。第14章初步介绍了 ASP.NET 与 Web 服务的概念和设计方法。

本书由王昊亮、李刚和陈昕共同执笔编写,此外,张维、吴轶秦、韩璐、王瑾、吴君华、付鑫育、李龙、钱少伟、刘荣强、李伟光、朱峰、许大中、魏勇、萧玉、丁桦、李林、邵华刚、朱莉、肖育新、戴礼华、钟行兆、马军、李志盛、柳欢兵、关政、徐方方、钟华和王昊平等同志在整理材料和调试代码方面给予了作者很大的帮助。在此向他们表示深深的谢意。

由于作者水平有限,在内容选材和叙述上难免有不当之处。欢迎广大读者对本书提出批评和建议。

作 者

目 录

第 1 章 Visual C# 简介	1
1.1 .NET Framework 1.1 简介	1
1.1.1 .NET Framework	1
1.1.2 .NET Framework 1.1 版本的新功能	3
1.2 Visual C# .NET 简介	4
1.2.1 Visual C# .NET 的产生背景	4
1.2.2 Visual C# .NET 语言的特点	5
1.3 Visual Studio .NET 2003 简介	6
1.3.1 Visual Studio .NET 2003 的主要作用	6
1.3.2 Visual Studio .NET 2003 的新功能	7
1.4 编写第一个 C# 程序	8
1.4.1 使用 Visual Studio .NET IDE 创建 C# 程序	8
1.4.2 编译 C# 应用程序	11
1.5 C# 程序的结构	13
1.6 本章小结	14
1.7 上机练习	14
1.8 习题	15
第 2 章 数据类型、运算符和表达式	16
2.1 数据类型	16
2.1.1 值类型	16
2.1.2 引用类型	20
2.1.3 装箱与拆箱	23
2.2 常量和变量	24
2.2.1 变量	24
2.2.2 常量	28
2.3 运算符	29
2.3.1 算术运算符	29
2.3.2 赋值运算符	31
2.3.3 关系运算符	31
2.3.4 逻辑运算符	32
2.3.5 位运算符	32
2.3.6 条件运算符	34

2.3.7 其他运算符	34
2.4 表达式	35
2.5 本章小结	38
2.6 上机练习	38
2.7 习题	40
第3章 结构化程序设计	41
3.1 条件语句	41
3.2 循环语句	43
3.2.1 while 语句	43
3.2.2 do-while 语句	44
3.2.3 for 语句	45
3.2.4 foreach 语句	46
3.3 分支语句	47
3.4 跳转语句	48
3.4.1 break 语句和 continue 语句	48
3.4.2 标签语句和 goto 语句	49
3.5 异常处理结构	50
3.6 using 语句	52
3.7 本章小结	54
3.8 上机练习	54
3.9 习题	56
第4章 面向对象程序设计基础	57
4.1 面向对象概述	57
4.1.1 类和对象概述	57
4.1.2 属性、方法、继承、重载	58
4.2 类和对象	60
4.2.1 类的声明	60
4.2.2 类的成员	62
4.2.3 声明对象	64
4.3 构造函数与析构函数	65
4.3.1 构造函数	66
4.3.2 析构函数	68
4.4 方法与重载	68
4.4.1 方法的声明	69
4.4.2 方法的重载	71
4.5 运算符重载	74

4.6 继承与多态	75
4.6.1 类的继承	76
4.6.2 类的多态性	78
4.7 本章小结	79
4.8 上机练习	80
4.9 习题	80
第5章 深入理解面向对象	81
5.1 类的深入讨论	81
5.1.1 类成员的可访问性	81
5.1.2 类的静态成员	84
5.1.3 this 关键字	85
5.1.4 方法的种类	86
5.1.5 密封类与密封方法	88
5.2 域、属性和事件	88
5.2.1 域	89
5.2.2 属性	91
5.2.3 事件	93
5.3 面向对象程序设计实例	96
5.4 命名空间	102
5.4.1 命名空间的声明	102
5.4.2 命名空间的成员	104
5.4.3 命名空间的使用	105
5.5 本章小结	111
5.6 上机练习	111
5.7 习题	111
第6章 常用数据结构与算法	112
6.1 字符串	112
6.1.1 字符串类型定义	112
6.1.2 字符串类型的应用	114
6.2 数组	120
6.2.1 数组的定义	121
6.2.2 数组的使用	123
6.3 集合	127
6.3.1 集合的定义	128
6.3.2 集合的使用	129
6.4 枚举	137

6.4.1 枚举类型的定义	138
6.4.2 枚举型的应用	140
6.5 搜索算法	143
6.5.1 顺序搜索	143
6.5.2 折半搜索	146
6.6 排序	148
6.6.1 直接插入排序	148
6.6.2 快速排序	152
6.6.3 选择排序	155
6.7 本章小结	157
6.8 上机练习	158
6.9 习题	158
第7章 接口	160
7.1 接口的意义	160
7.1.1 三层体系结构	160
7.1.2 组件编程思想	161
7.2 接口的定义	163
7.3 接口的成员	164
7.3.1 接口属性	165
7.3.2 接口事件	165
7.3.3 接口索引	165
7.3.4 接口成员的访问	166
7.3.5 接口成员的完全有效名称	168
7.4 接口的实现	168
7.4.1 在类中实现接口	168
7.4.2 显式接口成员执行体	170
7.4.3 接口映射	175
7.4.4 接口实现的继承	180
7.4.5 接口的重实现	181
7.4.6 抽象类与接口	183
7.5 接口应用实例	184
7.6 本章小结	187
7.7 上机练习	187
7.8 习题	187
第8章 异常处理	188
8.1 异常处理概念	188

8.2	异常类	190
8.3	异常的处理	191
8.4	把异常传给调用者	195
8.4.1	调用者处理	195
8.4.2	抛出异常	196
8.4.3	重发异常	197
8.4.4	添加异常信息	198
8.5	用户自定义异常	199
8.6	finally 的用法	201
8.7	异常处理实例	203
8.8	本章小结	205
8.9	上机练习	205
8.10	习题	205
第 9 章	Windows 应用程序设计	206
9.1	Windows 应用程序的结构	206
9.1.1	计算器窗体	207
9.1.2	计算器控件	208
9.1.3	计算器事件	209
9.2	窗体控件	211
9.2.1	窗体概述	211
9.2.2	多文档界面设计	213
9.3	控件共有的属性、事件和方法	216
9.4	常用控件	229
9.4.1	标签	229
9.4.2	按钮	230
9.4.3	文本框	230
9.4.4	单选按钮	231
9.4.5	复选框	231
9.4.6	组合框	232
9.4.7	列表框	233
9.4.8	进度条	235
9.4.9	滚动条	236
9.4.10	菜单	236
9.4.11	工具栏	237
9.4.12	编辑控件	239
9.5	定制控件	240

9.5.1 关于定制控件	240
9.5.2 定制控件示例	241
9.6 本章小结	245
9.7 上机练习	245
9.8 习题	246
第 10 章 GDI+与图形编程	247
10.1 .NET 中的图形绘制	247
10.1.1 GDI+简介	247
10.1.2 图形类库	247
10.1.3 绘图方法	248
10.2 GDI+中的图形对象	251
10.2.1 Pen 对象	252
10.2.2 Color 结构	252
10.2.3 Font 对象	253
10.2.4 Brush 对象	254
10.2.5 矩形结构	258
10.3 绘图实例	259
10.4 本章小结	266
10.5 上机练习	267
10.6 习题	267
第 11 章 多线程程序设计	268
11.1 线程的概念	268
11.2 C#多线程编程概述	269
11.3 多线程编程实例	270
11.4 线程同步	283
11.4.1 没有同步	283
11.4.2 一写多读同步	285
11.4.3 多读多写同步	286
11.4.4 使用 AutoResetEvent	288
11.5 增强多线程的交互性	289
11.5.1 交互性问题	289
11.5.2 简单实例	290
11.5.3 高级多线程实例	291
11.6 本章小结	295
11.7 上机练习	295
11.8 习题	296

第 12 章 文件与注册表	298
12.1 C#的文件处理系统	298
12.1.1 认识 C#的文件处理系统	298
12.1.2 文件和流	299
12.2 文件处理	299
12.2.1 目录管理	299
12.2.2 文件管理	301
12.3 读写文件	306
12.3.1 StreamReader 类	306
12.3.2 StreamWriter 类	306
12.4 读写注册表	308
12.4.1 注册表概述	308
12.4.2 C#对注册表编程的支持	310
12.4.3 注册表编程示例	314
12.5 本章小结	317
12.6 上机练习	318
12.7 习题	318
第 13 章 数据访问	319
13.1 ADO.NET 概述	319
13.1.1 ADO.NET 的历史	319
13.1.2 ADO.NET 的特点	320
13.1.3 SQL 语言简介	321
13.1.4 ADO 与 ADO.NET 的比较	323
13.2 ADO.NET 组件和对象	324
13.2.1 ADO.NET 组件和组件之间的关系	324
13.2.2 数据集组件	325
13.2.3 数据适配器组件	325
13.3 ADO.NET 编程	326
13.3.1 连接	326
13.3.2 数据集编程	327
13.3.3 数据绑定	328
13.3.4 Windows 窗体和 Web 页面	329
13.4 ADO.NET 可视化编程	329
13.4.1 创建 Access 数据库	330
13.4.2 创建新数据库项目	331
13.4.3 创建数据库连接	331

13.4.4	创建数据库对象和访问方法	339
13.4.5	设置程序显示界面	340
13.4.6	显示数据库中的数据	341
13.5	本章小结	341
13.6	上机练习	342
13.7	习题	342
第 14 章	开发 Web 应用程序	343
14.1	什么是 Web 应用程序	343
14.1.1	Web 程序发展简史	344
14.1.2	Web 程序设计的要素	345
14.2	设计 Web 窗体	345
14.3	创建和使用 Web 服务	356
14.3.1	分布式应用程序模式和浏览器的作用	357
14.3.2	什么是 Web 服务	358
14.3.3	Web 平台中的协议	359
14.3.4	Web 服务的实现	360
14.4	本章小结	368
14.5	上机练习	369
14.6	习题	369
附录 A	C#预处理指令	370

第1章 Visual C#简介

本章介绍了 .NET Framework、Visual C# .NET 的发展历程及特点, C#程序的结构及 C#常用编辑器、编译器的使用, 并通过一个简单的 C#应用程序介绍 C#应用程序的开发方法。

本章要点:

- .NET Framework 的功能
- Visual Studio .NET 开发工具包的使用
- 认识 C#应用程序

1.1 .NET Framework 1.1 简介

.NET 技术是微软公司推出的一个全新概念, 它代表了一个集合、一个环境、一个编程的基础结构。其目的是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行延伸。 .NET Framework、Visual Studio .NET 和 Visual C#语言就是 .NET 技术的重要组成部分。

1.1.1 .NET Framework

.NET Framework 是一个用于生成、部署和运行 XML Web 服务和应用程序的多语言环境。 .NET Framework 是一种新的计算平台, 它简化了在高度分布式 Internet 环境中的应用程序开发。 .NET Framework 旨在实现下列目标:

- 无论是在本地存储并在本地执行, 还是在本地存储在远程执行, 或者在 Internet 上分布但在本地执行的对象代码, .NET Framework 都提供了一致的面向对象的编程环境。
- 提供将软件部署和版本控制冲突最小化的代码执行环境。
- 提供保证代码安全执行的代码执行环境。
- 提供可消除脚本环境或解释环境性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- 按照工业标准生成所有通信, 以确保基于 .NET Framework 的代码可与任何其他代码集成。

.NET Framework 有两个主要组件: 公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础。您可以将运行库看作一个在执行时管理代码的代理,

它提供核心服务(如内存管理、线程管理和远程处理),而且还强制实施严格的类型安全,以及可确保安全性和可靠性的其他形式的代码准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码,而不以运行库为目标的代码称为非托管代码。.NET Framework 的另一个主要组件是类库,它是一个综合性的面向对象的可重用类型集合,您可以使用它开发包含从传统的命令行或图形用户界面 (GUI) 应用程序到基于 ASP.NET 所提供的创新的应用程序(如 Web 窗体和 XML Web 服务)在内的应用程序。

.NET Framework 可由非托管组件承载,这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行,从而创建一个可以同时利用托管和非托管功能的软件环境。.NET Framework 不但提供若干个运行库主机,而且还支持第三方运行库主机的开发。例如,ASP.NET 承载运行库以为托管代码提供可伸缩的服务器端环境。ASP.NET 直接使用运行库以启用 Web 窗体应用程序和 XML Web 服务。

Internet Explorer 是承载运行库(以 MIME 类型扩展的形式)的非托管应用程序的一个示例。使用 Internet Explorer 承载运行库使您能够在 HTML 文档中嵌入托管组件或 Windows 窗体控件。以这种方式承载运行库使得托管移动代码(类似于 ActiveX 控件)成为可能,但是它具有只有托管代码才能提供的重大改进(如不完全受信任的执行和安全的独立文件存储)。

图 1-1 显示了 .NET Framework 环境的组成结构。

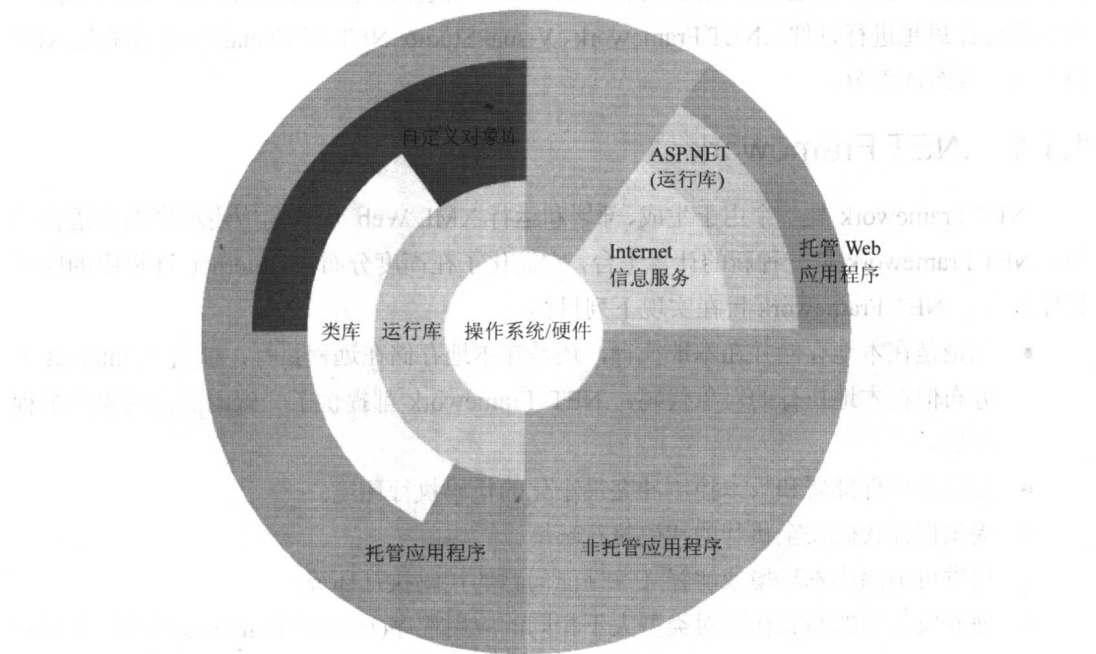


图 1-1 .NET Framework 环境的组成结构

从层次结构来看,.NET Framework 包含以下三个主要部分:公共语言运行库、统一编程类与 Web 应用程序,即 ASP.NET。

公共语言运行库(Common Language Runtime, CLR)是 .NET Framework 应用程序的执行

引擎,它提供包括代码管理、类型安全验证等许多服务。CLR 实际上在组件的运行和开发时都起作用。在组件运行时,CLR 除了负责满足此组件在其他组件上可能具有的依赖项外,还负责管理内存分配、启动和停止线程和进程,以及强制执行安全策略。在开发时,CLR 的作用稍有变化,由于做了大量的自动处理工作(如自动垃圾回收),CLR 使开发人员的操作非常简单。

统一编程类为开发人员提供了统一的、面向对象的、分层的和可扩展的类库集。目前,C++开发人员使用 Microsoft 基础类,而 Java 开发人员使用 Windows 基础类。.NET Framework 统一了这些完全不同的模型,并且为 Visual Basic 和 JScript 程序员提供了对类库的访问机制。通过创建所有编程语言的公共类库集,公共语言运行库使得跨语言继承、错误处理和调试成为可能。从 JScript 到 C++的所有编程语言具有对框架相似的访问,开发人员可以自由选择要使用的语言。

ASP.NET 建立在.NET Framework 的统一编程类之上,它提供了一个 Web 应用程序模型,还包含控件集和结构,这些控件集和结构使得生成 ASP.NET Web 应用程序变得更容易。ASP.NET 包含了封装了 HTML 公共用户界面元素(如网页上的文本框和下拉菜单等)的控件集,使这些控件在 Web 服务器上运行,并以 HTML 的形式将用户界面发送到浏览器。

1.1.2 .NET Framework 1.1 版本的新功能

.NET Framework 1.1 版包括.NET Framework 1.1 SDK、用于开发和运行 J#应用程序的附加程序。微软.NET Framework 1.1 版扩展了.NET Framework 1.0 版,并且增加了很多新的特征,改善了原有的功能特性,同时完善了文档部分。其主要特征如下:

(1) 提供了对开发可移动 Web 应用程序的本地支持。ASP.NET 移动控件(ASP.NET Mobile Control,以前称为 Microsoft Mobile Internet Toolkit)扩展了原来的 ASP 服务器端控件。

(2) 粒度版本控制和并肩(Side-by-side)执行。即在.NET Framework 1.1 中允许管理员同时在一台电脑上存储和执行一个应用程序或者组件的多个版本。

(3) 允许来自 Internet 的 Windows 窗体程序集执行。

(4) 支持 ASP.NET 应用程序的代码访问安全设置。系统管理员可以使用代码访问安全功能进一步控制 ASP.NET Web 应用程序和 XML Web 服务的访问权限。

(5) 对 ODBC 以及 Oracle 数据库进行通信的本地支持。.NET Framework ODBC 数据提供者(Data Provider for ODBC),现在和.NET Framework 一起提供给用户,位于 System.Data.Odbc 命名空间下。它提供了和 OLE DB .NET Data Provider 同样的对本地 ODBC 驱动的访问能力,此外还支持对 Oracle 数据库的访问。

(6) 对小型智能消费类电子设备应用开发提供了统一的程序设计模型。Microsoft .NET 紧凑框架向小型设备(包括 Pocket PC 2000、Pocket PC2002、Pocket PC 2002 电话版以及定制设计的嵌入式设备)提供了公共语言运行库环境,以及 Windows 窗体控件和其他的.NET Framework 功能特性。

- (7) 支持 IPv6。
- (8) 改善了可扩展性和性能。

1.2 Visual C# .NET 简介

C#是微软根据.NET 战略开发的一种新语言。究竟什么是 C#? 它和 C、C++有何异同? 通过下面的介绍, 读者可以对 C#及其特性有大致地了解。

1.2.1 Visual C# .NET 的产生背景

1995 年, SUN 公司正式推出了面向对象的开发语言 Java, 并提出了跨平台、跨语言的概念后, Java 就逐渐成为企业级应用系统开发的首选工具, 而且使得越来越多的基于 C/C++ 的应用开发人员转向了从事基于 Java 的应用开发。Java 的先进思想使其在软件开发领域前景可观。

很快, 在众多研发人员的努力下, 微软也推出了基于 Java 语言的编译器 Visual J++。Visual J++在最短的时间里由 1.1 版本升级到了 6.0 版本。Visual J++ 6.0 集成在 Visual Studio 6.0 中, 不但虚拟机(Java Virtual Machine, JVM)的运行速度大大加快, 而且增加了许多新特性, 同时支持调用 Windows API, 这些特性使得 Visual J++成为强有力的 Windows 应用开发平台, 并成为业界公认的优秀 Java 编译器。

Visual J++虽然具有强大的开发功能, 但主要应用在 Windows 平台的系统开发中, SUN 公司认为 Visual J++违反了 Java 的许可协议, 即违反了 Java 开发平台的中立性, 因而, 对微软提出了诉讼, 这使得微软处于极为被动的局面。为了改变这种局面, 微软另辟蹊径, 决定推出其进军互联网的庞大.NET 计划, 和该计划中重要的开发语言——C#。

微软的.NET 是一项非常庞大的计划, 也是微软今后几年发展的战略核心。Visual Studio .NET 则是微软.NET 技术的开发平台, C#就集成在 Visual Studio .NET 中。为了支持.NET 平台, Visual Studio .NET 在原来的 Visual Studio 6.0 的基础上进行了极大的修改和变更。在 Visual Studio .NET 测试版中 Visual J++就消失了, 取而代之的就是 C#语言。

集成在 Visual Studio .NET 中的微软新的开发语言 C#引起了众多开发者的瞩目。设计新的开发语言 C#, 对微软未来的发展有着举足轻重的意义。微软组织强大的人力, 很快成功研发出了 C#。并于 2000 年 6 月份举行的“职业开发人员技术大会”上正式发布了 C#语言。其英文名为 C-Sharp。微软公司对 C#的定义是: “C#是一种类型安全的、现代的、简单的, 由 C 和 C++衍生出来的面向对象的编程语言, 它是牢牢根植于 C 和 C++语言之上的, 并可立即被 C 和 C++开发人员所熟悉。C#的目的就是综合 Visual Basic 的高生产率和 C++的行动力。”