



.NET 开发专家

C#程序员开发手册

/HOPE/.NET/programming

更多图书请访问 www.bhp.com.cn

HOP^E programming

.NET 框架

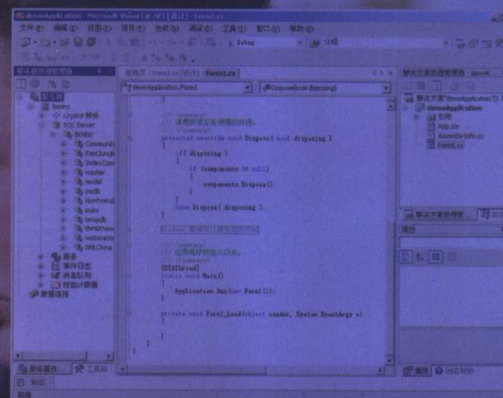
C# 类库

公共属性

公共方法

应用实例

**C# 核心类库
完整解析**



北京希望电子出版社 总策划
杨宏伟 李 晶 等 编 著

科学出版社
www.sciencep.com



.NET 开发专家

C#程序员开发手册

/HOPE/.NET/programming

更多图书请访问 www.bhp.com.cn √ 90

HOPE programming

.NET 框架

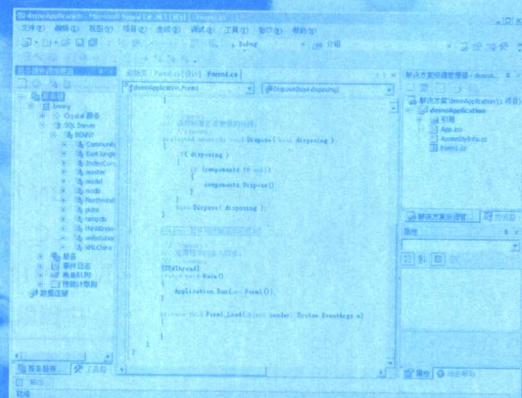
C# 类库

公共属性

公共方法

应用实例

C# 核心类库 完整解析



北京希望电子出版社 总策划
杨宏伟 李 晶 等 编 著

科学出版社
www.sciencep.com

内 容 简 介

本书是 C# 应用程序开发人员必备的工具书。本书分别从继承关系、公共属性和公共方法等角度详细介绍了 C# 类库,并以实例说明了各名字空间下方法及属性的具体应用,内容全面,范例丰富、实用。

本书共分两个部分,第 1 部分介绍了 C# 的基本应用类,从第 2 章~第 12 章,分别讲述了 C# 的命名空间及其相关的类。第 2 部分讲述 C# 的高级应用类,从第 13~第 26 章,介绍了应用 C# 开发项目程序所涉及的类。另外,第 1 章还介绍了 C# 及 Visual Studio.NET 框架结构的基本知识。

本书以命令空间→公共属性→公共方法→举例为序,便于查询,是 C# 应用项目开发人员必备的手边工具书;适合 Visual Studio .NET 爱好者和可视化程序设计培训班使用。

图书在版编目 (CIP) 数据

C# 程序员开发手册/杨宏伟,李晶等编著.—北京:科学出版社,2006.5

ISBN 7-03-015293-X

I. C... II. ①杨...②李... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 026904 号

责任编辑:但明天 / 责任校对:韩波波
责任印刷:双青 / 封面设计:刘孝琼

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2006 年 5 月 第 一 版 开本: 899×1194
2006 年 5 月 第一次印刷 印张: 49
印数: 1—3 000 册 字数: 1150000

定价: 69.00 元

前 言

C#语言由 C/C++ 演变而来,它在带来对应用程序的快速开发能力的同时,并没有牺牲 C/C++ 的优点。它简化了 C++ 在类、命名空间、方法重载和异常处理等领域的工作,摒弃了 C++ 的复杂性,使它易使用、更少出错。C# 主要特点包括:简单、面向对象、与 Web 紧密结合、完整的安全性、错误处理、版本控制以及灵活性与兼容性等。

现代编程技术的核心是面向对象技术、组件技术、代码可重用性是这些技术的关键。而要实现代码可重用性,则必须依赖类库,类库也就成为吸引程序员的一个重要部分。作为功能强大的新一代编程语言 C#,对类库的依赖更是必不可少。本书着眼于 C# 语言编程当中,程序员必须掌握的类以及该类与其他相关类的相互调用关系,利用简单易懂的例子来说明比较错综复杂的关系。作为一个 C# 程序员,必须熟练掌握常用的类。而很多参考书对 C# 相关的类基本上都是回避或者提之甚少,所以一本说明常用的在实际编程中必不可少的 C# 类库方面的书籍无疑是他们的良师益友。

本书分两大部分共 26 章,第 1 章简单地介绍了一下 .NET 的体系结构以及 C# 产生的原因,并从整体上说明了 C# 类库的概况,整个 C# 语言是建立在 .NET 基础之上的,C# 类库也是由 .NET 提供的。第 1 部分为基本应用类,从第 2 章~第 12 章介绍了 C# 编程中的命名空间以及相应的类,有了这些类程序员就可以独立完成一些 C# 项目的设计、开发、维护等任务了。第 2 部分是高级应用类,从第 13 章~第 26 章,介绍了 C# 的高级应用所涉及到的类,有了这部分内容,程序员就可以完成更大的 C# 项目。

对于每个类,首先概述该类的作用,然后分别从继承关系、构造函数、公共属性和公共方法等角度介绍。作者从基于自己的长期教学和开发经验,列出每个类的所有公共属性和公共方法,以便查询。同时,针对重要的、常用的属性和方法展开陈述,并适时地提供一些小程序以方便读者掌握当前介绍的主题。

本书由杨宏伟、李晶、陈蓓和刘向军等组织编写,龚波、刘宪军、强秀丽、李红玲、田军和张丽等提供很多很好的素材,并对文稿进行润色和修改。田野、龚志翔、牛献忠和刘聪对本书目录结构和组织方式提出很好的意见和建议,并参与部分章节的代码调试。赵军锁、白红利、于自跃、姜南和李志最后进行全书通稿。张溟、赵欣胜等负责输入书稿,并进行排版。其他参与本书代码调试和部分章节编写的人员还有:陈磊、冯军、刘砚、任志宏、郑启迪、周松建、陈发吉、于冰、蒋红霞、龚建、施燕妹、罗贤锋、葛丽、韩存兵、刘卫宏、李小将、邓波、王东霞、田丽韞等。希望出版社第 3 编辑室的老师们严肃认真的工作态度进一步保证了本书的质量。

由于作者水平有限,虽殚精竭虑,其中难免存在不合适的地方,希望各位读者和专家赐教。我们的联系邮件是 lihl_mail@163.com。

作 者

目 录

第 1 章 .NET 框架及类库	1	5.5 StreamReader 类	119
1.1 使用 C#语言的原因	1	5.6 StreamWriter 类	123
1.2 .NET Framework 中的命名空间	2	5.7 Stream 类	126
1.3 与 Windows 系统协合	3	5.8 BufferedStream 类	132
1.4 .NET 核心类	4	5.9 Directory 类	135
第 2 章 Microsoft.Win32 命名空间	8	5.10 DirectoryInfo 类	144
2.1 PowerModeChangedEventArgs 类	8	5.11 MemoryStream 类	149
2.2 Registry 类	8	5.12 Path 类	153
2.3 SystemEvents 类	9	第 6 章 System.Reflection 命名空间	157
2.4 RegistryKey 类	10	6.1 Assembly 类	158
第 3 章 System 命名空间	16	6.2 AssemblyName 类	163
3.1 Activator 类	18	6.3 ConstructorInfo 类	166
3.2 Array 类	22	6.4 FieldInfo 类	169
3.3 Console 类	29	6.5 Module 类	172
3.4 Convert 类	34	6.6 PropertyInfo 类	175
3.5 Delegate 类	38	第 7 章 System.Text 命名空间	180
3.6 Exception 类	41	7.1 Encoding 类	180
3.7 MarshalByRefObject 类	43	7.2 Decoder 类	186
3.8 Math 类	45	7.3 StringBuilder 类	187
3.9 Object 类	47	7.4 UnicodeEncoding 类	192
3.10 OverflowException 类	49	7.5 UTF8Encoding 类	195
3.11 Random 类	51	第 8 章 System.Text.RegularExpressions	
3.12 String 类	53	命名空间	201
3.13 Type 类	57	8.1 Capture 类	201
3.14 Uri 类	62	8.2 CaptureCollection 类	203
第 4 章 System.Collections 命名空间	66	8.3 Group 类	204
4.1 ArrayList 类	66	8.4 GroupCollection 类	205
4.2 BitArray 类	72	8.5 Match 类	206
4.3 CollectionBase 类	76	8.6 MatchCollection 类	208
4.4 DictionaryBase 类	79	8.7 Regex 类	209
4.5 Hashtable 类	82	8.8 RegexCompilationInfo 类	215
4.6 Queue 类	85	第 9 章 System.Threading 命名空间	217
4.7 SortedList 类	89	9.1 AutoResetEvent 类	217
4.8 Stack 类	93	9.2 Monitor 类	219
第 5 章 System.IO 命名空间	98	9.3 Mutex 类	223
5.1 File 类	98	9.4 ReaderWriterLock 类	225
5.2 FileStream 类	104	9.5 Thread 类	228
5.3 BinaryReader 类	111	9.6 ThreadPool 类	231
5.4 BinaryWriter 类	115	9.7 Timer 类	235

9.8 WaitHandle 类	238	13.6 DataRelationCollection 类	373
第 10 章 System.Timers 命名空间	241	13.7 DataRow 类	375
10.1 ElapsedEventArgs 类	241	13.8 DataRowCollection 类	380
10.2 Timer 类	242	13.9 DataRowView 类	382
10.3 TimersDescriptionAttribute 类	244	13.10 DataSet 类	385
第 11 章 System.Windows.Forms 命名空间	246	13.11 DataTable 类	392
11.1 Application 类	251	13.12 DataTableCollection 类	396
11.2 Control 类	254	13.13 DataView 类	399
11.3 Form 类	262	第 14 章 System.Net 命名空间	403
11.4 TextBox 类	269	14.1 Dns 类	404
11.5 ComboBox 类	274	14.2 DnsPermission 类	406
11.6 Clipboard 类	281	14.3 HttpWebRequest 类	409
11.7 Cursor 类	282	14.4 HttpWebResponse 类	412
11.8 DataFormats 类	285	14.5 IPAddress 类	414
11.9 DataGrid 类	288	14.6 IPEndPoint 类	417
11.10 DataGridColumnStyle 类	296	14.7 IPEndPoint 类	418
11.11 DataObject 类	297	14.8 SocketAddress 类	419
11.12 InputLanguage 类	300	14.9 WebClient 类	421
11.13 Menu 类	302	14.10 WebPermission 类	424
11.14 Screen 类	304	14.11 WebProxy 类	428
11.15 ToolTip 类	307	14.12 WebRequest 类	430
11.16 TreeNode 类	309	14.13 WebResponse 类	434
11.17 TreeView 类	312	第 15 章 System.Runtime.Remoting 命名空间	436
第 12 章 System.Drawing 命名空间	318	15.1 ActivatedClientTypeEntry 类	436
12.1 Bitmap 类	319	15.2 ActivatedServiceTypeEntry 类	438
12.2 Brush 类	323	15.3 ObjectHandle 类	439
12.3 ColorTranslator 类	324	15.4 ObjRef 类	441
12.4 Font 类	326	15.5 RemotingServices 类	443
12.5 FontFamily 类	329	15.6 WellKnownClientTypeEntry 类	446
12.6 Graphics 类	332	第 16 章 System.Net.Sockets 命名空间	448
12.7 Icon 类	340	16.1 NetworkStream 类	448
12.8 Image 类	342	16.2 Socket 类	451
12.9 ImageAnimator 类	345	16.3 TcpClient 类	457
12.10 Pen 类	347	16.4 TcpListener 类	459
12.11 Region 类	350	16.5 UdpClient 类	462
12.12 StringFormat 类	354	第 17 章 System.Globalization 命名空间	466
12.13 TextureBrush 类	356	17.1 Calendar 类	466
第 13 章 System.Data 命名空间	360	17.2 CompareInfo 类	471
13.1 Constraint 类	361	17.3 CultureInfo 类	476
13.2 ConstraintCollection 类	363	17.4 DateTimeFormatInfo 类	479
13.3 DataColumn 类	366	17.5 GregorianCalendar 类	483
13.4 DataColumnCollection 类	368	17.6 NumberFormatInfo 类	487
13.5 DataRelation 类	371	17.7 RegionInfo 类	490

17.8	SortKey 类	492	23.1	OdbcCommand 类	575
17.9	StringInfo 类	493	23.2	OdbcCommandBuilder 类	578
17.10	TextElementEnumerator 类	495	23.3	OdbcConnection 类	581
17.11	TextInfo 类	497	23.4	OdbcDataAdapter 类	584
第 18 章	System.Management 命名空间	499	23.5	OdbcDataReader 类	586
18.1	ConnectionOptions 类	500	23.6	OdbcParameter 类	589
18.2	EventQuery 类	502	23.7	OdbcTransaction 类	591
18.3	ManagementBaseObject 类	504	第 24 章	System.Diagnostics 命名空间	593
18.4	ManagementClass 类	506	24.1	ConditionalAttribute 类	594
18.5	ManagementObject 类	511	24.2	Debug 类	596
18.6	ManagementPath 类	514	24.3	EventLog 类	599
18.7	WqlObjectQuery 类	516	24.4	EventLogEntry 类	603
第 19 章	System.Data.SqlClient 命名空间	518	24.5	EventLogEntryCollection 类	605
19.1	SqlCommand 类	518	24.6	PerformanceCounter 类	607
19.2	SqlCommandBuilder 类	522	24.7	Process 类	612
19.3	SqlConnection 类	524	24.8	ProcessModule 类	613
19.4	SqlDataAdapter 类	527	24.9	ProcessThread 类	615
19.5	SqlDataReader 类	530	24.10	ProcessStartInfo 类	617
19.6	SqlTransaction 类	533	24.11	StackFrame 类	619
第 20 章	System.DirectoryServices 命名空间	536	24.12	StackTrace 类	621
20.1	DirectoryEntries 类	536	24.13	TextWriterTraceListener 类	624
20.2	DirectoryEntry 类	538	第 25 章	System.Drawing.Printing 命名空间	627
20.3	DirectorySearcher 类	541	25.1	Margins 类	627
20.4	PropertyCollection 类	544	25.2	PageSettings 类	629
20.5	PropertyValueCollection 类	545	25.3	PaperSource 类	631
20.6	SchemaNameCollection 类	548	25.4	PrintDocument 类	631
20.7	SearchResult 类	549	25.5	PrinterSettings 类	634
20.8	SearchResultCollection 类	551	25.6	PrintPageEventArgs 类	636
第 21 章	System.Security.Cryptography		25.7	PrinterResolution 类	637
	X509Certificates 命名空间	554	25.8	PaperSizeCollection 类	638
21.1	X509Certificate 类	554	25.9	PaperSourceCollection 类	639
21.2	X509CertificateCollection 类	557	25.10	QueryPageSettingsEventArgs 类	640
21.3	X509CertificateEnumerator 类	560	第 26 章	System.Web 命名空间	642
第 22 章	System.Collections.Specialized		26.1	HttpApplication 类	643
	命名空间	562	26.2	HttpApplicationState 类	645
22.1	HybridDictionary 类	562	26.3	HttpBrowserCapabilities 类	648
22.2	ListDictionary 类	565	26.4	HttpClientCertificate 类	649
22.3	NameObjectCollectionBase 类	568	26.5	HttpContext 类	651
22.4	StringDictionary 类	570	26.6	HttpResponse 类	653
22.5	StringEnumerator 类	573	26.7	HttpServerUtility 类	656
第 23 章	System.Data.Odbc 命名空间	575	26.8	HttpUtility 类	659
			参考文献		662

第1章 .NET框架及类库

1.1 使用 C#语言的原因

C#是一种精确、简单、类型安全、面向对象的语言，它使企业程序员得以构建广泛的应用程序。C#还凭借以下功能，提供了生成持久系统级组件的能力：

- 对集成现有代码提供完全 COM/平台支持。
- 通过提供垃圾回收和类型安全实现可靠性。
- 通过提供内部代码信任机制保证安全性。
- 完全支持可扩展元数据概念。

C#还可以凭借以下功能，与其他语言交互操作、跨平台互用并与遗留的数据交互操作：

- 通过 COM+ 1.0 和 .NET 框架服务提供具有紧密库访问的完全相互作用支持。
- 对基于 Web 的组件交互提供 XML 支持。
- 版本转换功能使管理和部署变得简易。

.NET 平台中的 C#

图 1-1 就是著名的 .NET 平台结构图，从这个图上可以看到，ASP.NET，Windows Forms 和 VS.NET 都是 .NET 开发平台的一部分，用于 .NET 应用程序的开发及展示。.NET 平台的核心技术为：通用语言运行时 CLR (Common Language Runtime)，基类库 (Base Class Library)，.NET 语言及 Visual Studio.NET。

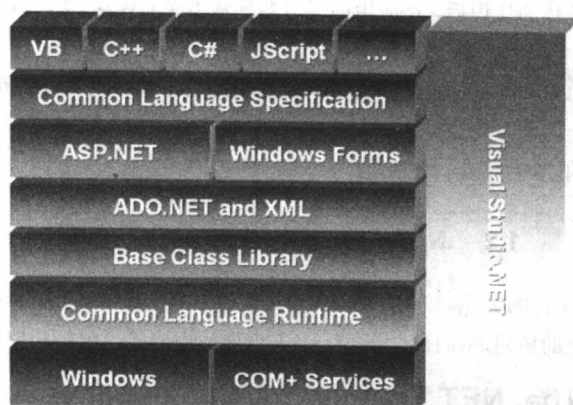


图 1-1 .NET 平台结构图

从图 1-1 可以看出，.NET 架构在 Windows 平台上是一个虚拟的运行平台，若将最下层的 Windows 换作其他的操作系统，比如 Linux，同样可以实现使用符合了 CLS (Common Language Specification，通用语言规范) 的 .NET 语言 (VB.NET、C#、JScript.NET 等) 来创建 ASP.NET 或 Windows Forms (可能会叫做 Linux Forms) 应用程序的功能，这就是 Mono 计划所要实现的功能。理论上，C# 是一种可以跨平台的语言，这很像 Java，另一个比较像 Java 的地方是，C# 也是一种 (特殊意义上的) 解释性的语言。同 Java 一样，C# 编写的程序代码也是先通过 C# 编译器编译为一种特殊的字节代码 (Microsoft Intermediate Language，中间语言)，运行的时候再经由特定的编译器 JIT (Just In Time，编译器) 编译为机器代码以供操作系统执行。

不仅是 C# 语言，所有 .NET 语言 (包括常用几十种现代编程语言) 都可以编写面向 CLR 的程序代码，这种代码在 .NET 中被称为托管代码 (Managed Code)，所有的 Managed Code 都直接运行在 CLR 上，具有与平台无关的特性。

为什么 C# 语言具有那么多好的特性

解释性的语言很安全，并且可以通过它的运行平台为其赋予更多的功能，比如自动内存管理、异常处理等。事实上，C# 语言的许多特点都是由 CLR 提供的，图 1-2 的 CLR 结构图说明了这一点。

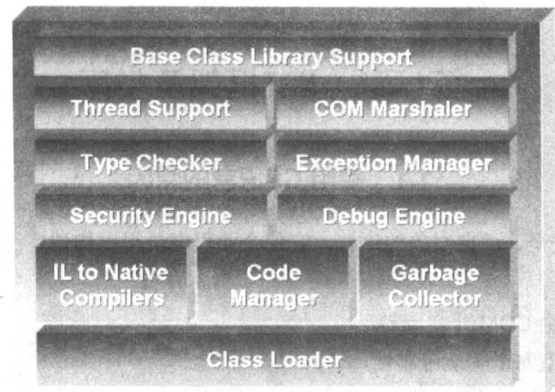


图 1-2 CLR 结构图

可以看到，类型安全 (Type Checker)、垃圾回收 (Garbage Collector)、异常处理 (Exception Manager)、向下兼容 (COM Marshaler)、多线程支持 (Thread Support) 等 C# 的特点都是由 CLR 提供的。CLR 最早被称为下一代 Windows 服务运行时 (NGWS Runtime)，是直接建立在操作系统层上的一个虚拟的运行环境，主要的功能是管理代码的运行。在 .NET 平台结构图中，CLR 的上面是 .NET 的基类库 BCL (Base Class Library)，这组基类库包括了从基本输入/输出到数据访问等方面，提供了一个统一的面向对象的、层次化的、可扩展的编程接口。从 .NET 平台结构图中也可以看到，基类库可以被各种语言调用和扩展，也就是说，不管是 C#、VB.NET 还是 VC++.NET，都可以自由地调用 .NET 的基类库。事实上，C# 并没有属于自己的类库，它所使用的编程接口就是 .NET 提供的基类库。在决定使用 C# 时，真正需要费工夫学习的其实是 .NET 框架的基类库。C# 自身只有 77 个关键词，而且其语法对许多程序员来说都是非常熟悉的。BCL 则相反，它包含了超过 4500 个以上的类和无数的方法、属性，在你的 C# 程序中随时都可能用到它来完成自己的任务。

对运行平台和语法有了一个整体的认识后，在应用中学习各种基类库的用法。鉴于 C# 语言的特殊性，全面了解它的运行平台必会使你的学习事半功倍。

任何一个想要学习 C# 语言的人，在面对如此巨大的 .NET Framework 类库时，都会有一种无处着手，就如同进入了一个到处是奇珍异宝的森林，必然会迷失方向。

本章的目的是为你提供关于 .NET 类库的一个顶级路标，从而使你不致于迷失方向。

1.2 .NET Framework 中的命名空间

下面从整体上了解一下 .NET 平台的类库。 .NET 框架类库是一个由 Microsoft.NET 框架 SDK 中包含的类、接口和值类型组成的库。该库提供对系统功能的访问，并且被设计为 .NET 框架应用程序、组件和控件的生成基础，如图 1-3 所示。

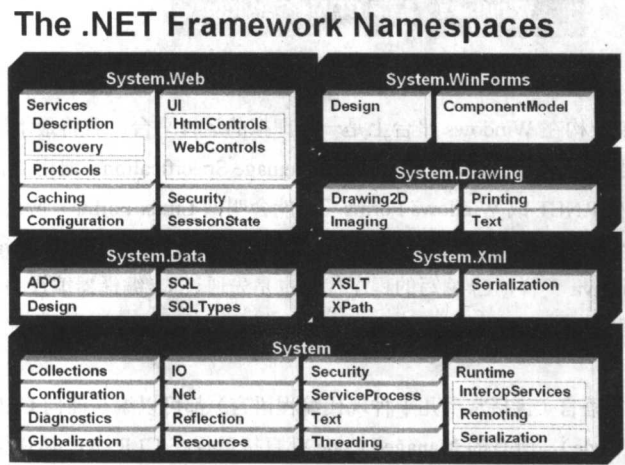


图 1-3 .NET 框架的命名空间

.NET 框架的功能

.NET 框架类型是生成.NET 应用程序、组件和控件的基础。.NET 框架包括的类型执行下列功能:

- 表示基础数据类型和异常。
- 封装数据结构。
- 执行 I/O。
- 访问关于加载类型的信息。
- 调用.NET 框架安全检查。
- 提供数据访问、多客户端 GUI 和服务器控制的客户端 GUI。

.NET 框架提供一组丰富的接口、抽象类和具体(非抽象)类。可以按原样使用这些具体的类,或者在多数情况下从这些类派生自己的类。若要使用接口的功能,既可以创建实现接口的类,也可以从某个实现接口的.NET 框架类中派生类。

命名约定

.NET 框架类型使用点语法命名方案,该方案隐含了层次结构的意思。此技术将相关类型分为不同的命名空间组,以便更容易地搜索和引用它们。全名的第一部分(最右边的点之前的内容)是命名空间名,全名的最后一部分是类型名。例如 `System.Collections.ArrayList` 表示 `ArrayList` 类型,该类型属于 `System.Collections` 命名空间。`System.Collections` 中的类型可用于操作对象集合。

此命名方案使扩展.NET 框架的库开发人员可以轻松地创建分层类型组,并用一致的、带有提示性的方式对其进行命名。库开发人员在创建命名空间的名称时应使用以下原则:

公司名称.技术名称

例如, `Microsoft.Word` 命名空间就符合此原则。

利用命名模式将相关类型分组为命名空间是生成和记录类库的一种非常有用的方式,但是此命名方案对可见性、成员访问、继承、安全性或绑定无效。一个命名空间可以被划分在多个程序集中,而单个程序集可以包含来自多个命名空间的类型。程序集为公共语言运行库中的版本控制、部署、安全性、加载和可见性提供外形结构。

1.3 与 Windows 系统整合

.NET 框架类库针对.NET SDK 随附了 3 种编译器,提供了 3 个不同的命名空间。

- `Microsoft.CSharp` 包含支持用 C# 语言进行编译和代码生成的类。其中包含了代表 C# 编译器提供的托管包装器的类,编译器生成的一条错误或诊断消息以及能够访问 C# 代码生成器和编译器的实例的类。
- `Microsoft.JScript` 包含支持用 JScript 语言进行编译和代码生成的 JScript 运行库和类。
- `Microsoft.VisualBasic` 包含 `Visual Basic.NET` 运行库,此运行库与 `Visual Basic.NET` 语言一起使用。此命名空间还包含支持用 `Visual Basic.NET` 语言进行编译和代码生成的类。

以下的命名空间包含了用来创建或使用编译器的类:

`System.CodeDom`

`System.CodeDom.Compiler`

`System.Reflection.Emit`

`System.Runtime.CompilerServices`

.NET 框架类库提供了一个命名空间用来与 Windows 操作系统一起工作,那就是 `Microsoft.Win32` 命名空间。

`Microsoft.Win32` 命名空间提供两种类型的类:处理由操作系统引发的事件的类和操作系统注册表的类。此外还包括处理操作系统生成事件的委托,定义了一些处理各种操作系统级操作的枚举。

表 1-1 Microsoft.Win32 命名空间定义的类

类	说明
<code>PowerModeChangedEventArgs</code>	为 <code>PowerModeChanged</code> 事件提供数据
<code>Registry</code>	提供访问注册表中的值和子项的基 <code>RegistryKeys</code>
<code>RegistryKey</code>	表示 Windows 注册表中的项级节点。此类是注册表封装

(续表)

类	说明
SessionEndedEventArgs	为 SessionEnded 事件提供数据
SessionEndingEventArgs	为 SessionEnding 事件提供数据
SystemEvents	为调用方提供一组全局系统事件。无法继承此类
TimerElapsedEventArgs	为 TimerElapsed 事件提供数据
UserPreferenceChangedEventArgs	为 UserPreferenceChanged 事件提供数据
UserPreferenceChangingEventArgs	为 UserPreferenceChanging 事件提供数据

1.4 .NET 核心类

整个.NET的所有类都是基于 System.Object 类的。下面把所有的核心类 System 的层次结构展示一下：

```

System.Object
  System.Activator
  System.AppDomainSetup ---- System.IAppDomainSetup
  System.Array ---- System.ICloneable, System.Collections.IList, System.Collections.ICollection,
System.Collections.IEnumerable
  System.Attribute
    System.AttributeUsageAttribute
    System.CLSCompliantAttribute
    System.ContextStaticAttribute
    System.FlagsAttribute
    System.LoaderOptimizationAttribute
    System.MTAThreadAttribute
    System.NonSerializedAttribute
    System.ObsoleteAttribute
    System.ParamArrayAttribute
    System.SerializableAttribute
    System.STAThreadAttribute
    System.ThreadStaticAttribute
  System.BitConverter
  System.Buffer
  System.CharEnumerator ---- System.Collections.IEnumerator, System.ICloneable
  System.Console
  System.Convert
  System.DBNull ---- System.Runtime.Serialization.ISerializable, System.IConvertible
  System.Delegate
    System.MulticastDelegate
      System.AssemblyLoadEventHandler
      System.AsyncCallback
      System.CrossAppDomainDelegate
      System.EventHandler
      System.ResolveEventHandler
      System.UnhandledExceptionHandler
  System.Environment
  System.EventArgs

```

```
System.AssemblyLoadEventArgs
System.ResolveEventArgs
System.UnhandledExceptionEventArgs
System.Exception
System.ApplicationException
System.SystemException
    System.AppDomainUnloadedException
    System.ArgumentException
        System.ArgumentNullException
        System.ArgumentOutOfRangeException
        System.DuplicateWaitObjectException
    System.ArithmeticException
        System.DivideByZeroException
        System.NotFiniteNumberException
        System.OverflowException
    System.ArrayTypeMismatchException
    System.BadImageFormatException
    System.CannotUnloadAppDomainException
    System.ContextMarshalException
    System.ExecutionEngineException
    System.FormatException
        System.UriFormatException
    System.IndexOutOfRangeException
    System.InvalidCastException
    System.InvalidOperationException
        System.ObjectDisposedException
    System.InvalidProgramException
    System.MemberAccessException
        System.FieldAccessException
        System.MethodAccessException
        System.MissingMemberException
            System.MissingFieldException
            System.MissingMethodException
    System.MulticastNotSupportedException
    System.NotImplementedException
    System.NotSupportedException
        System.PlatformNotSupportedException
    System.NullReferenceException
    System.OutOfMemoryException
    System.RankException
    System.StackOverflowException
    System.TypeInitializationException
    System.TypeLoadException
        System.DllNotFoundException
        System.EntryPointNotFoundException
```

```

    System.TypeUnloadedException
    System.UnauthorizedAccessException
System.GC
System.LocalDataStoreSlot
System.MarshalByRefObject
    System.AppDomain ---- System._AppDomain, System.Security.IEvidenceFactory
    System.ContextBoundObject
    System.Uri ---- System.Runtime.Serialization.ISerializable
System.Math
System.OperatingSystem ---- System.ICloneable
System.Random
System.Reflection.MemberInfo
    System.Type ---- System.Reflection.IReflect
System.String ---- System.IComparable, System.ICloneable, System.IConvertible,
System.Collections.IEnumerable
System.TimeZone
System.UriBuilder
System.ValueType
    System.ArgIterator
    System.Boolean ---- System.IComparable, System.IConvertible
    System.Byte ---- System.IComparable, System.IFormattable, System.IConvertible
    System.Char ---- System.IComparable, System.IConvertible
    System.DateTime ---- System.IComparable, System.IFormattable, System.IConvertible
    System.Decimal ---- System.IFormattable, System.IComparable, System.IConvertible
    System.Double ---- System.IComparable, System.IFormattable, System.IConvertible
    System.Enum
        System.AttributeTargets
        System.DayOfWeek
        System.Environment.SpecialFolder
        System.LoaderOptimization
        System.PlatformID
        System.TypeCode
        System.UriHostNameType
        System.UriPartial
    System.Guid ---- System.IFormattable, System.IComparable
    System.Int16 ---- System.IComparable, System.IFormattable, System.IConvertible
    System.Int32 ---- System.IComparable, System.IFormattable, System.IConvertible
    System.Int64 ---- System.IComparable, System.IFormattable, System.IConvertible
    System.IntPtr ---- System.Runtime.Serialization.ISerializable
    System.RuntimeArgumentHandle
    System.RuntimeFieldHandle ---- System.Runtime.Serialization.ISerializable
    System.RuntimeMethodHandle ---- System.Runtime.Serialization.ISerializable
    System.RuntimeTypeHandle ---- System.Runtime.Serialization.ISerializable
    System.SByte ---- System.IComparable, System.IFormattable, System.IConvertible
    System.Single ---- System.IComparable, System.IFormattable, System.IConvertible

```

```

System.TimeSpan ---- System.IComparable
System.TypedReference
System.UInt16 ---- System.IComparable, System.IFormattable, System.IConvertible
System.UInt32 ---- System.IComparable, System.IFormattable, System.IConvertible
System.UInt64 ---- System.IComparable, System.IFormattable, System.IConvertible
System.UIntPtr ---- System.Runtime.Serialization.ISerializable
System.Void

System.Version ---- System.ICloneable, System.IComparable
System.WeakReference ---- System.Runtime.Serialization.ISerializable

```

System 命名空间包含基本类和基类，这些类定义常用的值和引用数据类型、事件和事件处理程序、接口、属性和异常处理。其他类提供的服务支持数据类型转换、方法参数操作、数学运算、远程和本地程序调用、应用程序环境管理和对托管与非托管应用程序的监控。

第2章 Microsoft.Win32命名空间

Microsoft.Win32 命名空间提供两种类型的类：处理由操作系统引发的事件的类和操作系统注册表的类。System.Text 命名空间包含的类及说明见表 2-1。

表 2-1 Microsoft.Win32 命名空间包含类

类	说明
PowerModeChangedEventArgs	为 PowerModeChanged 事件提供数据
Registry	提供访问注册表中的值和子项的基 RegistryKeys
RegistryKey	表示 Windows 注册表中的顶级节点，此类是注册表封装
SessionEndedEventArgs	为 SessionEnded 事件提供数据
SessionEndingEventArgs	为 SessionEnding 事件提供数据
SystemEvents	为调用方提供一组全局系统事件，无法继承此类
TimerElapsedEventArgs	为 TimerElapsed 事件提供数据
UserPreferenceChangedEventArgs	为 UserPreferenceChanged 事件提供数据
UserPreferenceChangingEventArgs	为 UserPreferenceChanging 事件提供数据

2.1 PowerModeChangedEventArgs 类

继承关系

System.Object

System.EventArgs

Microsoft.Win32.PowerModeChangedEventArgs

说明：为 PowerModeChanged 事件提供数据。当用户挂起或继续操作系统时引发 PowerModeChanged 事件。

函数声明

```
public PowerModeChangedEventArgs(PowerModes mode);
```

参数 mode 表示新的电源模式的 PowerModes 值之一。

公共属性 Mode 获取电源模式。

表 2-2 PowerModeChangedEventArgs 类的公共方法

方法	说明
Equals	(从 Object 继承) 已重载。确定两个 Object 实例是否相等
GetHashCode	(从 Object 继承) 用作特定类型的哈希函数，适合在哈希算法和数据结构中使用
GetType	(从 Object 继承) 获取当前实例的 Type
ToString	(从 Object 继承) 返回表示当前 Object 的 String

2.2 Registry 类

继承关系

System.Object

Microsoft.Win32.Registry

说明：提供访问注册表中的值和子项的基 RegistryKeys。此类提供在运行 Windows 的计算机上的注册表中找到标准根项集。注册表是一个存储设备，包含有关应用程序、用户和默认系统设置的信息。由 Registry 公开的基（根）RegistryKey 实例描绘注册表中的子项和值的基本存储机制。所有项都是只读的，因为注册表依赖于它们的存在。一旦标识了希望在其下存储/检索注册表中信息的根项后，可以使用 RegistryKey 类添加或删除子项和操作给定项的值。硬件设备可以使用即插

即用接口将信息自动放在注册表中。安装设备驱动程序的软件可以通过向标准 API 写入将信息放在注册表中。

表 2-3 Registry 类的公共字段

字段名称	说明
ClassesRoo	定义文档的类型（或类）以及与那些类型关联的属性，该字段读取注册表基项 HKEY_CLASSES_ROOT
CurrentConfig	包含有关非用户特定的硬件的配置信息，该字段读取注册表基项 HKEY_CURRENT_CONFIG
CurrentUser	包含有关当前用户首选项的信息，该字段读取注册表基项 HKEY_CURRENT_USER
DynData	包含动态注册表数据，该字段读取 Windows 注册表基项 HKEY_DYN_DATA
LocalMachine	包含本地计算机的配置数据该字段读取注册表基项 HKEY_LOCAL_MACHINE
PerformanceData	包含软件组件的性能信息，该字段读取注册表基项 HKEY_PERFORMANCE_DATA
Users	包含有关默认用户配置的信息，该字段读取注册表基项 HKEY_USERS

示例：说明如何利用 Registry 类检索注册表中 HKEY_CLASSES_ROOT 项的子项，以及如何将它们名称输出到屏幕。使用 OpenSubKey 方法创建相关的特定子项的实例，然后可使用 RegistryKey 中的其他方向操作该项。

```
public static void Main() {
    RegistryKey rk = Registry.ClassesRoot;
    PrintKeys(rk);
}

static void PrintKeys(RegistryKey rkey) {
    //调用 RegistryKey 类的方法 GetSubKeyNames 获取注册表中各项的名称
    String [] names = rkey.GetSubKeyNames();
    int icount = 0;
    Console.WriteLine("Subkeys of " + rkey.Name);
    Console.WriteLine("-----");
    foreach (String s in names) {
        Console.WriteLine(s);
        icount++;
        if (icount >= 10) break;
    }
}
```

程序的执行结果如图 2-1 所示。

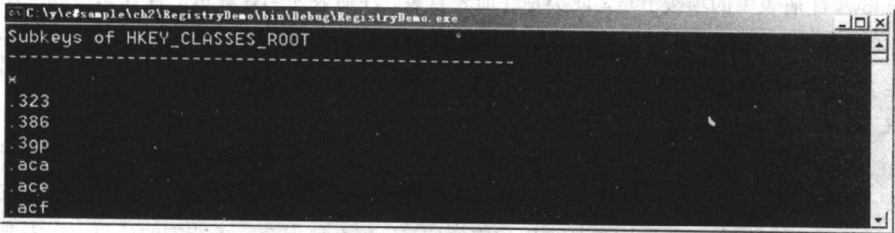


图 2-1 程序的执行结果

2.3 SystemEvents 类

继承关系

System.Object

Microsoft.Win32.SystemEvents

说明：为调用方提供一组全局系统事件，无法继承此类，系统事件是指诸如关闭通知和对用户首选项的更改这样的事件。当引发系统事件时，将从另一个线程回调相应的委托，因此应确保调用是线程安全的。如果需要调用未公开为此类成员的系统事件，可以使用 InvokeOnEventsThread 方法，但不要在系统事件通知中执行需耗费大量时间的处理，因为这可能

会妨碍其他应用程序的工作。

公共方法 `InvokeOnEventsThread` 表示在侦听系统事件的线程上执行给定的委托。

表 2-4 `SystemEvents` 类的公共事件

事件名称	说明
<code>DisplaySettingsChanged</code>	当用户更改显示设置时发生
<code>EventsThreadShutdown</code>	在侦听系统事件的线程终止前发生。将在事件线程上调用委托
<code>InstalledFontsChanged</code>	当用户在系统中添加或删除字体时发生
<code>LowMemory</code>	当系统用完可用 RAM 时发生
<code>PaletteChanged</code>	当用户切换到使用其他调色板的应用程序时发生
<code>PowerModeChanged</code>	当用户挂起或继续系统时发生
<code>SessionEnded</code>	当用户注销或关闭系统时发生
<code>SessionEnding</code>	当用户试图注销或关闭系统时发生
<code>TimeChanged</code>	当用户更改系统时钟上的时间时发生
<code>TimerElapsed</code>	在窗口计时器间隔过期时发生
<code>UserPreferenceChanged</code>	在用户首选项更改后发生
<code>UserPreferenceChanging</code>	当用户首选项更改时发生

`InvokeOnEventsThread` 方法

在侦听系统事件的线程上执行给定的委托，当需要处理未通过 `SystemEvents` 类公开的系统事件时，都可以使用此方法。当调用此方法时，应用程序正在使用的线程将调用委托以处理系统事件。

函数声明

```
public static void InvokeOnEventsThread( Delegate method );
```

参数 `method` 侦听系统事件的处理程序。

2.4 `RegistryKey` 类

继承关系

`System.Object`

`System.MarshalByRefObject`

`Microsoft.Win32.RegistryKey`

说明：表示 Windows 注册表中的项级节点，此类是注册表封装。注册表充当计算机上操作系统和应用程序的中央信息储存库。注册表根据存储在其中的元素的逻辑顺序，以分层形式组织。在注册表中存储信息时，请根据存储的信息类型选择适当的位置。一定要避免损坏由其他应用程序创建的信息，原因是这样会导致那些应用程序出现意外的行为，并且还会对你的应用程序带来不利影响。

`RegistryKeys` 是注册表中的基本组织单位，好比 Windows 资源管理器中的文件夹。特定的项可以有子项（与文件夹可以有子文件夹一样），只要用户具有适当的删除权限，且此项不是基项或基项的下一级项，就可以删除。每个项还可以有多个关联的值（值好比是文件），用于存储你感兴趣的应用程序的信息。每个值保存一条特定的信息，在需要时可以检索或更新此信息。存储在注册表中的信息可由其他应用程序和用户使用，因此不要用注册表存储安全信息或关键的应用程序信息。若要获取 `RegistryKey` 的实例，请使用静态成员 `OpenSubKey` 或者使用 `Registry` 类的静态成员。

表 2-5 `RegistryKey` 类的公共属性

属性	说明
<code>Name</code>	检索项的名称
<code>SubKeyCount</code>	在基级检索当前项的子项计数
<code>ValueCount</code>	检索项中值的计数