

别让你的计算机闲着



# 探秘 C#

如何像计算机科学家一样思考

How to Think Like a Computer Scientist  
C# Version

◆ 张平 编



浙江大學出版社  
ZHEJIANG UNIVERSITY PRESS

**别让你的计算机闲着**

# 探秘 C#

——如何像计算机科学家一样思考

张 平 编

浙 江 大 学 出 版 社

图书在版编目(CIP)数据

探秘 C#: 如何像计算机科学家一样思考 / 张平编.  
杭州: 浙江大学出版社, 2005.4  
(别让你的计算机闲着)  
ISBN 7-308-04142-5

I. 探... II. 张... III. C#语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 018657 号

丛书策划 张 明  
封面设计 张作梅

---

责任编辑 张 明  
出版发行 浙江大学出版社  
(杭州浙大路 38 号 邮政编码 310027)  
(E-mail: zupress@mail. hz. zj. cn)  
(网址: <http://www. zjupress. com>)  
排 版 浙江大学出版社电脑排版中心  
印 刷 浙江省邮电印刷厂  
开 本 787mm × 960mm 1/16  
印 张 29.75  
字 数 583 千  
版 印 次 2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷  
印 数 0001 - 3000  
书 号 ISBN7 - 308 - 04142 - 5/TP · 280  
定 价 40.00 元(附赠光盘一张)

## 内 容 简 介

C#语言是最新流行的面向对象的编程语言。它汲取了 Java 语言的精华,改进了 C++语言的微瑕,具有强大的功能和易开发特征,既能面向一般工程应用和网络应用开发,又能面向计算机底层应用,为广大编程者和学习者所青睐。

本书的特点有五:一是用类似英语教学中的情景教学方法,尽可能用简洁明快的方式按部就班地叙述。二是多用读者日常生活中信手拈来的趣味性实例作为编程材料。三是把计算机科学家们在编程实践中总结出来的诀窍“捅”给读者。四是用最清晰准确的语言介绍 C#语言的难点——属性、索引器、引用、装箱与拆箱、结构与类的区别等。五是所有的习题均给出答案,放在附赠的光盘中。

本书从结构上可分两部分。其中 1~14 章为 C#语言的基础部分,循序渐进地介绍 C#语言的各种要素:变量、操作符、条件语句、循环语句、嵌套与递归、引用、数组、结构、类等。15~21 章为数据结构部分,分门别类地介绍各种常用的数据结构:链表、堆栈、队列、优先队列、树、堆、映射表、哈希表以及哈夫曼码等。第 22 章简单介绍了图形编程和窗口编程。

另外,本书附赠的光盘含有 C#编程语言(Microsoft 公司的 .NET Framework Software Development Kit,简称 SDK)、优秀编程环境 SciTE 和 SharpDevelop 以及其他一些资料。

阅读本书的读者可以是真正的初学者,可以是在校学习的研究生、本科生或大专生,也可以是企事业单位的初、中级用户。最合适的读者是非计算机专业的大学本科生或大专生,以及中学生中信息技术的爱好者。

本书可用作各类学校的计算机课程教科书,也可用作学习计算机编程的参考书。

# 目 录

<b>第 1 章 按部就班的方式</b> .....	<b>1</b>
1.1 什么是程序语言.....	1
1.2 什么是程序.....	4
1.3 什么是调试.....	5
1.4 形式化语言和自然语言.....	7
1.5 第一个程序.....	9
1.6 术 语.....	10
1.7 练 习.....	12
<b>第 2 章 变量和类型</b> .....	<b>14</b>
2.1 再打印一些东西.....	14
2.2 变 量.....	16
2.3 赋 值.....	17
2.4 打印变量.....	18
2.5 关键字.....	20
2.6 操作符.....	21
2.7 操作符的执行顺序.....	22
2.8 对字符串的操作.....	23
2.9 组合句.....	23
2.10 术 语.....	24
2.11 练 习.....	25
<b>第 3 章 方 法</b> .....	<b>27</b>
3.1 浮点数.....	27
3.2 把 double 类型转换成 int 类型.....	29
3.3 数学函数.....	30
3.4 组 合.....	31
3.5 自己编写新方法.....	31
3.6 类和方法.....	34

3.7 具有多个方法的程序.....	35
3.8 形式参数和实际参数.....	36
3.9 堆栈状态图.....	38
3.10 具有多个参数的方法.....	38
3.11 具有结果的方法.....	39
3.12 术 语.....	40
3.13 练 习.....	40
<b>第 4 章 条件和递归.....</b>	<b>43</b>
4.1 模除运算.....	43
4.2 条件执行.....	44
4.3 选择执行.....	45
4.4 链式条件.....	46
4.5 嵌套条件.....	46
4.6 返回语句.....	47
4.7 类型转换.....	48
4.8 递 归.....	48
4.9 递归调用方法的堆栈图.....	50
4.10 惯例和神圣规则.....	51
4.11 术 语.....	52
4.12 练 习.....	53
<b>第 5 章 “开花结果”的方法.....</b>	<b>56</b>
5.1 返回值.....	56
5.2 逐渐“生长”的程序.....	58
5.3 组 合.....	61
5.4 重 载.....	62
5.5 布尔函数.....	63
5.6 逻辑运算符.....	64
5.7 布尔方法.....	65
5.8 递归的例子.....	66
5.9 确信跳跃.....	69
5.10 另一个递归例子.....	69
5.11 术 语.....	70
5.12 练 习.....	71

<b>第 6 章 重 复</b> .....	<b>76</b>
6.1 变量的多次赋值.....	76
6.2 重 复.....	77
6.3 while 语句.....	78
6.4 表 格.....	79
6.5 两维表.....	82
6.6 封装和泛化.....	82
6.7 方 法.....	84
6.8 进一步封装.....	84
6.9 局部变量.....	85
6.10 进一步泛化.....	86
6.11 术 语.....	88
6.12 练 习.....	89
<b>第 7 章 字符串和其他</b> .....	<b>92</b>
7.1 字符串类型.....	92
7.2 从字符串中提取字符.....	93
7.3 长 度.....	94
7.4 遍 历.....	95
7.5 运行错误.....	96
7.6 阅读说明文档.....	96
7.7 IndexOf 方法.....	97
7.8 循环和计数.....	98
7.9 加一和减一操作符.....	99
7.10 字符的算术.....	100
7.11 不可更改的字符串.....	102
7.12 字符串之间的比较.....	103
7.13 术 语.....	104
7.14 练 习.....	105
<b>第 8 章 结 构</b> .....	<b>110</b>
8.1 乏味的字符串.....	110
8.2 组合数据.....	110
8.3 点 (Point) 结构.....	111
8.4 对实例变量的存取.....	112

8.5	把对象作为参数.....	113
8.6	参数的值传递.....	114
8.7	参数的引用传递.....	115
8.8	矩形 (Rectangle) 结构.....	117
8.9	从方法中返回结构.....	118
8.10	以引用传递的方式传递其他类型的参数 .....	119
8.11	程序执行时的输入.....	120
8.12	术 语.....	123
8.13	练 习.....	124
<b>第 9 章</b>	<b>更多的结构.....</b>	<b>127</b>
9.1	时间(Time)结构.....	127
9.2	构造器.....	128
9.3	生成新对象.....	130
9.4	打印结构.....	131
9.5	数据封装与属性成员.....	131
9.6	方法作用的分类.....	135
9.7	无瑕作用.....	135
9.8	改动作用.....	137
9.9	填入作用.....	138
9.10	哪一种更好.....	139
9.11	发展型风格 vs 规划型风格 .....	139
9.12	进一步思考时间结构.....	141
9.13	更好的 Time 结构 .....	141
9.14	算 法.....	144
9.15	结构与简单类型.....	144
9.16	术 语.....	147
9.17	练 习.....	147
<b>第 10 章</b>	<b>数 组.....</b>	<b>152</b>
10.1	数组元素的存取.....	154
10.2	数组的拷贝.....	155
10.3	空引用与垃圾回收机制.....	156
10.4	for 循环.....	158
10.5	数组长度.....	159



10.6	随机数.....	159
10.7	随机数数组.....	160
10.8	计 数.....	162
10.9	频率直方图.....	163
10.10	一次遍历解决问题.....	164
10.11	素 数.....	165
10.12	快速获得素数.....	166
10.13	哥德巴赫猜想.....	169
10.14	二维数组.....	172
10.15	扫雷游戏中的数组.....	173
10.16	扫雷游戏的初始化(1).....	175
10.17	扫雷游戏的初始化(2).....	177
10.18	扫雷游戏的初始化(3).....	181
10.19	扫雷游戏过程的分步模拟.....	184
10.20	扫雷游戏过程的合成.....	189
10.21	锯齿数组与哥德巴赫猜想.....	195
10.22	术 语.....	198
10.23	练 习.....	198
<b>第 11 章</b>	<b>对象数组.....</b>	<b>203</b>
11.1	各种组合.....	203
11.2	扑克牌对象.....	203
11.3	PrintCard 方法.....	205
11.4	IsSameCard 方法.....	207
11.5	CompareCard 方法.....	208
11.6	扑克牌数组.....	210
11.7	寻找和搜索.....	211
11.8	一整副牌和一部分牌.....	215
11.9	术 语.....	216
11.10	练 习.....	216
<b>第 12 章</b>	<b>数组对象.....</b>	<b>218</b>
12.1	枚举(Enumerate)类型的纸牌.....	218
12.2	选择(switch)语句.....	221
12.3	类的定义和对象类型.....	223

12.4 Deck (一副纸牌) 类 .....	224
12.5 洗牌 .....	227
12.6 选择排序 .....	228
12.7 一手牌 (一部分牌) .....	229
12.8 洗牌和发牌 .....	230
12.9 混合排序 (Mergesort) .....	231
12.10 “配对”纸牌游戏 .....	233
12.11 从一手牌中删除指定的一张牌 .....	234
12.12 从一手牌中删除配对的两张牌 .....	236
12.13 游戏的基本操作单元 .....	239
12.14 由计算机来“玩”配对游戏 .....	241
12.15 术语 .....	244
12.16 练习 .....	244
<b>第 13 章 面向对象程序设计 .....</b>	<b>247</b>
13.1 程序设计语言及风格 .....	247
13.2 对象方法和类方法 .....	248
13.3 当前对象 .....	248
13.4 介绍一种对象——复数对象 .....	248
13.5 第一个复数方法 .....	250
13.6 另一个复数方法 .....	251
13.7 具有改动作用的方法 .....	252
13.8 打印复数 .....	253
13.9 引用类型的对象 .....	253
13.10 别名和引用 .....	255
13.11 空引用 .....	258
13.12 相同 .....	259
13.13 操作符重载：“==”和“+” .....	261
13.14 在对象方法里调用对象方法 .....	262
13.15 小心无大错 .....	262
13.16 引用类型的特点 .....	263
13.17 术语 .....	264
13.18 练习 .....	265
<b>第 14 章 继承 .....</b>	<b>266</b>

14.1	继 承.....	266
14.2	从复数 (Complex) 继承.....	266
14.3	ComplexSon 类.....	269
14.4	属 性.....	270
14.5	复数的显示输出.....	271
14.6	复数的加法.....	272
14.7	复数的乘法.....	273
14.8	标志符的四种访问特性.....	275
14.9	程序 (工程) 与命名空间.....	277
14.10	类的层级.....	279
14.11	术 语.....	280
14.12	练 习.....	280
<b>第 15 章</b>	<b>链 表.....</b>	<b>281</b>
15.1	对象的引用.....	281
15.2	节点 (Node) 类.....	281
15.3	链表是聚集器.....	284
15.4	链表与递归.....	286
15.5	无穷链表.....	287
15.6	原义含糊定理.....	288
15.7	针对节点的对象方法.....	289
15.8	改动链表.....	289
15.9	“外包装”和“内贤助”.....	291
15.10	IntList 类.....	291
15.11	真实量.....	293
15.12	术 语.....	294
15.13	练 习.....	294
<b>第 16 章</b>	<b>堆 栈.....</b>	<b>296</b>
16.1	抽象数据结构.....	296
16.2	抽象数据结构——堆栈.....	297
16.3	C#堆栈对象.....	297
16.4	装箱与拆箱.....	300
16.5	值类型数据进出堆栈.....	302
16.6	后缀表达式.....	303

16.7	语法分析.....	304
16.8	抽象数据结构的实施.....	308
16.9	运用数组来实施堆栈.....	309
16.10	变动数组大小.....	310
16.11	老鼠探迷宫.....	312
16.12	“建造”迷宫.....	313
16.13	在方格怎么“摸索”行走.....	315
16.14	怎样找到迷宫的通路.....	318
16.15	迷宫游戏的“序”与“跋”.....	320
16.16	术 语.....	323
16.17	练 习.....	324
<b>第 17 章 队列和优先队列.....</b>		<b>327</b>
17.1	队列抽象数据结构.....	328
17.2	装饰板.....	330
17.3	链接型队列.....	332
17.4	循环缓冲区.....	334
17.5	队列的应用——颜色区域的选择.....	338
17.6	优先队列.....	342
17.7	接 口.....	343
17.8	优先队列的数组实施.....	343
17.9	使用优先队列.....	345
17.10	高尔夫球手记分分类.....	347
17.11	术 语.....	349
17.12	练 习.....	350
<b>第 18 章 树.....</b>		<b>352</b>
18.1	树的节点.....	352
18.2	创建树对象.....	353
18.3	周游一棵树.....	354
18.4	表达式树.....	355
18.5	树的遍历.....	356
18.6	对象封装.....	359
18.7	定义 IVisitable 接口.....	360
18.8	IVisible 接口的实施.....	361

18.9	向量类 (ArrayList)	362
18.10	IEnumerator (巡流) 类	364
18.11	多态性与虚方法	366
18.12	抽象类与封闭类	369
18.13	接 口	372
18.14	建立表达式树	373
18.15	树、森林和二叉树	379
18.16	术 语	382
18.17	练 习	383
<b>第 19 章</b>	<b>堆</b>	<b>385</b>
19.1	用数组来实施树	386
19.2	算法占用资源的分析	390
19.3	混合排序的分析	392
19.4	附加消耗	394
19.5	优先队列的实施	395
19.6	堆的定义	396
19.7	从堆中删除元素	398
19.8	向堆中添加元素	399
19.9	堆中操作的时间特性	400
19.10	堆排序	401
19.11	术 语	402
19.12	练 习	402
<b>第 20 章</b>	<b>映射表</b>	<b>404</b>
20.1	数组、向量与映射表	404
20.2	映射表抽象数据结构	405
20.3	内置的哈希表	405
20.4	用向量实施映射表	409
20.5	映射表索引器	412
20.6	链表(IList)接口	413
20.7	哈希表的实施	413
20.8	哈希函数	414
20.9	哈希表的伸缩	416
20.10	哈希表伸缩的时间特性	417

20.11 术语.....	417
20.12 练习.....	418
<b>第 21 章 哈夫曼码.....</b>	<b>421</b>
21.1 变长码.....	421
21.2 字母频率表.....	422
21.3 哈夫曼树.....	424
21.4 base 方法.....	427
21.5 解码.....	429
21.6 编码.....	430
21.7 术语.....	431
<b>第 22 章 图形.....</b>	<b>433</b>
22.1 画板和图形对象.....	433
22.2 调用各种作图方法.....	435
22.3 坐标系.....	437
22.4 米老鼠的脸.....	438
22.5 另外的绘图命令.....	439
22.6 分形的米老鼠.....	440
22.7 艺术图形作品——花边图案.....	442
22.8 练习.....	444
<b>附录 A 与 C#环境混个脸熟.....</b>	<b>448</b>
A.1 下载和安装 C#语言.....	448
A.2 下载和安装 SciTE.....	451
A.3 配置 SciTE.....	451
A.4 SciTE 的功能.....	453
A.5 下载安装和第一次使用 SharpDevelop.....	455
<b>附录 B 练习分析与解答.....</b>	<b>460</b>
附赠光盘内容.....	461

# 第 1 章 按部就班的方式

这本书和这门课程的目标没有别的，就是教您像一位计算机领域的科学家那样去思考、去解决问题。我非常赞赏计算机科学家们的思考方式，因为他们的思考方式表现出数学家、工程师和自然科学家的某些最具特征的能力。对于计算机科学家来说，他们必须得像数学家一样，利用形式化的语言来表达思想和推导概念（尤其是计算）；他们得像工程师一样，想像和设计各种各样的零部件，并且得把它们装配成完整的系统，再对整个系统进行评价和测试，然后在各种各样可能的方案中进行权衡、折衷、取舍；他们还得像自然科学家那样，观察复杂系统的各种各样表现形式，进行各种假定，并且用实践来验证理论假设。

对于计算机科学家来说，最重要的技能就是解决问题的能力。这种能力意味着，要能够把所想要解决的问题予以概念化和公式化，能够创造性地和开创性地获得最终的结论，能够清晰地和精确地表达解决问题的途径。在阅读本书的过程中，我们将逐渐地学会用计算机编程的方式，用调试和实践的方法来解决实际问题，来获得这些能力。这就是为什么这一章取名为“按部就班的方式”的原因。

从某个方面来说，阅读本书，您就可以学会编程——当然，这是一种非常有用的技能。而在另一个方面，本书就是用按部就班的、渐进的方式，一步一步地引导您达到这个目标。随着您阅读下去，这个说法和用意将变得越来越清晰。

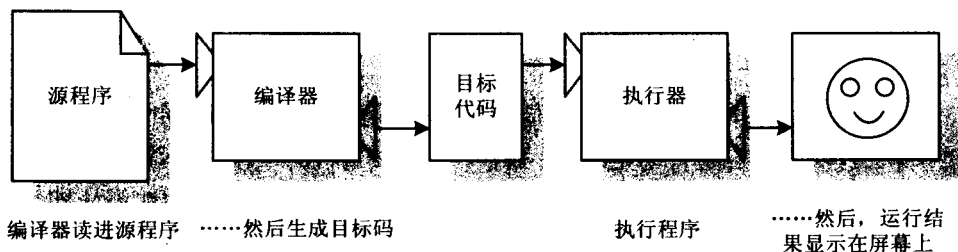
## 1.1 什么是程序语言

我们就要开始学习的这种程序设计语言的名字叫做 C#。C#语言是一种相当新





言编写的程序，并且一次性地全部翻译成机器语言，之后，才完整地执行整个程序。当然，更为普遍的对高级编程语言的编译处理过程还是分成几个步骤分别进行的，最后再统一执行。在编译型方式中，用高级编程语言撰写的程序被称为源代码，相应被翻译成机器语言的程序被称为目标码，也称为执行码。



举个例子，假定我们正在用 C 语言编写一个程序。首先，我们可用一个文字编辑器来编写程序（文字编辑器是一种简单的字处理程序）。当编好程序后，可以将这个程序取名为 `program.c`，把它在磁盘上存储为 C 语言文件。在此，`program` 是由我们自己随意命名的，而后缀 `.c` 是约定俗成，表示此文件的内容是 C 语言源代码。

至此，即可依据编程所在的环境进行不同的处理。一般，我们先关闭文字编辑器，然后调出并且运行 C 语言的编译程序。编译程序读入源程序，然后编译它，生成了一个名叫 `program.o` 的且在其中包含了目标码的新文件，甚至还可以一步生成包含执行码的 `program.exe` 执行文件。

但是，C# 编程语言却很不寻常，因为它既是编译型的，又是解释型的。C# 语言并不是直接转变成计算机可以直接运行的机器码，而是由编译程序生成一类特殊的 C# 语言特有的微软中间语言代码（MSIL）。说中间语言代码特殊是因为：一方面中间语言代码可像机器码一样，非常容易（也就是较快）被解释程序解释执行；另一方面，中间语言代码又像高级语言一样，移植性非常好。这样一来，C# 程序就可以在一台计算机中被编译成中间语言代码，然后通过网络转移到另一台机器上，由后者对中间语言代码进行解释执行。这就是 C# 编程语言比其他许许多多高级语言更为卓越的一个特征。