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。

本书可用作为各类学校的计算机课程教科书，也可用作为学习计算机编程的参考书。

目　录

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

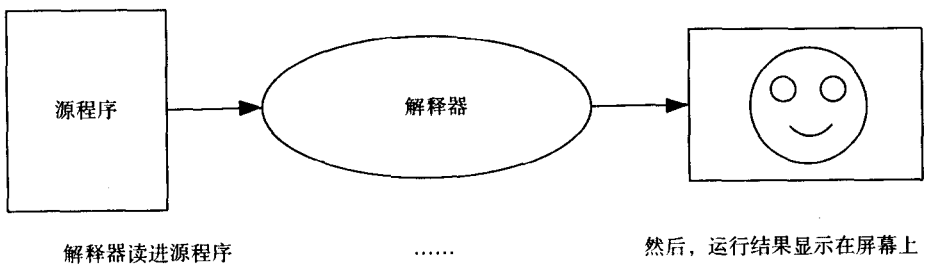
颖的编程语言（微软公司于 2000 年首次发布了这种语言），C#也是**高级编程语言**的一种。当然，你可能也已经听说过另外的一些高级编程语言，比如 Java，Pascal，C，C++和 FORTRAN。

说不定，目前你可能会对“高级编程语言”这一提法感到某种程度的困惑。事实上，对应地确实还存在着“低级编程语言”。在某种意义上可以说，低级编程语言才是“真正的计算机语言”，有时，它们也被称为“汇编语言”。应该这么说，计算机本身只能理解也只能运行用低级语言编写的程序。如果是用高级编程语言编写的程序，那么，它们在运行之前，必须进行先一步的预先处理，以便把它们“翻译”成计算机自身能够理解和执行的机器语言。这个预处理的过程，一般还是由计算机自身来进行，并且还需要一些处理时间和代价。从这个角度上说，这是高级编程语言的微瑕。

然而，好处却是巨大的。首先，用高级编程语言进行编程非常容易。所谓“容易”，其意思是用高级编程语言编写代码花费的时间较少、程序较短，而且容易阅读和理解，并且也容易改错。其次，高级语言的**移植性**非常好，这意味着不用修改或者仅作少量修改，就可以在不同型号的机器上运行。低级编程语言就不同了。一种低级编程语言只能在某一种特定型号的机器上运行，当想要在另一种型号机器上运行的话，就不得不重新予以编写。

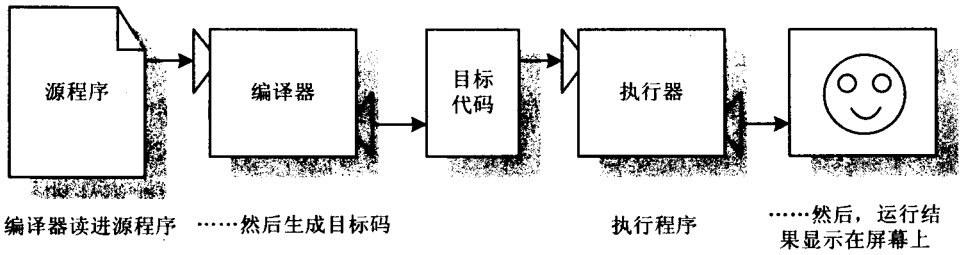
由于高级编程语言具有这样的优点，所以几乎所有程序都是用高级编程语言编写的。只有在非常特殊的某些应用场合，才用低级编程语言编写一小部分专门程序。

把高级编程语言转变成低级编程语言有两种处理方式。一种称为“**解释**”型方式，另一种称为“**编译**”型方式。如果采用的是解释型方式，解释程序（解释器）就一边阅读高级语言程序，一边就按程序所“说”（所指示）的去做。实际上，解释程序是一行接一行地循序“翻译”和“执行”程序的：它循序地阅读一行源代码程序，随即把它翻译成机器语言，然后就执行这些机器语言命令。



而采用编译型的方式时，编译程序（编译器）则一次性地全部读入高级编程语

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。