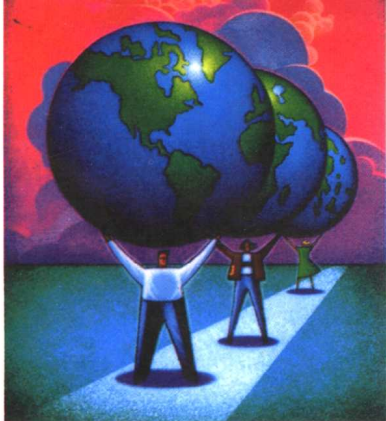




PH
PTR

C# Web 编程

- ▶ 构造在浏览器环境中运行的应用程序。
- ▶ 掌握代码开发、Visual Studio .NET 设计环境、控件、事件、属性等。
- ▶ 完整的操作代码例子将帮助你研究概念和掌握 C# 的 Web 开发功能。

〔美〕 Chris H. Pappas 著
William H. Murray
袁鹏飞 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

TP312
902

C# Web 编程

[美] Chris H. Pappas William H. Murray 著

袁鹏飞 译

人民邮电出版社

图书在版编目(CIP)数据

C# Web 编程 / (美) 帕帕斯 (Pappas, C.H.), (美) 莫里 (Murray, M.H.) 著;
袁鹏飞译. —北京: 人民邮电出版社, 2002.10

ISBN 7-115-10564-2

I. C... II. ①帕...②莫...③袁... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 061700 号

版权声明

Simplified Chinese Copyright © 2002 by PEARSON EDUCATION NORTH ASIA LIMITED and POSTS & TELECOMMUNICATIONS PRESS.

C# Web Programming

By Chris H. Pappas William H. Murray

Copyright © 2002

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice-Hall PTR

This edition is authorized for sale only in People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书封面贴有 Pearson Education 出版集团激光防伪标签, 无标签者不得销售。

C# Web 编程

- ◆ 著 [美] Chris H.Pappas William H.Murray
- 译 袁鹏飞
- 责任编辑 陈 昇

- ◆ 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>
读者热线 010-67132705
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

- ◆ 开本: 787×1092 1/16
印张: 27.25
字数: 660 千字 2002 年 10 月第 1 版
印数: 1-4 000 册 2002 年 10 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2002 - 3729 号

ISBN 7-115-10564-2/TP · 3044

定价: 48.00 元

本书如有印装质量问题, 请与本社联系 电话: (010)67129223

内 容 提 要

C#是 Microsoft 针对 .NET 框架而专门开发的一种面向对象程序设计语言，它继承了 C/C++ 的强大功能，但舍弃了 C/C++ 中很多复杂、容易出现编程错误的特性和功能，并融合了 Visual Basic 的易用性。

本书以循序渐进的方式介绍如何使用 C# 进行 Web 程序设计，并提供了丰富的实例和完整的实例代码。全书共分为 12 章，分别介绍 C# 的发展历史、语法要点、C# 和 Visual Studio.NET、控件属性、事件、Web 表单输入/输出处理、GDI 图形设计、图形绘制、数值应用和转换、Web 服务等。

本书适合于初中级程序开发人员阅读。通过阅读本书，读者可以快速、全面地掌握 C# Web 程序设计技术。

序

本书是为具有 C/C++ 知识的 Web 程序员而设计的，如果你用任何其他一种语言开发过 Web 应用，并且想了解怎样在 C# 中实现它，那么这本书适合你！

本书的目的是向你介绍用 C# 语言开发 Web 应用时所使用的该语言的关键元素，本书首先快速介绍 C# 的主要功能，之后把你带入 Web 应用开发领域。你将学习到 Web 开发的程序设计策略，这将使用 C# 编写代码更加容易。

本书通过例子进行讲解，我们尽力在本书中包含多数 Web 应用中常见的所有重要功能。例子包含程序代码的所有内容，而不只是代码的一部分。对于有经验的程序员来说，这似乎过于重复，但是经验告诉我们，一个完整的程序清单有助于解释最困难的程序设计概念。

C# 似乎从用 MFC 进行程序设计的传统 C++（Windows 程序员对此非常熟悉）来了一个 180 度的大转弯。虽然该语言肯定与 C 类似，但是，接口肯定不是传统的 SDK。相反，它与 Visual Basic 语言更加紧密相关，因为它使用表单、控件等。你肯定会喜欢这种环境！

本书将重点介绍 C# Web 应用开发与其他语言的根本区别。程序员在 C# 中仍然使用 C/C++ 元素，但是，他们使用一种更简单的拖放方式设计，单是这一变化就可能使 Web 应用设计发生巨大的变化。

作者

目 录

第 1 章 C#发展历史	1
1.1 所有都始自 Algol.....	1
1.1.1 为什么在 C#中看到 C	2
1.1.2 C 与旧的高级语言	3
1.1.3 C 的优点	4
1.2 从 C 到 C++ 和面向对象的程序设计	5
1.3 Visual Basic	6
1.4 进入 Internet	6
1.5 HTML 家族	7
1.5.1 CGI.....	7
1.5.2 PERL	8
1.5.3 JavaScript 和 JScript	8
1.5.4 VBScript.....	8
1.5.5 插件和 ActiveX	8
1.5.6 允许所有类型的计算机访问	9
1.5.7 Web 浏览器	9
1.5.8 多彩浏览, 唾手可得	9
1.5.9 信息, 到处都是信息	9
1.5.10 全双向通信	10
1.6 C#——另一种金字塔方案	10
1.6.1 ANSI C#.....	10
1.6.2 什么是 MSIL	10
1.6.3 Microsoft 和 .NET	11
1.6.4 C#和 .NET	11
1.6.5 通用语言规范——CLS	12
1.6.6 CLS 对 C++ 的扩展	12
1.6.7 互操作能力的重要性	12
1.7 C#引言与综述	12

1.7.1	C#——广泛的应用领域	13
1.7.2	C#的效率	13
1.7.3	C#和新的 Web 标准	13
1.7.4	C#使你成为更好的程序员	14
1.7.5	C#延长应用程序的生命期	14
1.7.6	从设计到实现的精确转换	14
1.7.7	广泛的互操作性	14
1.8	小 结	15
第 2 章	C#语法要点	17
2.1	Visual Studio.NET 和 C#	17
2.2	独特的 C#	18
2.3	概 述	18
2.3.1	容易使用	18
2.3.2	一致性	20
2.3.3	最新的资源	20
2.3.4	面向对象功能	21
2.3.5	类型安全	22
2.3.6	可伸缩性	22
2.3.7	版本控制	23
2.3.8	可移植性	23
2.3.9	灵活性	23
2.4	总体介绍	24
2.4.1	可见性	24
2.4.2	访问成员	28
2.4.3	拆分	31
2.4.4	名称解析	32
2.5	重要的数据、标识符和关键字	33
2.5.1	关键字	34
2.5.2	预处理伪指令	35
2.5.3	运算符	39
2.5.4	预定义类型	45
2.5.5	数值转换	51
2.6	C#程序设计元素	53
2.6.1	数组	53
2.6.2	属性、事件、指示器、特性和版本控制	55

2.6.3 装箱、拆箱和统一类型系统	56
2.6.4 类、结构和枚举	57
2.6.5 名域	58
2.6.6 语句	59
2.7 小 结	62
第 3 章 C#和 Visual Studio.NET	65
3.1 为什么要在 Web 工程中使用 C#	65
3.2 C#语言和 Web 应用	65
3.3 开始使用 Visual Studio.NET	65
3.3.1 设置一个新工程	66
3.3.2 为什么选择 ASP.NET Web Application 模板	68
3.3.3 ASP.NET Web Application 模板需求	68
3.3.4 什么时候选择 Web Application 而不是 Web Service	69
3.3.5 GridLayout 与 FlowLayout	69
3.3.6 Solution Explorer	73
3.3.7 Visual Studio.NET 解决方案文件	73
3.3.8 使用动态帮助	74
3.3.9 编辑窗口显示选项	75
3.4 基于事件的程序设计	77
3.5 C#使用的 Visual Studio 工具	77
3.6 标准控件	78
3.7 控件属性	79
3.7.1 修改控件默认属性	80
3.7.2 一种修改几种控件属性的快速方法	82
3.7.3 对象名称和标记	82
3.7.4 事件处理程序	84
3.7.5 用代码修改属性	84
3.8 创建友好的用户界面	86
3.8.1 良好设计的因素	86
3.8.2 使用表格	86
3.8.3 控件基础	87
3.9 添加代码	97
3.10 执行程序	98
3.11 小 结	100

第 4 章 静态控件属性	101
4.1 Properties 窗口详细介绍	101
4.2 属性详细介绍	102
4.3 修改属性	103
4.3.1 Button 属性	103
4.3.2 CheckBox 控件属性	104
4.3.3 Calendar 属性	106
4.3.4 表单颜色和图像属性	108
4.3.5 Panel 控件属性	110
4.3.6 Label 控件属性	112
4.3.7 ListBox 控件属性	112
4.3.8 Image 控件属性	115
4.3.9 HyperLink 控件	115
4.3.10 LinkButton 控件	116
4.3.11 RadioButtonList 控件属性	116
4.3.12 TextBox 控件属性	117
4.3.13 DataGrid 控件	119
4.3.14 ToolTip 属性	123
4.4 小 结	124
第 5 章 动态控件属性	125
5.1 编写控件代码	125
5.2 属性加代码	126
5.3 事件处理程序	126
5.4 动态修改控件属性	127
5.4.1 Button 属性	148
5.4.2 CheckBox 属性	149
5.4.3 Label 属性	153
5.4.4 Image 属性	155
5.4.5 LinkButton 和 HyperLink 属性	156
5.4.6 DropDownList 属性	159
5.4.7 ListBox 属性	163
5.4.8 Panel 属性	166
5.4.9 RadioButton 属性	169
5.5 Tip Calculator 的另一种设计方法	172

5.5.1 用正确的控件完成正确的工作	173
5.5.2 工程代码	174
5.6 小 结	176
第 6 章 事 件	177
6.1 事件处理程序概述	177
6.2 事 件	178
6.2.1 AdCreated 事件	184
6.2.2 CancelCommand 事件	185
6.2.3 ChartDrillDown 事件	186
6.2.4 CheckedChanged	186
6.2.5 Click 事件	187
6.2.6 Command 事件	188
6.2.7 DayRender 事件	189
6.2.8 DeleteCommand 事件	189
6.2.9 EditCommand 事件	190
6.2.10 GroupTreeDrillDown 事件	190
6.2.11 GroupTreeGrow 事件	191
6.2.12 GroupTreeNavigate 事件	191
6.2.13 ItemCommand 事件	192
6.2.14 LevelUp 事件	192
6.2.15 PageDrillDown 事件	192
6.2.16 PageIndexChanged 事件	193
6.2.17 PageNavigate 事件	193
6.2.18 Refresh 事件	194
6.2.19 Search 事件	194
6.2.20 SelectedIndexChanged 事件	194
6.2.21 SelectionChanged 事件	196
6.2.22 SortCommand 事件	196
6.2.23 SubreportDrillDown 事件	197
6.2.24 TextChanged 事件	197
6.2.25 UpdateCommand 事件	198
6.2.26 VisibleMonthChanged 事件	199
6.2.27 Zoom 事件	199
6.3 事件间的细微差别	200
6.3.1 鼠标和键盘事件	200

6.3.2 继承事件和定制事件	200
6.3.3 冒泡	201
6.4 小 结	201
第 7 章 处理 Web 表单输入	203
7.1 用于输入的 Web 控件	203
7.1.1 使用 TextBar 控件	204
7.1.2 使用 RadioButton 和 RadioButtonList 控件	208
7.1.3 使用 CheckBox 和 CheckBoxList 控件	217
7.1.4 使用 Calendar 控件	226
7.1.5 使用 Validation 控件	229
7.1.6 使用 Button 和 ImageButton 控件	234
7.1.7 使用 ListBox 和 DropDownList 控件	238
7.2 HTML 服务器控件	242
7.3 小 结	247
第 8 章 处理 Web 表单输出	249
8.1 输 出	249
8.1.1 使用 TextBox 或 Label 控件	249
8.1.2 使用多行 TextBox 控件	253
8.1.3 使用 TextBox 控件的表格输出	257
8.1.4 输出到打印机	262
8.1.5 使用 AdRotator 控件	270
8.2 以列表形式输出	275
8.2.1 使用 Table 控件	276
8.2.2 使用 DataGrid 控件	279
8.2.3 使用 DataList 控件	283
8.2.4 使用 Repeater 控件	287
8.3 小 结	292
第 9 章 GDI 图形设计基础	293
9.1 System.Drawing 名域	293
9.2 System.Drawing.Drawing2D 名域	296
9.3 C#的图形类	298
9.4 坐标系统	302
9.5 绘图界面	306

9.6 图形属性	307
9.6.1 颜色	307
9.6.2 直线绘图样式	309
9.6.3 画刷填充样式	310
9.6.4 DrawAndFill 工程	310
9.7 绘图基本元素	315
9.7.1 DrawArc	315
9.7.2 DrawEllipse	315
9.7.3 DrawLine	316
9.7.4 DrawPie	316
9.7.5 DrawPolygon	317
9.7.6 DrawRectangle	317
9.7.7 DrawString	318
9.7.8 FillEllipse	318
9.7.9 FillPie	319
9.7.10 FillPolygon	319
9.7.11 FillRectangle	320
9.7.12 DrawingPrimitives 工程	320
9.8 具有 GDI+ 输出的 Web 工程	325
9.9 小 结	332
第 10 章 绘制图形	333
10.1 BarChart 工程	333
10.1.1 BarChart 工程代码	334
10.1.2 绘制独特的直方图	345
10.2 小 结	346
第 11 章 数值应用和转换	347
11.1 基数转换计算器	347
11.2 素数计数器	352
11.3 三角表	356
11.4 确定一月中的天数	364
11.5 日 历	368
11.6 统 计	372
11.7 排 序	377
11.8 资产折旧	382

11.8.1 把代码附加到表单	382
11.8.2 资产折旧和数据共享	387
11.9 还 贷	388
11.9.1 开发还贷代码	388
11.9.2 查看贷款偿还选项	391
11.10 小 结	392
第 12 章 Web 服务	393
12.1 方法、库和 Web 服务	394
12.2 “HelloWorld”Web 服务	394
12.2.1 创建 HW1 Web 服务工程	394
12.2.2 设计 HW1 Web 服务	395
12.2.3 测试 HW1 Web 服务代码	401
12.2.4 分发 HW1 Web 服务	403
12.2.5 创建测试 HW1 的客户端应用	406
12.2.6 分发和测试客户端应用	409
12.3 一个转换 Web 服务	413
12.3.1 创建 DegToRad1 Web 服务工程	413
12.3.2 设计 DegToRad1 Web 服务	414
12.3.3 测试 DegToRad1 Web 服务代码	415
12.3.4 分发 DegToRad1 Web 服务	417
12.3.5 创建测试 DegToRad1 的客户端应用	417
12.3.6 分发和测试客户端应用	421
12.4 小 结	422

第 1 章 C#发展历史

“各就位，预备，跑！”这一运动场上的口令很好地描述了理解和编写 C# Web 应用是多么地快捷而简单。无论你是刚接触程序设计还是一位经验丰富的老手，Microsoft 新的、功能强大的、有趣的程序设计语言 C#将使你很快成为其信徒。

有趣？是的！C#提供 GUI(Graphical User Interface)设计和布局方面的易用性，Visual Basic 程序员多年来已经使用这一功能。C#组件允许通过简单地在工具栏上点击、把一个控件拖到设计页面并从直观的下拉列表中设置该控件属性来创建应用程序或小应用程序（applet）的用户界面。

随着今天越来越多的程序要求以 Web 方式显示，C#会稳居开发语言之首。由于 C#结合了 Java 或 J++、CGI、PERL、C/C++ 和 Visual Basic 与 Java 字节码(bytecode)——即本地代码格式——结构独立性的优点，所以其代码解决方案具有在演化和发展过程中完全不需要重新编写的优点。

基于 Web 的解决方案不再要求 Visual Basic 程序员做界面设计，不再要求 C++程序员做纯粹的、原始的数据提取功能，不再要求 Java、CGI 或 PERL 专家使整个程序包在世界范围内可以使用。

为了弄清曾经熟悉的 C#语法怎样成为今天最新的程序设计语言，可以先快速浏览一下第 2 章。但是，还是请马上回到本章，后面的内容将简要介绍产生 C#程序设计开发语言的历史。

1.1 所有都始自 Algol

Algol、CPL、BCPL、B、Basic、PL/I、汇编语言、COBOL、FORTRAN、PL/I、Pascal、Modula-2、Ada、SmallTalk、Lisp、Java、J++、CGI、PERL、Visual Basic、C、C++和现在的 C#（读作“see-sharp”）——这个语言清单看起来有些吓人。为什么有这么多种语言？为什么不能发明一种语言实现所有这些语言的功能？应该学习哪种或哪些语言？

无论你是一位刚接触程序设计的人还是一位经验丰富的软件工程师，这都是一个好消息——C#容易学习、容易使用、所有都围绕着问题的解决方案。然而，在深入研究这种新语言之前，让我们来了解一下历史，看看我们是怎样到达 C#的。这一历史过程将回答前一段中所提出的所有问题。

在了解这一发展历史时，我们将发现今天很多语言的根源和构造基础，这一信息有助于正确使用这些新的开发工具。你还将了解包含在 C# 中的各项语言功能。从这些字里行间你会发现程序设计语言将来会怎样发展或变革。

1.1.1 为什么在 C# 中看到 C

研究 C 历史是值得的，因为它揭示了该语言成功设计的基本原理，这有助于理解为什么 C# 可能是未来几年选择使用的语言。我们对 C# 语言起源的“考古挖掘”从 Algol 60 开始。

Algol 60 是在引入 FORTRAN 几年后就出现的一种语言。这种基于欧洲语系的语言更加复杂，它对后来的程序设计语言有很大的影响。其作者非常注重语法规则、模块结构和其他通常与高级结构语言相关的功能。遗憾的是，Algol 60 从没有在美国真正流行起来，很多人认为这是因为该语言的抽象性和普遍性。

CPL (Combined Programming Language) 的发明者想把 Algol 60 崇高的目标带入实际计算机实现。然而，就像 Algol 60 难以学习和实现一样，CPL 也是这样。这导致其最终失败。BCPL (Basic Combined Programming Language) 的创建者仍然执着于 CPL 所提供的优点，他们想浓缩 CPL 的基本功能。

当 Ken Thompson 为 UNIX 的早期实现设计 B 语言时，他曾试图进一步简化 CPL。他成功创建了一种非常稀有的语言，这种语言非常适合在他所使用的硬件上使用（也就是 DEC PDP-7，具有 8 位寄存器长度）。然而，BCPL 和 B 是有局限性的语言，只适用于处理某些种类的问题。

例如，Ken Thompson 在 DEC PDP-7 上实现 B 语言时就出现了一种新机器 PDP-11（16 位字长）。虽然 PDP-11 是一种比其前辈 PDP-7 更大的机器，但是，按照今天的标准它仍然是相当小。它只有 24KB 内存（其中系统使用 16KB）和一个 512KB 的硬盘。一些人想用 B 重写 UNIX，但是因为 B 语言的解释设计，其运行速度慢。还有另外一个问题：B 是面向字节的，而 PDP-11 是面向字的。由于这些原因，关于 B 后继语言方面的工作从 1971 年开始，这被命名为 C（组合（Combine）其前辈的优点）。

这里我们需要讨论 UNIX 操作系统，因为该系统和在其上运行的多数程序都用 C 编写。UNIX OS 原来由位于新泽西州 Murray Hill 的 Bell 实验室开发。按照设计，该操作系统是对“程序员友好的”（programmer friendly），它提供有用的开发工具、较少的命令和一个相当开放的环境。然而，这并不意味着 C 依赖于 UNIX 或其他任何操作系统或机器。UNIX/C 协同开发环境为 C 争得一个荣誉：它是系统程序设计语言，因为它适合于编写编译器和操作系统。C 还非常适合于编写很多不同领域中的多数程序。

Dennis Ritchie 被认为是 C 的创建者，该语言恢复了在 BCPL 和 B 中丧失的通用性。他通过精确使用数据类型来实现这一点，并保持了简单性和对硬件的直接访问，这是 CPL 的最初设计目标。

单个人所开发的很多语言（C、Pascal、Lisp 和 APL）所包含的凝聚力正是大型程序设计小组创建的语言（Ada、PL/I 和 Algol 60）所失去的。一个人所编写的语言通常反映作者的专业技术领域。Dennis Ritchie 在系统软件——计算机语言、操作系统和程序生成器——设计方面的工作非常出名。

知道了 Ritchie 的专业领域，就很容易理解为什么 C 这种语言适合于系统软件设计。C 是一种相当低级的语言，允许指出算法逻辑中的每个细节，以获得最大的计算效率。但是 C 也是一种高级语言，它能够隐藏计算机结构的细节，因此提高程序效率。

1.1.2 C 与旧的高级语言

到这里，你可能会问：“怎样比较 C 与其他程序设计语言？”一种可以绘制的连续区域如图 1-1 所示。如果从该连续区域的底端向上移动，则从简单走向复杂，从经验走入理论。其中的点代表主要的进步，很多步骤被省略。

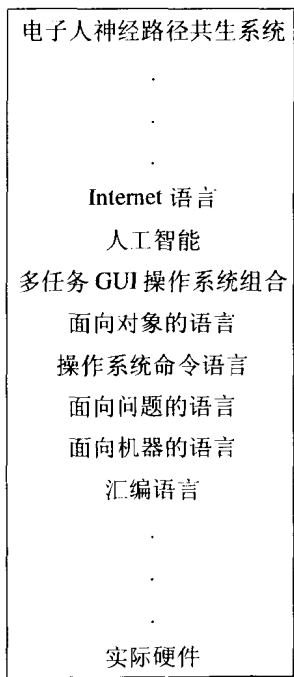


图 1-1 程序设计语言的理论演化

计算机的鼻祖，如 Jacquard 的织布机（1805）和 Charles Babbage 的“分析引擎（analytical engine）”（1834），都是用硬件进行程序设计。也许有一天，我们可以通过把神经路径通信器插入颞叶（temporal lobe）（语言内存）的插座或者大脑皮层的 Broca 区域（语言运动区域），进行编程。

最初的汇编语言可追溯到电子计算机的发明时代，它提供一种直接使用计算机内置指令集的方法，并且很容易学习，因为汇编语言要求按照硬件进行思考，所以必须用机器术语指定每个操作。因此总是把位移进或移出寄存器、对它们进行相加、把寄存器内容从一个寄存器移到另一个寄存器，并且最后把结果存储在内存中。这一工作是单调乏味的，而且容易出错。

最初的高级语言，如 FORTRAN，作为汇编语言的替代品被创建。高级语言更加通用和抽象，它们允许按照要解决的问题而不是按照计算机硬件进行思考。

遗憾的是，高级语言的创建者做了一些不合理的设想：每个建立标准的人更喜欢进一步自动化。他们省略一些必要的选项，这样做的好处是使程序设计简单化。FORTRAN 和 Algol 60 对系统级工作都太抽象，它们是面向问题的语言，一种用于解决工程、科学或商业中问题的语言。想编写系统软件的程序员仍然必须依赖于他们机器的汇编器。

根据对这一情况的反映，几个系统软件开发人员后退了一步——或者按照连续统一体是更低一点——创建了面向机器的语言。如你在 C 谱系中所看到的，BCPL 和 B 是适合这类非常低级的软件工具。这些语言非常适合具体的机器，但对其他则没有太大用处，它们与特殊的架构关系太密切。C 语言位于面向机器的语言之上一个台阶，但是仍然位于多数问题解决语言之下一个台阶。C 与计算机关系非常密切，使你能够更好地控制应用程序实现的细节，然而这足以忽略硬件的细节。这就是为什么 C 语言一开始被认为是高级语言和低级语言的原因。

1.1.3 C 的优点

每种计算机语言的源代码都有明确的外观，APL 有其象形外观，汇编语言有其记忆列，Pascal 语法容易阅读。很多第一次遇到 C 的程序员都发现其语法含义模糊，C 很少包含友好的类似英语的语法结构，而这在很多其他程序设计语言中都具有。相反，C 为软件工程师提供看起来不常用的运算符和大量的指针。新的 C 程序员会立即发现很多语言特性，其根源可以追溯到 C 原来的硬件/软件始祖。

如果你已经熟悉 C/C++ 的整套运算符，知道 C# 使用同样的定义则会感到很高兴。否则，如果你第一次遇到 C，则要概要学习一下 C/C++ 的运算符集。因为 C# 应用程序/小应用程序仍然有大量的机会与 C 或 C++ 代码交互，所以这具有双重优点。

在这里回顾一下 Ken Thompson B 语言（C 的直接前辈）后的起源和历史是有益的。下面是一个全面的 C 家族。

语 言	起源/发明者
Algol 60	由国际委员会在 20 世纪 60 年代早期设计
CPL	于 1963 年在剑桥和伦敦大学开发