



TM

流行软件应用指南



程序语言编写指南

[美] Norman E. Smith, CDP 编著 招兆铿 译

Write Your Own Programming Language Using C++TM

電子工業出版社

香港万里机构出版有限公司



C++TM

流行软件应用指南

程序语言编写指南

Write Your Own Programming Language Using C++TM

[美] Norman E. Smith, CDP 编著

招兆铨 译

电 子 工 业 出 版 社

香港万里机构出版有限公司

Original title: Write Your Own Programming Language Using C++™

Copyright © Wordware Publishing, Inc. Reprinted in Chinese by Wan Li Book Company

Limited in arrangement with Wordware Publishing, Inc., Plano, Texas 75074, USA.

Chinese edition: C++ 程序语言编写指南 © 1995 by Wan Li Book Co., Ltd.

This simplified Chinese language edition is jointly published by the Electronic Industry Publishing House of China and Wan Li Book Co., Ltd. of Hong Kong, which in turn is granted an exclusive license by Wordware Publishing, Inc. for all Chinese language editions, and to be sold only in the PRC.

This is an authorized translation of the English edition © by Wordware Publishing, Inc., USA.

本书英文版由美国 Wordware Publishing, Inc. 出版, Wordware Publishing, Inc. 已将中文简体版独家版权授予香港万里机构出版有限公司和电子工业出版社。本书不得在港、澳、台及其他海外地区销售发行。未经许可,不得以任何形式和手段复制或抄袭本书内容。

流行软件应用指南

C++ 程序语言编写指南

[美] Norman E. Smith, CDP 编著

招兆铨 译

责任编辑: 焦桐顺

*

电子工业出版社

出版

香港万里机构出版有限公司

电子工业出版社发行 各地新华书店经销

北京市海淀区万寿路 173 信箱 (100036)

电子工业出版社计算机排版室排版

北京科技大学印刷厂印刷

*

开本: 850×1168 毫米 1/32 印张: 4.125 字数: 107 千字

1996 年 6 月第 1 版 1996 年 6 月北京第 1 次印刷

印数: 5000 册 定价: 7.20 元

ISBN 7-5053-3537-5/TP • 1432

版权合同登记号: 图字: 01-95-706 号

“流行软件应用指南”

出版说明

近几年来,随着计算机的发展及其广泛普及,它在人们的工作和生活中发挥着越来越大的作用。作为计算机“灵魂”的软件,不但功能越来越强大,而且还朝着“界面友好”、“易学易用”的方向发展。现在,使用计算机已不再是高深莫测的事情了。

这套丛书英文版由美国 Wordware Publishing, Inc. 出版,中文繁体版由香港万里机构出版有限公司出版。这套由资深计算机专家编著的“流行软件应用指南”丛书,精选了功能最为完善且被广泛使用的数种通用应用软件,包括操作系统、文字处理、数据库、电子表格、编程语言、网络系统等方面的应用软件。丛书以简明易懂的解说,辅以大量计算机屏幕实例,由浅入深地详细、系统介绍了各种应用软件的基本功能及其使用方法和技巧。能够使初学读者轻轻松松花费两三天或更短的时间学懂、学透一个通用软件。这套丛书也适合一般读者作为手头参考书。随时方便地查阅和练习,从而达到提高的目的。

这套丛书的英文版和中文繁体版出版后深受读者的欢迎。为了满足广大读者的需要,根据我国计算机的应用情况和读者的逻辑思维特点及阅读习惯,电子工业出版社对这套丛书进行了改写,使之更贴近广大读者的需求,并由电子工业出版社和香港万里机构出版有限公司联合推出这套丛书的中文简体版。

动手实践是学习计算机软件的有效捷径,这套丛书根据读者的

2 出版说明

需求,还精心设计了大量的练习题目供读者在学习中实践,以便读者熟练掌握软件的使用方法和技巧。这套丛书中大部分都有配套练习、演示磁盘,收录了书中大量的练习题目,读者可另行到新华书店选购或与电子工业出版社发行部软件销售科联系邮购。

电子工业出版社

1995 年 4 月

前 言

这是一本关于应用语言 (Application Language) 的书。更具体地说,这是关于如何编写你自己风格的应用语言。这本书的内容包括开始编写过程的必需工具以及一个完整的示例语言实现。

应用语言是嵌入某种应用程序的语言。它为用户提供编写通常称为宏指令的小程序,自动重复执行任务,或者对应用程序有用扩充的能力。许多商品化程序包含应用语言,只是没有采用应用语言这个名字。Lotus 1-2-3 有一种宏指令语言。Word Star 有命名的宏指令。dBASE 有一种语言。AutoCAD 有 AutoLisp。这些例子不胜枚举。《C++ 程序语言自编》是为普通的 C++ 程序员提供一些必需的工具,用于任何一个应用程序上建立一种应用语言。

目 录

第一章	关于本书	(1)
	为何编写客户应用语言	(2)
	应用语言的实现方法	(3)
第二章	CALC 指南	(6)
	CALC 的运行及基本内容	(6)
	下一步：编译宏指令	(11)
	CALC 文件	(15)
	深一层的介绍	(16)
	其他用途	(17)
第三章	CALC 的设计和实现	(19)
	设 计	(19)
	CALC 的操作	(21)
	小 结	(26)
第四章	样本程序	(29)
	MORE 的设计	(29)
	小 结	(34)
第五章	CALC 与 MORE	(35)
	操作原理	(35)
	对 MORE 的修改	(35)
	小 结	(37)
第六章	非常规贯穿式解释语言	(39)
	设计构想	(39)

	背 景	(42)
	贯穿类别	(44)
	UNTIL 的设计	(45)
第七章	UNTIL 的数据结构	(47)
	栈	(47)
	字典	(49)
	其他结构	(53)
	小 结	(55)
第八章	UNTIL 解释器	(57)
	内部解释器	(57)
	外部解释器	(61)
	外部解释器代码	(63)
	宏指令的执行	(66)
	输入流的名法分析	(72)
	小 结	(74)
第九章	CALL 编译器	(75)
	CALL 编译器的概貌	(76)
	原 语	(78)
	高层宏指令	(78)
	文 字	(82)
	分支,循环和条件	(83)
	小 结	(90)
第十章	UNTIL 的核心原语	(91)
	加进原语	(91)
	原 语	(97)
	小 结	(99)
第十一章	求助设施	(101)
	操作原理	(101)
	求助代码	(102)

小 结	(107)
第十二章 发展方向	(109)
使用和提高	(109)
从 C++ 转换到 C	(112)
小 结	(112)
附录 1 源文件	(113)
附录 2 参考资料	(115)
书 籍	(115)
机 构	(115)
附录 3 CALC 的参考卡	(117)

第一章

关于本书

编写一种客户应用语言 (Custom Application Language) 不是很难的事情。问题在于要知道从何入手。许多商业性的软件产品都提供应用语言。例如, AutoCAD 提供 AutoLisp, Procomm Plus 有 ASPECT。应用语言会使各种程序增加专业性能。

本书介绍简单的应用语言 CALC 的实现方法。在使用目前流行的 Borland International, gnc.。产品 Borland C++ 或者 Turbo C++ 编写的互动式 (交互式) C++ 程序上, 都可以附加 CALC 语言。这种语言是由贯穿式解释器 (Threaded Interpreter) Until (UNconventional Threaded Interpretive Language 的缩写词) 以及 RPN (逆波兰记法) 编译器 (Compiler) CALL (Callable Application Language Library 的缩写) 实现的。

头几章介绍 CALC 语言以及用它编写的程序样例。后几章给出实现应用语言的主要软件技术 (Until 解释器和 CALL 编译器)。

本书中的代码用 Borland C++ 开发。

本书假定你已经在 PC 上装入并且运行 Borland C++ 或者 Turbo C++, 熟悉在该环境内编辑和编译程序。注意尽可能让代码一般化, 以便能够毫无困难地将它们移植到其它 C++ 甚至 C 编译器上运行。

为何编写客户应用语言

编写客户应用语言的原因有多种。其实,要增加用户会用到的但又与程序无直接关系的功能,使用应用程序是一种好办法。最佳的例子是能够让用户编写新的宏指令(Macros)的计算器语言。

应用语言对于一些需用重复的或者复杂命令来完成一个动作的情况最为有用。

其实主要的原因是,每种商业性的文字处理器、試算表(Spreadsheet)和通讯程序都有某种宏命令或者应用语言。所起的作用包括,从在 PFS:Professional Write 中简单的击键(Keystroke)记录和四放功能,到完整的程序设计语言,例如 GNU Emacs 中的 Mocklisp。

含有完整语言的程序向用户提供某种办法,使这种应用扩充或者做成适合其日常工作环境。倘若在所有应用之间都使用一种核心的公用宏命令/应用语言,对用户便有了一致的共同语言。通过应用的一致性,可以降低培训和文档的费用。

用户方面倾向于选择满意的产品。因此,要使公用的应用语言具备一致性,还要费相当的力气。

应用语言无需提供源代码或者对象库程序,就能够让用户访问内部的程序数据结构。例如,应用程序可以让用户在一份报告上改变每页的行数。

编写应用语言的另一个主要原因是程序的测试。访问程序数据结构的功能,很容易被纳入到互动式应用语言内。对编译的变量值进行处理,可以相当程度上提高测试的速度。但是,在使用 Borland C++ 时,由于提供了内部的排错(Debugging)环境,上述的作用便不明显。倘若在你的 C/C++ 编译器上设有这些性能,在排错程序时,互动地访问变量的方法会明显地节省时间。

应用语言的实现方法

宏命令语言或应用语言的实现方法有多种,包括:

- 在缓冲器上记录击键
- 预定义要执行的命令
- 编译要执行的新命令
- 贯穿式解释器

为得到如本书上给出的合适工具,而开发你自己的应用语言,只需略懂编译器的知识。本书开发一个应用语言样例,包括编译器和解释器。这个应用语言的名字是 CALC,编译器是 CALL,以及解释器是 Until。

以下几段将对会用到的一些应用语言实现方法,作概括的介绍。

记录击键

实现宏命令或应用语言的最简单办法是记录击键。将每个键复制入一个或多个缓冲器(Buffer),存放起来供以后再使用。通常一个宏命令列由某个特殊键,如 Alt-A 或者名字表示。很少涉及循环或逻辑条件的功能。文字处理器通常采用这个办法。

预定义命令

有些程序采用稍许不同的办法实现打字处理的应用语言。所提供的功能完全由开发者定义。预定义命令方法的例子,如许多通讯程序中的自动拨号设施。这能使用户越过正常的操作方式,利用调制解调器(Modem)打电话。预定义命令通常被指定为某种特定的命令键组合,譬如 Alt-D。

编译命令

有些程序包含专供具体应用的编译器。例如 Procomm Plus 中的

ASPECT, 和 AutoCAD 中的 AutoLisp。它们也许是, 也许不是传统意义上的全编译器, 总之, 在实现上是复杂的。

贯穿式解译器

贯穿式解译器或贯穿式解释性语言 (TIL-Threaded Interpretive Language 的缩写) 是把编译器和解释器合成一体。同 TIL 有关的过程、函数、字或宏命令都被编译。由一个独立的解释器执行自编译器建立的“中间引线 (Threads)”。TIL 一般是互动性的。最著名的 TIL 是 Forth。TIL 比解释性的 Basic 速度稍为快一些。TIL 提高了八十年代中期典型编译器实施方法的速度。据我们所知, Microsoft Quick 语言采用了贯穿式方法。

书写风格

本书约定采以下的书写风格以提高文本的阅读性。

- 用斜体表示 C++ 的标识符和保留字。
- 用 Shift 类键和一字母键构成两键的组合。

Shift 类键包括 Alt、Ctrl 和 Shift 等键。例如“按 Ctrl-C 以退出”的意思是, 按下 Ctrl 键的同时按 ‘C’ 键。“按”是指应该按下一个键或者两键的组合才会工作下去。

组织

每一章讨论单个专题。其中包括对该专题或性能的引言。在合适之处给出一些例子。引言部分作为所讨论专题的形式定义。一章内还有几节来给定的专题作全面的说明。

编码风格

本书所用的编程风格, 目的在加强例子中 C++ 代码的清晰性。多数 C 程序员的编码风格是, 尽可能地将语句合并。例如, 判断字符串长度等于 9 的典型方法是:

```
if(strlen(xxx) == 9) { ... }
```

本书的编码风格把这段代码编写成：

```
len = strlen(xxx);
```

```
if(len == 9) {
```

```
    ...
```

```
}
```

既然要介绍的是新概念,提高代码的可读性应该比程序的绝对效率更加重要。

练习磁盘

本书另备练习磁盘,收录了本书给出的源代码,可供读者选购。

第二章

CALC 指南

前一章介绍了应用语言的概念和一个例子, CALC。本章将为大家提供一个对 CALC 全面介绍。内容包括被作为一个独立程序编译的 CALC 的用法。本章中的例子介绍 CALC, 同时打算说明, 应用语言作为你的 C++ 工具箱有用部分所具有潜力。

CALC 使用一种先写分量后写操作符的 RPN 型操作。CALC 不使用中缀记法而用 RPN 记法, 原因在于如何编写编译器。CALC 使用 Until 以提供互动性解释性能, 以及用 CALL 编译宏指令。Until 是贯穿式解释语言, 但并不表示据此建立的应用语言一定是 RPN 型的。Until 只是 CALC 使用的解释器。这正是由编译器来决定了, 应用语言使用 RPN, 中缀的或后缀的操作记法。C/C++ 使用中缀记法, 而 Lisp 使用后缀记法。关于 CALL 编译器将在以后几章中详细讨论。

本章提供一些例子让你了解 CALC 的操作。这些例子亦会涉及编译新的 CALC 宏指令。本章还会补充应用语言的一些概况, 以及说明除应用语言外 CALC 和 Until 的一些用法。

CALC 的运行及基本内容

请将本书练习磁盘上的文件 CALC.EXE、CALC.APP 和 HELP.APP 拷贝到你的硬盘上。这些例子向你介绍多个内部的

CALC 命令。

1. 在 DOS 提示符下键入 CALC, 按 Enter 键启动 CALC。

CALC

Copyright 1992

All Rights Reserved

Version 1.1

JANUARY 1992

CALC> Welcome to the CALC Programmable Calculator

Type 'macros' to show list of available macros.

Type 'help help' to type the macro usage.

Type 'exit' to exit CALC and return to DOS.

你可以随时键入 **hello** 再次显示启动信息。屏幕右上角主要用来显示数据栈(Data Stack, 又称数据堆叠)。数据栈空时, 便显示 'Stack Empty'。每次往栈上推入一个数或执行一个宏指令后, 便更新显示的内容。

注 意

任何时候键入 **exit** 再按 Enter 键, 便退出 CALC 并且返回 DOS。

2. 键入 **12 + =** 和按 **Enter** 键执行 1 加 2。记号 '=' 打印结果 3。

CALC> 1 2 + =

3