Mc Graw Hill Education

C# 2.0
The Complete Reference
Second Edition

“本书将成为所有C#编程人员的必备参考。”
——Daniel Fernandez，
微软公司C#产品经理

C# 2.0
完全参考手册
(第2版)

(美) Herbert Schildt 著
赵铭 吴雷 译

免费源代码下载

- 详述 C# 语言的全部内容
- 涵盖 C# 2.0 的新特性，包括泛型、可空类型，以及匿名方法等
- 列举大量的示例和应用实例

Mc Graw Hill
清华大学出版社

C# 2.0 完全参考手册

(第 2 版)

(美) Herbert Schildt 著

赵铭 吴雷 译

清华大学出版社

北 京

Herbert Schildt
C# 2.0: The Complete Reference, Second Edition
EISBN: 0-07-226209-5

北京市版权局著作权合同登记号　图字：01-2005-5112

图书在版编目(CIP)数据

C# 2.0 完全参考手册(第 2 版)(美)希尔特(Schildt，H.)著；赵铭，吴雷译. —北京：清华大学出版社，2007.9
书名原文：C# 2.0: The Complete Reference, Second Edition
ISBN 978-7-302-15842-4

I.C…　II.①希…　②赵…　③吴…　III.C 语言—程序设计—手册　IV.TP312-62
中国版本图书馆 CIP 数据核字(2007)第 116382 号

责任编辑：王　军　王　婷
装帧设计：孔祥丰
责任校对：成风进
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：清华大学印刷厂
装 订 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185×260　**印　　张**：55.25　**字　　数**：1345 千字
版　　次：2007 年 9 月第 1 版　　**印　　次**：2007 年 9 月第1 次印刷
印　　数：1～4000
定　　价：99.80 元

前　言

编程人员往往喜欢精益求精，他们总是不断地想方设法提高程序的性能、效率和可移植性。因此，他们对所用的工具同样也要求甚多，特别是对编程语言的选择。编程语言的种类很多，但只有一小部分可称之为伟大。伟大的编程语言往往功能强大而又灵活，语法简洁且清晰，能够为创建正确的代码提供方便而不是设置障碍，能够不断地支持最新的功能，而不是逐渐被淘汰出局。最重要的是，伟大的编程语言注定会有这样一个无形的品质，那就是：让程序员在使用时感觉良好。C#就是这样一种编程语言。

C#是微软为支持.NET 框架而创建的语言，它建立在丰富的程序设计资源之上，它的首席设计师是长期以来公认的软件大师 Anders Hejlsberg。C#继承自今世界上最成功的两种计算机语言 C 和 C++，它继承了 C 语言的语法、部分关键字和运算符，并以 C++定义的对象模型为基础加以改进。C#还和另一非常成功的语言 Java 有紧密关系。C#和 Java 有共同的起源，像双胞胎一样，但是在许多重要方面两者也有所不同，所以 C#和 Java 更像是堂兄弟。例如，两者都支持分布式程序设计，并且都使用中间代码来获得安全性和可移植性，但是两者的实现细节是不同的。

尽管 C#构建在坚实的基础上，它所做的重要创新也是不容忽视的。例如，新增的委托、属性、索引器和事件等语言元素，新增的特性(attribute)语法。此外，C#消除了 COM 中的有关问题，简化了组件的创建。另外，它和 Java 一样，提供了许多运行时错误检查、安全性和可管理的执行。然而，不同于 Java，C#允许访问指针。因此，C#结合了 C++的功能和 Java 的类型安全。而且，它在安全和功能之间达到了最佳平衡，且实现了透明化。

在计算技术变革的历史中，为了适应计算环境的变化、计算机语言的发展，以及人们在思维方式和程序设计方式上的改变，程序设计语言得到了不断的发展，C#也不例外。在不断的提炼、适应和创新过程中，C#已经展示了它能不断满足现代编程人员需求的能力。C# 2.0 的发布就是最好的证明。

C# 2.0 是 C#语言最初 1.0 版本发布五年后的第一次重大修订。这些年中，编程人员对语言的新需求一直在增长。应用程序及其运行环境变得愈加复杂。C#通过增加一系列的创新特性来使编程人员更容易地写出更有弹性、更可靠和更高效的代码。具体体现在以下三

个方面：泛型使得创建类型安全、可重用的代码成为可能。部分类的声明允许类定义分散化，使得大型类的编写更容易。匿名方法简化了传递方法给委托类型的过程。总之，C# 2.0包含许多领先的特性，使得在不断变化的环境中编程变得更容易。

本书介绍了C#语言的方方面面，涵盖了C# 2.0中新增加的所有特性。

内容简介

编写C#书籍最具挑战性的问题之一是不知道该何时结束对它的讨论。由于C#语言本身就是一门博大精深的学问，而C#类库的内容更加庞大。为了更好地组织这些内容，本书分成如下3个部分：

- C#语言
- C#类库研究
- C#应用

第I部分全面讨论了C#语言，包括C# 2.0版中增加的新特性。这是全书内容比重最大的一部分，它描述了关键字、语法和一些定义C#语言的特性，并介绍了I/O操作、文件处理、反射和预处理程序。

第II部分研究了C#类库，同时也是.NET 框架类库。这是一个巨大的类库，由于篇幅有限，本书不可能深入探讨整个类库，而只能集中地讨论System命名空间中包含的核心类库，这也是和C#关系最紧密的库。另外，本部分还包括对集合、多线程和网络类库的介绍。这些是类库中几乎每个C#编程人员都会用到的部分。

第III部分给出了C#应用的实例。第25章展示了如何创建软件的组件，第26章描述了如何使用Windows Forms库构建Windows应用程序，第27章开发了一个递归表达式的分析程序。

读者对象

本书的读者无需具有任何编程经验。如果您已掌握C++或者Java语言，那么阅读本书时，将毫不费力，因为C#和这两种语言有很多共同之处。如果之前没有任何编程经验，您也能够通过对本书的学习逐渐掌握C#语言，但需要您仔细研究每章中的实例。

编程环境

必须使用Visual Studio 2005或更高版本来编译和运行C# 2.0程序。同时，必须在本机安装.NET Framework 2.0。

在线源代码下载

本书中所有程序的源代码都可以从www.osborne.com 和 www.tupwk.com.cn/downpage网站上免费获得。

反馈信箱

本书是Herb Schildt系列编程书籍之一，Herbert Schildt网站(www.HerbSchildt.com)上展示了更多他所编著的书籍，并提供了他的联系方式。最后，请将您的反馈意见发送至：wkservice@tup.tsinghua.edu.cn，我们将不胜感激。

目　录

第Ⅰ部分　C# 语 言

第 1 章　C#的起源……3
1.1　C#的族谱……3
1.1.1　C 语言：现代程序设计的开端……3
1.1.2　OOP 和 C++语言的创建……4
1.1.3　Internet 和 Java 的出现……4
1.1.4　C#的创建……5
1.1.5　C#的发展……6
1.2　C#如何与.NET 框架相关……7
1.3　公共语言运行库的工作原理……8
1.4　托管和非托管代码……8

第 2 章　C#概述……9
2.1　面向对象程序设计……9
2.1.1　封装……10
2.1.2　多态性……10
2.1.3　继承……11
2.2　简单示例一……11
2.2.1　C#命令行编译器 csc.exe……11
2.2.2　Visual Studio 集成环境……12
2.2.3　逐行分析第一个示例程序……15
2.3　处理语法错误……18
2.4　改写示例一……18
2.5　简单示例二……19
2.6　另一种数据类型……21
2.7　两种控制语句……22
2.7.1　if 语句……23
2.7.2　for 循环……24
2.8　代码块……25
2.9　分号、定位和缩进……27
2.10　C#语言的关键字……28
2.11　标识符……29
2.12　C#类库……30

第 3 章　数据类型、直接量和变量……31
3.1　数据类型的重要性……31
3.2　C#的数据类型……31
3.3　整数类型……32
3.4　浮点类型……34
3.5　decimal 类型……36
3.6　字符类型……38
3.7　布尔类型……38
3.8　自定义输出格式……39
3.9　直接量……42
3.9.1　十六进制直接量……43
3.9.2　字符转义序列……43
3.9.3　字符串直接量……44
3.10　变量……46
3.10.1　初始化变量……46
3.10.2　动态初始化变量……46
3.11　变量的作用域和生命周期……47
3.12　类型转换……50
3.12.1　自动类型转换……50
3.12.2　强制类型转换……51
3.13　表达式中的类型转换……54

第 4 章 运算符……59
4.1 算术运算符……59
4.2 关系和逻辑运算符……63
4.3 赋值运算符……68
4.4 位运算符……69
4.4.1 按位与、或、异或和取反运算符……69
4.4.2 移位运算符……75
4.4.3 位复合赋值……78
4.5 问号运算符“?”……78
4.6 空白符和圆括号……80
4.7 运算符优先级……80

第 5 章 程序控制语句……81
5.1 if 语句……81
5.1.1 if 语句嵌套……82
5.1.2 if-else-if 阶梯结构……83
5.2 switch 语句……85
5.3 for 循环……88
5.4 while 循环……96
5.5 do-while 循环……98
5.6 foreach 循环……99
5.7 使用 break 语句退出循环……99
5.8 continue 语句……101
5.9 goto 语句……102

第 6 章 类和对象……105
6.1 类基础……105
6.1.1 类的基本形式……105
6.1.2 定义一个类……106
6.2 如何创建对象……110
6.3 引用类型的变量和赋值……111
6.4 方法……112
6.4.1 给 Building 类添加方法……112
6.4.2 从方法返回……115
6.4.3 返回值……115
6.4.4 使用参数……117
6.4.5 给 Building 类添加带参数的方法……120
6.4.6 避免产生不可到达的代码…122
6.5 构造函数……122
6.5.1 带参数的构造函数……123
6.5.2 给 Building 类添加构造函数……124
6.6 new 运算符……125
6.7 垃圾收集和析构函数……126
6.8 this 关键字……129

第 7 章 数组和字符串……131
7.1 数组……131
7.2 多维数组……135
7.2.1 二维数组……135
7.2.2 三维或更多维的数组……137
7.2.3 初始化多维数组……137
7.3 非齐整数组……139
7.4 数组引用赋值……141
7.5 Length 属性……142
7.6 foreach 循环……146
7.7 字符串……149
7.7.1 构建字符串……149
7.7.2 操作字符串……150
7.7.3 字符串数组……153
7.7.4 字符串是不可变的……154
7.7.5 在 switch 语句中使用字符串……155

第 8 章 方法和类……157
8.1 控制对类成员的访问……157
8.1.1 C#的访问限定符……157
8.1.2 公有访问和私有访问的应用……159
8.1.3 访问控制：案例分析……160
8.2 给方法传递引用……164
8.3 使用 ref 和 out 参数……168
8.3.1 ref 关键字……169
8.3.2 out 关键字……170
8.3.3 对引用参数使用 ref 和 out…173
8.4 数量可变的参数……174
8.5 返回对象……177
8.6 方法重载……181

8.7 构造函数重载……186
8.8 Main()方法……191
8.8.1 从 Main()返回值……191
8.8.2 给 Main()传递参数……192
8.9 递归……194
8.10 static 关键字……196
8.11 静态类……202

第 9 章 运算符重载……205
9.1 运算符重载基础……205
9.1.1 重载二元运算符……206
9.1.2 重载一元运算符……208
9.2 针对 C#内置类型的数据重载运算符……212
9.3 重载关系运算符……216
9.4 重载 true 和 false……218
9.5 重载逻辑运算符……221
9.5.1 一种重载逻辑运算符的简单方法……221
9.5.2 使用短路运算符……223
9.6 类型转换运算符……227
9.7 运算符重载的注意事项……231
9.8 运算符重载的另一个示例……232

第 10 章 索引器和属性……237
10.1 索引器……237
10.1.1 创建一维索引器……237
10.1.2 索引器重载……241
10.1.3 索引器不需要一个潜在的数组……243
10.1.4 多维索引器……244
10.2 属性……247
10.3 对访问器使用访问限定符……252
10.4 使用索引器和属性……253

第 11 章 继承……259
11.1 继承基础……259
11.2 成员访问和继承……262
11.3 构造函数和继承……266
11.4 继承和名称隐藏……273
11.5 创建多级层次结构……276
11.6 构造函数的调用……279
11.7 基类引用和派生对象……280
11.8 虚方法和重写……284
11.8.1 为什么要重写方法？……288
11.8.2 应用虚方法……288
11.9 使用抽象类……292
11.10 使用 sealed 来阻止继承……296
11.11 object 类……296
11.11.1 装箱和拆箱……298
11.11.2 object 是通用数据类型吗？……300

第 12 章 接口、结构和枚举……303
12.1 接口……303
12.2 使用接口类型的引用……308
12.3 接口属性……311
12.4 接口索引器……312
12.5 接口的继承……314
12.6 接口继承引起的名称隐藏……315
12.7 显式实现……315
12.7.1 创建私有实现……316
12.7.2 使用显式实现来消除多义性……317
12.8 是接口还是抽象类？……318
12.9 .NET 标准接口……319
12.10 接口的实例研究……319
12.11 结构……324
12.12 枚举……329
12.12.1 初始化一个枚举……331
12.12.2 指定枚举的基本类型…331
12.12.3 使用枚举……331

第 13 章 异常处理……335
13.1 System.Exception 类……335
13.2 异常处理的基础……336
13.2.1 使用 try 和 catch……336
13.2.2 一个简单的异常示例……337
13.2.3 另一个异常示例……338
13.3 未捕获异常的后果……339

13.4 使用多个 catch 语句 ……342
13.5 捕获所有的异常 ……343
13.6 嵌套 try 模块 ……344
13.7 抛出异常 ……345
13.8 finally 语句 ……347
13.9 进一步分析异常 ……349
13.10 派生异常类 ……352
13.11 捕获派生类异常 ……356
13.12 checked 语句和 unchecked 语句 ……357

第 14 章 I/O 系统 ……361
14.1 C#的 I/O 依赖于数据流 ……361
14.1.1 字节数据流和字符数据流 ……361
14.1.2 预定义数据流 ……362
14.1.3 数据流类 ……362
14.1.4 Stream 类 ……362
14.1.5 字节数据流类 ……363
14.1.6 字符数据流封装类 ……364
14.1.7 二进制数据流 ……365
14.2 控制台 I/O ……365
14.2.1 读取控制台输入 ……365
14.2.2 使用 ReadKey()方法 ……367
14.2.3 写入控制台输出 ……369
14.3 文件数据流和面向字节的文件 I/O 操作 ……370
14.3.1 打开和关闭文件 ……370
14.3.2 从 FileStream 中读取字节 ……372
14.3.3 写入文件 ……373
14.3.4 使用 FileStream 复制文件 ……374
14.4 基于字符的文件 I/O 操作 ……376
14.4.1 StreamWriter 类 ……376
14.4.2 StreamReader 类 ……378
14.5 重定向标准数据流 ……379
14.6 读取和写入二进制数据 ……381
14.6.1 BinaryWriter ……381
14.6.2 BinaryReader ……382
14.6.3 二进制 I/O 操作的程序示例 ……383
14.7 随机访问文件 ……388
14.8 使用 MemoryStream ……390
14.9 StringReader 和 StringWriter …392
14.10 把数值型字符串转换为内部表示格式 ……393

第 15 章 委托和事件 ……397
15.1 委托 ……397
15.1.1 委托的方法组转换 ……400
15.1.2 用委托引用类的实例方法 ……400
15.1.3 多播委托 ……402
15.1.4 匿名方法 ……404
15.1.5 给匿名方法传递参数 ……405
15.1.6 从匿名方法中返回一个值 ……406
15.1.7 在匿名方法中使用外部变量 ……407
15.1.8 协变和逆变 ……409
15.1.9 System.Delegate ……410
15.1.10 使用委托的原因 ……411
15.2 事件 ……411
15.2.1 多播委托事件的示例 ……413
15.2.2 事件处理程序中实例方法和静态方法的区别 ..414
15.2.3 使用事件访问器 ……417
15.2.4 事件的其他特性 ……421
15.3 .NET 事件的规则 ……421
15.4 在事件处理中使用匿名方法 …424
15.5 事件的应用：案例分析 ……425

第 16 章 命名空间、预处理器和程序集 ……429
16.1 命名空间 ……429
16.1.1 命名空间的声明 ……430
16.1.2 命名空间可以避免名称冲突 ……432

16.1.3 using 命令……433
16.1.4 using 命令的另一种形式…435
16.1.5 命名空间的合成……436
16.1.6 嵌套命名空间……438
16.1.7 默认的命名空间……439
16.1.8 使用命名空间限定符(::)…440
16.2 预处理器……444
16.2.1 #define 命令……444
16.2.2 #if 命令和#endif 命令……444
16.2.3 #else 命令和#elif 命令……446
16.2.4 #undef 命令……448
16.2.5 #error 命令……448
16.2.6 #warning 命令……448
16.2.7 #line 命令……448
16.2.8 #region 命令和#endregion 命令……449
16.2.9 #pragma 命令……449
16.3 程序集和 internal 访问修饰符……450

第 17 章 运行时类型标识、反射和属性……453
17.1 运行时类型标识……453
17.1.1 is 运算符……453
17.1.2 as 运算符……454
17.1.3 typeof 运算符……456
17.2 反射……457
17.3 使用反射……459
17.3.1 获取方法的相关信息……459
17.3.2 GetMethods()的另一种形式……462
17.3.3 使用反射来调用方法……463
17.3.4 获取 Type 对象的构造函数……466
17.3.5 从程序集获得类型……470
17.3.6 全自动类型查询……475
17.4 特性……478
17.4.1 特性基础……478
17.4.2 创建特性……478
17.4.3 添加特性……479
17.4.4 获取对象的特性……479
17.4.5 位置参数和命名参数……481
17.5 3 个内置特性……485
17.5.1 AttributeUsage 特性……485
17.5.2 Conditional 特性……486
17.5.3 Obsolete 特性……487

第 18 章 泛型……489
18.1 泛型概念……489
18.2 一个简单的泛型示例……490
18.2.1 泛型类型因类型参数的不同而不同……493
18.2.2 泛型如何实现类型安全…493
18.3 一个使用两个类型参数的泛型类……496
18.4 泛型类的通用形式……497
18.5 类型约束……498
18.5.1 基类约束……498
18.5.2 接口约束……506
18.5.3 new()构造函数约束……511
18.5.4 引用类型和值类型约束…512
18.5.5 使用约束来建立两个类型参数之间的关系……514
18.5.6 多重约束……515
18.6 创建类型参数的默认对象…516
18.7 泛型结构……518
18.8 创建泛型方法……519
18.8.1 调用泛型方法时显式地指定类型参数……522
18.8.2 为泛型方法指定约束……522
18.9 泛型委托……523
18.10 泛型接口……526
18.11 比较同一类型参数的两个实例的值……530
18.12 泛型类的层次结构……533
18.12.1 使用泛型基类……533
18.12.2 泛型派生类……535
18.13 重写泛型类中的虚拟方法…536

18.14 重载带类型参数的方法……538
18.15 泛型类型的实例化……539
18.16 使用泛型时的一些局限……540
18.17 小结……540

第 19 章 不安全代码、指针和其他主题……541
19.1 不安全代码……541
19.1.1 指针类型……542
19.1.2 unsafe 关键字……543
19.1.3 fixed 修饰符……544
19.1.4 通过指针访问结构成员…545
19.1.5 指针运算……545
19.1.6 指针的比较……547
19.1.7 指针和数组……548
19.1.8 指针和字符串……550
19.1.9 多重间接寻址……550
19.1.10 指针数组……551
19.1.11 sizeof……552
19.1.12 stackalloc……552
19.1.13 创建固定大小的缓冲区……553
19.2 空类型……554
19.2.1 空类型基础……555
19.2.2 表达式中的空对象……556
19.2.3 ??运算符……557
19.2.4 在空对象上使用关系和逻辑运算符……558
19.3 局部类定义……559
19.4 友元程序集……560
19.5 其他关键字……561
19.5.1 lock 关键字……561
19.5.2 readonly 关键字……561
19.5.3 using 语句……562
19.6 const 和 volatile 修饰符……563
19.7 extern 关键字……563
19.7.1 声明 extern 方法……563
19.7.2 声明 extern 程序集别名…563

第 II 部分 C#类库研究

第 20 章 System 命名空间……569
20.1 System 的成员……569
20.2 Math 类……571
20.3 与内置数值类型对应的 .NET 结构……576
20.3.1 整型结构……577
20.3.2 浮点类型结构……579
20.3.3 Decimal 结构……583
20.3.4 Char 结构……587
20.3.5 Boolean 结构……592
20.4 Array 类……593
20.4.1 排序和查找……602
20.4.2 反转数组……604
20.4.3 复制数组……605
20.4.4 谓词……606
20.4.5 Action 委托……607
20.4.6 BitConverter 类……608
20.5 用 Random 产生随机数……610
20.6 内存管理和 GC 类……611
20.7 Object 类……613
20.8 IComparable 和 IComparable<T>接口……613
20.9 IConvertible 接口……614
20.10 ICloneable 接口……614
20.11 IFormatProvider 接口和 IFormattable 接口……616
20.12 IEquatable<T>接口……617

第 21 章 字符串和格式化……619
21.1 C#中的字符串……619
21.2 String 类……620
21.2.1 字符串构造函数……620
21.2.2 String 类的字段、索引和属性……621
21.2.3 字符串运算符……621
21.2.4 字符串方法……621
21.2.5 填充和剪裁字符串……637

21.2.6 插入、删除和替换……638
21.2.7 改变字母大小写……640
21.2.8 使用 Substring()方法……640
21.3 格式化类型……641
21.3.1 格式化类型概述……641
21.3.2 数值型数据的格式说明符……642
21.3.3 参数编号……643
21.4 使用 String.Format()和 ToString()格式化数据……644
21.4.1 使用 String.Format()格式化数值……644
21.4.2 使用 ToString()格式化数据……647
21.5 自定义数字格式……648
21.6 格式化日期和时间……651
21.7 格式化枚举……656

第 22 章 多线程程序设计……659
22.1 多线程基础……659
22.2 Thread 类……660
22.2.1 创建和启动线程……660
22.2.2 一些简单的改进……663
22.2.3 创建多个线程……664
22.3 确定线程结束的时间……667
22.4 为线程传递参数……669
22.5 IsBackground 属性……672
22.6 线程优先级……672
22.7 同步……675
22.7.1 实现同步的另一种方式…678
22.7.2 锁定静态方法……680
22.7.3 Monitor 类和锁……680
22.8 使用 Wait()、Pulse()和 PulseAll()实现线程通信……680
22.8.1 Wait()和 Pulse()的示例…681
22.8.2 死锁……685
22.9 MethodImplAttribute 属性……685
22.10 互斥锁和信号量……687
22.10.1 互斥锁……687
22.10.2 信号量……691
22.11 同步事件……694
22.12 Interlocked 类……697
22.13 终止线程……698
22.13.1 Abort()的另一种形式…700
22.13.2 取消 Abort()……701
22.14 挂起和恢复线程……703
22.15 判断线程的状态……703
22.16 使用主线程……703
22.17 多线程编程提示……705
22.18 开启独立任务……705

第 23 章 集合、枚举器和迭代器……707
23.1 集合概述……707
23.2 非泛型集合……708
23.2.1 非泛型接口……709
23.2.2 DictionaryEntry 结构……712
23.2.3 非泛型集合类……713
23.3 使用 BitArray 类存储位……729
23.4 专用集合……732
23.5 泛型集合……732
23.5.1 泛型接口……733
23.5.2 KeyValuePair<TK,TV>结构……736
23.5.3 泛型集合类……736
23.6 在集合中存储用户自定义的类……754
23.7 实现 IComparable 接口……756
23.7.1 为非泛型集合实现 IComparable 接口……756
23.7.2 为泛型集合实现 IComparable<T>接口……758
23.8 使用 IComparer 接口……759
23.8.1 非泛型 IComparer……760
23.8.2 泛型 IComparer……761
23.9 通过枚举器访问集合……763
23.9.1 使用枚举器……763
23.9.2 使用 IDictionary Enumerator……764

23.10 实现 IEnumerable 和 Ienumerator 接口……766
23.11 迭代器……767
23.11.1 停用迭代器……770
23.11.2 使用多个 yield 语句……770
23.11.3 创建命名迭代器……771
23.11.4 创建泛型迭代器……773

第 24 章 通过 Internet 连网……775
24.1 System.Net 的成员……775
24.2 统一资源标识符……777
24.3 Internet 访问基础……777
24.3.1 WebRequest 类……778
24.3.2 WebResponse 类……780
24.3.3 HttpWebRequest 类和 HttpWebResponse 类……781
24.3.4 一个简单的示例……781
24.4 处理网络错误……784
24.4.1 Create()产生的异常……784
24.4.2 GetResponse()产生的异常……784
24.4.3 GetResponseStream()产生的异常……785
24.4.4 使用异常处理……785
24.5 Uri 类……786
24.6 访问附加的 HTTP 响应信息……788
24.6.1 访问报头……789
24.6.2 访问 Cookie……790
24.6.3 使用 LastModified 属性…791
24.7 MiniCrawler:案例分析……792
24.8 使用 WebClient……796

第III部分 C# 应 用

第 25 章 构建组件……803
25.1 组件的概念……803
25.2 C#组件的概念……804
25.2.1 容器和站点……804
25.2.2 C#组件与 COM 组件……804
25.3 IComponent 接口……805
25.4 Component 类……805
25.5 一个简单的组件示例……806
25.5.1 编译 CipherLib……807
25.5.2 使用 CipherComp 的客户端程序……807
25.6 重写 Dispose(bool)……808
25.6.1 Dispose(bool)实例……809
25.6.2 防止使用已删除的组件…814
25.7 使用 using 语句……815
25.8 容器……816
25.9 组件是程序设计的未来……819

第 26 章 创建基于窗体的 Windows 应用程序……821
26.1 Windows 程序设计简史……821
26.2 编写基于窗体的 Windows 应用程序的两种方式……822
26.3 Windows 与用户交互操作的方法……822
26.4 Windows 窗体……823
26.5 基于窗体的 Windows 框架程序……823
26.6 添加按钮……825
26.6.1 按钮概述……826
26.6.2 给窗体添加按钮……826
26.6.3 一个简单的按钮示例……826
26.7 消息处理……827
26.8 使用消息框……830
26.9 添加菜单……832
26.9.1 创建传统样式的主菜单…833
26.9.2 使用 MenuStrip 创建新式菜单……837
26.10 总述……840

第 27 章 设计递归表达式的分析程序…843
27.1 表达式……843
27.2 分析表达式：发现问题……844
27.3 分析表达式：解决问题……845

27.4 解析表达式……846
27.5 一个简单的表达式分析程序……849
27.6 为分析程序添加变量……855
27.7 递归分析程序中的语法检查…863
27.8 尝试一些改进……864

附录 A XML 注释 ……865

PART I

C# 语 言

第 I 部分主要讨论 C＃语言的元素，包括：关键字、语法和运算符。另外还描述了一些和 C#语言紧密相关的 C#基本技术，比如：I/O 操作和反射。

第 1 章：
C#的起源
第 2 章：
C#概述
第 3 章：
数据类型、直接量和变量
第 4 章：
运算符
第 5 章：
程序控制语句
第 6 章：
类和对象
第 7 章：
数组和字符串
第 8 章：
方法和类
第 9 章：
运算符重载
第 10 章：
索引器和属性
第 11 章：
继承
第 12 章：
接口、结构和枚举
第 13 章：
异常处理
第 14 章：
I/O 系统
第 15 章：
委托和事件
第 16 章：
命名空间、预处理器和程序集
第 17 章：
运行时类型标识、反射和属性
第 18 章：
泛型
第 19 章：
不安全代码、指针和其他主题

1 CHAPTER

C#的起源

C#是微软.NET 开发者的首选语言，它具有的新特性经受住了时间的考验，且始终位于科技前沿，为现代企业计算环境提供了一种高可用且高效的编程方法。无论从哪个角度看，C#都是 21 世纪最重要的编程语言之一。

本章的目的是回顾 C#发展的历程，包括：C#创建的原动力、设计理念以及它是如何受其他计算机语言影响的。本章也解释了 C#和.NET 框架的相关性，正如你将看到的，C#语言和.NET 框架协同工作，一起构建了一个高度优雅的编程环境。

1.1 C#的族谱

计算机语言并不是凭空创造的，相反，它们彼此相关，新语言都或多或少地受到它之前的语言的影响。类似于异花授粉的过程，一种语言的特性会被另一种语言沿用，但新的创新内容会被集成到现有的环境中，而陈旧的构造则会被消除。就这样，编程语言不断地进化，编程艺术也不断地完善。C#也不例外。

C#继承了多种程序设计语言的精髓，它直接继承了当今最成功的两种计算机语言——C 和 C++语言的特性，并且与 Java 有紧密联系。理解它们之间的关系对于理解 C#语言是很重要的，因此，我们将分析这三种语言的发展环境，首先要研究的是 C#语言。

1.1.1 C 语言：现代程序设计的开端

C 语言的创建标志着现代程序设计时代的开始，它是 Dennis Ritchie 于 20 世纪 70 年代在一台装有 UNIX 操作系统的 DEC PDP-11 机上创造的。尽管一些早期的语言，最著名的如 Pascal 语言，已经取得了相当的成功，但 C 语言首先建立了面向过程编程的规范，至今仍适用。

C 语言成长于 20 世纪 60 年代的“结构化程序设计”革命。在结构化程序设计之前，大型程序很难编写，因为程序逻辑容易混乱而呈现所谓的“无头绪的代码”，比如掺杂大量