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
Learning with C++

[美] Allen Downey 等著
张 平 译

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

Allen Downey
[美] Jonah Cohen 著
Paul Bui

张 平 译

浙江大學出版社

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/（美）唐尼（Downey, A.B.）等著；张平译. —杭州：浙江大学出版社，2004.12
（别让你的计算机闲着）
ISBN 7-308-04035-6

Ⅰ.探… Ⅱ.①唐… ②张… Ⅲ.C 语言—程序设计
Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2004）第 119865 号

本书依据 *How to Think Like a Computer Scientist(Learning with C++)*一书译出

丛书策划 张 明
封面设计 张作梅

责任编辑 张 [illegible] 宋 铮
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 杭州杭新印务有限公司
开 本 787 mm × 960 mm 1 / 16
印 张 22.5
字 数 446 千
版 印 次 2004 年 12 月第 1 版 2004 年 12 月第 1 次印刷
印 数 0001—3000
书 号 ISBN 7-308-04035-6 / TP · 276
定 价 35.00 元 (附[illegible]

内 容 简 介

C++语言是目前最流行的面向对象的编程语言。它具有强大的功能，既能面向一般工程开发，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随手可拈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——指针、引用和模板。五是所有的习题均用脚注形式给出答案。

本书从结构上可分两部分。其中 1~17 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、指针、引用、数组、结构、类以及模板等。18~23 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、矩阵、集合以及文件等。

另外，本书附赠的光盘含有 C++编程语言（Borland 公司免费软件 BCC 5.5.1，自由软件 Dev-CPP.4960 和 Dev-CPP.4990 以及 Microsoft 公司的 Visual C++ Express Bata）、优秀编程环境 SciTE 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、小学生中信息技术的爱好者。

本书可用为各类学校的计算机课程教科书，也可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

这本书和这门课的目标没有别的，就是教您能像一位计算机领域的科学家那样去思考，去编程。我很喜欢计算机科学家们的思考方式，因为他们的思考方式，集中了数学家、工程师和自然科学工作者某些最具代表性的那些能力。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（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的预言。

对于计算机科学家来说，最重要的技能是**解决问题**的能力。这种能力意味着，能够把所想要解决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果，能够清晰地和精确地表达解决问题的途径，能够用计算机语言来最终解决问题。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。这就是为什么首章取名为“按部就班的方式”的原因。

从某个方面来说，阅读过本书，您就应该，而且可以进行编程——当然，这是一种非常有用的技能。而从另一个方面来说，本书所采用的渐进的方式，将能够一步一步地引导您达到这个目标。伴随着您的阅读，您将会越来越清晰地感觉到这一点。

1.1 什么是程序语言

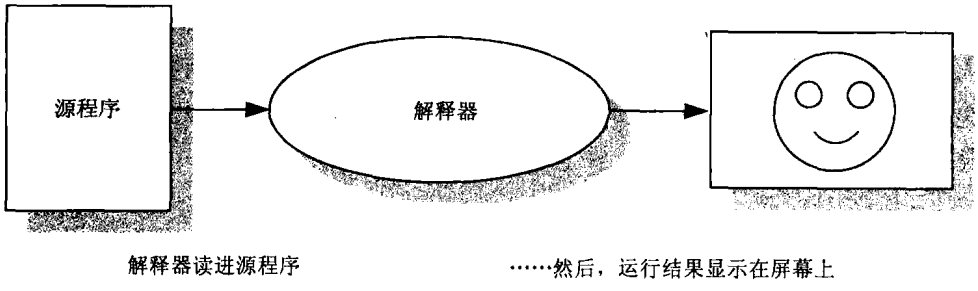
在本书中，我们将要学习的这种程序设计语言的名字叫 C++。自 1998 年以来，它就是美国 AP（College Board's Advanced Placement，学院高级课程）考试所用的计算机程序设计语言。在此之前，AP 考试用的是 Pascal 语言。C++和 Pascal 语言都是**高级编程语言**。当然，你可能也已经听说过其他的一些高级编程语言，如 Java，C 和 FORTRAN。

说不定，目前你可能会对“高级编程语言”这一提法感觉到某种程度的困惑。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“计算机的语言”，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。如果要运行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，以把它们翻译成为计算机自身能够理解和执行的低级的机器语言。这个增添的预处理过程，通常是由计算机自身来进行处理和完成的。无疑，进行这样的处理还是需要花费一些额外的处理时间的，所以从这个角度上说，它可算是高级编程语言的一点微瑕。

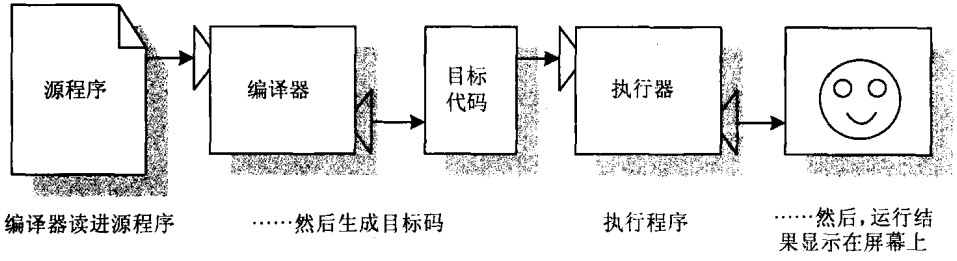
然而，用高级编程语言进行编程的好处却是巨大的。首先，用高级编程语言进行编程非常容易。所谓“容易”，其意思是用高级编程语言编写代码花费的时间较少，程序也较短，并且也容易阅读和理解，还容易改错。其次，高级语言的**移植性**非常好，这意味着不用修改，或者仅做少量修改，就可以在不同型号的机器上运行。低级编程语言就不同了。一种低级编程语言只能在某一种型号的机器上运行，当想要在另一种型号机器上运行的话，就不得不重新予以编写。

由于高级编程语言具有这样的优点，所以几乎所有的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，另一种称为**“编译”**型。如果采用的是解释型的方式，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“翻译”程序，循环地先阅读一行源代码程序，随即把它翻译成机器语言，然后迅速执行这些命令。



采用编译型的方式时，编译程序（编译器）一次性地读入全部高级编程语言，同时也一次性地全部翻译成机器语言，之后，就可完整地执行整个程序。一般而言，高级编程语言的编译，通常先分成几个步骤分别进行，然后再统一链接成一体。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**。



例如，在我们用C++语言编写某个程序时，首先，我们可先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.cpp，在磁盘上存储为cpp语言文件。在此，program由我们自己随意命名，而后缀.cpp则约定俗成，表示此文件的内容是C++语言的源代码。

至此，再依据编程所在的环境进行不同的处理。通常，我们就可以关掉文字编辑器，然后调出C++语言的编译程序并运行它。待编译程序读入源程序，就能编译它，生成了一个名叫program.o的，在其中包含了目标码的新文件，甚至可以一步性地生成包含全部执行码的program.exe执行文件。

再下一步就是执行这个程序。这时，就需要某种执行器。执行器的作用是装入程序（把程序从磁盘读取到内存中），随后执行该程序。

乍一看，这个过程相当复杂，但实际上好处很多：在许多编程环境中（某些时候叫做集成编程环境），这些步骤都可以自动完成。通常我们所需做的事情，就是编写程序，然后再输入一条命令，或按下键盘上的某按钮，让编程环境自动地编译

和执行它。应该说，即使您不知道以上这些过程，您也同样可以编程和编译；但是，从另一个角度看，了解隐藏在编程环境后面的各个步骤还是很有益处的。因为知道这些过程后，如果在编程和编译过程中出现了错误，我们就可以大致分析判断出错误是由什么原因或是在什么步骤上引起的。

1.2 什么是程序

程序，是一系列指令，它们指定计算机如何进行操作。计算机可以进行某些数学运算，像对方程组求解，或者找出多项式的根。它们也可以进行符号运算，例如，在文档中搜索和替换文字。甚至，它们还可以编译程序。（觉得有点奇怪吧！）

程序的指令（或者命令，或者语句）形式在各种编程语言中看起来似乎不同，但是，它们最基本的结构却几乎完全一样：

- **输入**：从键盘上、从文件中，或者从另外的设备里获得数据。
- **输出**：在显示屏上显示数据，或者写入文件内，或者发往其他设备。
- **计算和操作**：执行基本的数学操作，如加法、乘法；或者打印文字等各种操作。
- **按条件执行**：对条件进行判定，然后执行相关分支的一系列语句。
- **重复执行**：利用某些变化，重复地执行一些代码。

不管你信与不信，程序的结构就这么多。每一个你用到的程序，不管它们有多大，有多么复杂，但把它们结合起来的结构就是这些，所以，当您理解程序时，就可以把这些看上去极大而且非常复杂的程序，一步一步地分解成小小的分支，直到显示出上述的最基本的结构为止。

1.3 什么是调试和排错

编程是一个非常复杂的过程，又因为人们做事经常可能出错，所以程序内部隐含有一些错误就没有什么可以大惊小怪了。出于最初的一次偶然因素，程序中的错误被称为“小虫（bug）”，因而寻找它们的过程和排除程序中错误的工作就被称作