



美国最畅销“傻瓜”丛书

最新出版



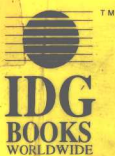
“傻瓜”系列

C++编程指南

C++ For Dummies

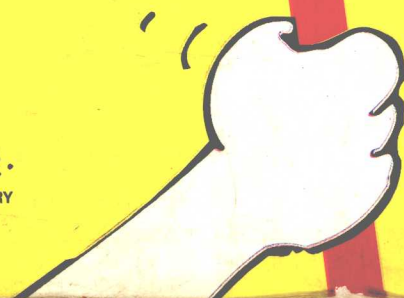
〔美〕 Stephen R. Davis 著
卢凌云 等译

- 介绍简单有趣的方法学习C++编程
- 为你提供面向对象编程的指导
- 用通俗的语言告诉你如何编写强力程序



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

从C转到C++的理想
参考书



美国最畅销傻瓜丛书

C++编程指南

[美] Stephen R. Davis 著

卢凌云 等译

电子工业出版社

内 容 提 要

本书以 ANSI C++ 为准,同时也兼顾以后很可能被 ANSI 正式采纳的某些新内容,在全面介绍了 C++ 的基本功能基础上,围绕对于学习 C++ 必须掌握的核心概念——“类”,由浅到深地对构造函数、析构函数、成员、继承、多态性等重点进行剖析,并配以大量例子,详细阐述了为什么需要这些概念,它们是怎样工作的,什么情况下应该使用它们,以及怎样使用它们,以使读者掌握 C++ 的精髓,并能够编写出高效的面向对象程序。本书既可作为各类院校 C++ 课程的教材,也适用于计算机软件和其他工程技术人员阅读。

C++ For Dummies by Stephen R. Davis

"Copyright© 1995 by Publishing House of Electronics Industry.

Original English language edition copyright© 1995 by IDG Books Worldwide, Inc.

All rights reserved including the right of reproduction in whole or in part in any form.

This edition published by arrangement with the original publisher, IDG Books Worldwide, Inc., Foster City, California, USA.

... *For Dummies* is a trademark of International Data Group".

本书获得 IDG Books Worldwide, Inc. 正式授权,在中国大陆内翻译发行,未经许可,不得以任何形式和手段复制或抄袭本书内容。

美国最畅销“傻瓜”丛书

C++ 编程指南

卢凌云 等译

特约编辑:俞天林 张孝仁 责任编辑:洋 溢

电子工业出版社出版(北京市万寿路)

电子工业出版社总发行 各地新华书店经销

北京科技印刷厂印刷

开本:787×1092 毫米 1/16 印张:21 字数:530 千字

1996 年 1 月第一版 1996 年 1 月第一次印刷

印数:5000 册 定价:40 元

ISBN7-5053-3108-6/TP·1095

著作权合同登记号 图字:01-95-305 号

译者前言

C++以其对C的全兼容性和完善的性能,已成为最受计算机软件开发人员欢迎的面向对象程序设计语言之一。越来越多的软件系统将C++作为源程序语言,许多高等院校已把C++作为计算机专业的必修课程。掌握了C++,从某种意义上讲,就是在竞争中为自己添加了制胜的一个关键砒码。

就国内所出版的有关C++的书籍而言,绝大多数都是“手册式的”。通过阅读这些书籍,你能够了解诸如数据类型、变量定义形式、语句格式、库函数以及编译连接过程等C++的一系列功能。如果你要用某一种C++语言来实际编程,你应该有一本该种C++语言的手册(或类似的书籍)。然而使用手册的侧重点是告诉你C++各种要素的语法格式,为什么要用和怎样合理地使用这些要素和格式,往往不是手册所包含的内容。对于功能十分强大、使用极为灵活的C++,了解它的功能是一回事,怎样掌握和利用这些功能来高效地完成实际编程任务,又是另一回事。

由Stephen R. Davis所著的本书,正是针对各种C++手册未包含,而真正掌握C++又必须具备的知识而编写的。它并不是对某一种具体的C++的描述(虽然书中的程序例子能直接在大多数C++编译器,例如Borland C++下编译连接),而是以ANSI C++为准(同时也兼顾以后很可能被ANSI正式采纳的某些新内容),围绕对于学习C++必须掌握的核心概念——“类”,对构造函数、析构函数、成员、继承、多态性等重点进行剖析,并配大量例子,以使读者明白为什么需要这些概念,它们是怎样工作的,什么情况下应该使用它们,怎样使用它们。本书的最终目的就是要使读者掌握C++的精髓,并能够编写出高效的面向对象程序。

本书的引言、第一部分和第二部分由齐剑锋翻译,第三部分由于洁和王希武翻译,第四部分和第五部分由卢凌云翻译,第六部分、第七部分和附录由王红霞翻译。全书由卢凌云统稿和校对。

在本书翻译过程中得到了军械工程学院计算机教研室王森教授的指导和帮助,在此表示衷心感谢!

译 者

1995年8月于石家庄

目 录

引言.....	(1)
第一部分 预备课程:C 语言回顾	(7)
第一章 C,C 的变量、函数和表达式	(9)
声明简单变量.....	(9)
存储分类	(10)
全局变量	(11)
局部变量	(11)
声明函数	(12)
C 的表达式	(13)
运算符的优先级	(14)
算术运算符	(14)
位运算符	(15)
逻辑运算符	(15)
赋值运算符	(16)
杂项运算符	(17)
关于特殊运算符的考虑	(17)
结束语	(18)
第二章 C 指针	(19)
声明和使用简单指针	(19)
指针的运算	(21)
指针增量	(21)
指针的其它运算	(22)
数组和指针:我们能把它联系在一起吗?	(22)
向函数传递指针	(23)
函数指针	(24)
堆内存	(25)
结束语	(26)
第三章 用户自定义类型:C 的结构.....	(27)
C 结构 101	(27)

C 结构指针	(28)
结构指针的运算	(29)
使用结构指针	(29)
结构和函数	(30)
从堆中分配结构	(31)
结束语	(31)
 预算程序的 C 版本: BUDGET1. C	(33)
25 分钟练习	(39)
 第二部分 C++ 的非面向对象的特征	(43)
 第四章 进入 C++	(45)
我们是如何到达这里的?	(45)
最初的年代(1950~1960)	(45)
函数的年代(1960~1975)	(46)
结构化的年代(1975~1990)	(46)
面向对象的年代(1990~现在)	(46)
为什么 C 要++?	(46)
结束语	(47)
 第五章 几个简单的问题	(49)
新的注解风格	(49)
为什么需要它?	(49)
它怎样工作?	(49)
在任何地方声明变量	(50)
为什么需要它们?	(50)
它们怎样工作?	(50)
常数变量	(52)
为什么需要它们?	(52)
它们怎样工作?	(52)
挥发性变量(Volatile Variables)	(54)
为什么需要它们?	(54)
它们怎样工作?	(55)
引用类型说明符	(55)
为什么需要它?	(56)
它怎样工作?	(56)
结束语	(58)

第六章 声明函数	(59)
函数原型	(59)
为什么需要它们?	(60)
为什么我喜欢编译器挑我的错?	(60)
它们怎样工作?	(61)
内联函数(Inline Functions)	(63)
为什么需要它们?	(63)
在 C 中我做不到吗?	(63)
它们怎样工作?	(64)
快了多少?	(65)
函数重载	(66)
为什么需要它?	(67)
它怎样工作?	(67)
什么可以作为“本质区别”?	(68)
什么不足以区别重载函数?	(68)
该做啥就做啥,但是别重组函数名.....	(69)
函数的缺省变量	(70)
为什么需要它们?	(71)
它们怎样工作?	(71)
和重载的冲突	(71)
结束语	(72)
第七章 I/O 流	(73)
在我有了 printf()后,为什么还需要流?	(73)
关于 I/O 流的案件:第一部分	(73)
关于 I/O 流的案件:第二部分	(74)
I/O 流怎样工作?	(74)
结束语	(76)
把预算程序改写为 C++ 程序:BUDGET2. CPP	(77)
25 分钟练习	(83)
第三部分 类的入门	(87)
第八章 面向对象的程序设计	(89)
抽象和微波炉	(89)
函数方法下的 nachos 制作	(90)
面向对象方法下的 nachos 制作	(90)
分类和微波炉	(91)

函数的分类	(91)
面向对象的分类	(91)
为什么要分类?	(91)
面向对象的程序设计和效率	(93)
结束语	(94)
第九章 在C++中加入类	(95)
为什么把类增加到C++中?	(96)
怎样把“类”加入C++中?	(96)
命名成员函数	(97)
在类中定义一个成员函数	(98)
在类之后保持一个成员函数	(99)
调用一个成员函数	(99)
用指针调用成员函数	(100)
在成员函数中访问成员	(101)
重载成员函数	(102)
结束语	(103)
第十章 保护成员	(105)
保护成员	(105)
为什么需要它?	(105)
它怎样工作?	(106)
再谈为什么应该使用保护成员?	(107)
类能够保护它的内部状态	(107)
通过一个限制接口很容易使用类	(108)
通过限制接口很容易支持类	(108)
简单地说友元函数是什么?	(111)
为什么需要友元函数?	(111)
它们怎样工作?	(111)
结束语	(113)
第十一章 构造函数	(115)
建立对象	(115)
构造函数	(117)
为什么需要它们?	(117)
它们怎样工作?	(118)
析构函数	(121)
为什么需要它?	(121)
它怎样工作?	(121)

结束语.....	(123)
第十二章 找出类.....	(125)
面向对象的分析和设计.....	(125)
一个分析问题和设计解决方案的例子.....	(126)
快速分析和设计:一种结构化方法	(127)
快速分析和设计:一种面向对象的方法	(127)
一个对该问题的面向对象的解决.....	(129)
结束语.....	(130)
使用类的预算程序:BUDGET3.CPP	(131)
25 分钟练习	(137)
第四部分 轻松自如地运用类.....	(143)
第十三章 制作构造函数的参数.....	(145)
带有参数的构造函数.....	(145)
为什么需要它?	(146)
它怎样工作?	(146)
重载构造函数.....	(148)
缺省的缺省构造函数.....	(150)
构造类成员.....	(151)
构造对象的顺序.....	(155)
局部对象有序地进行构造.....	(156)
静态对象只被构造一次.....	(156)
所有全局对象在 main() 之前被构造	(156)
构造全局对象无特殊顺序.....	(157)
成员以其被声明的顺序进行构造.....	(158)
析构函数以构造函数相反的顺序被调用.....	(158)
结束语.....	(159)
第十四章 更具有价值的新关键字.....	(161)
新关键字:现在你可以不再使用 malloc() 和 free() 了	(161)
为什么需要它们?	(161)
它们怎样工作?	(162)
分配数组.....	(162)
结束语.....	(164)
第十五章 拷贝构造函数.....	(165)

拷贝构造函数.....	(165)
为什么需要它?	(165)
它怎样工作?	(166)
自动拷贝构造函数.....	(167)
浅拷贝对比深拷贝.....	(169)
对临时对象还有很多要了解的东西.....	(171)
结束语.....	(173)
第十六章 改变对象的类型	(175)
无名对象.....	(175)
为什么需要它们?	(175)
它们怎样工作?	(176)
用构造函数进行类型转换.....	(177)
结束语.....	(179)
第十七章 静态成员	(181)
静态数据成员.....	(181)
为什么需要它们?	(181)
它们怎样工作?	(182)
访问静态数据成员.....	(182)
静态数据成员的用途.....	(184)
静态成员函数.....	(185)
结束语.....	(187)
维护一个更合理的预算: BUDGET4.CPP	(189)
25 分钟练习	(195)
第五部分 继承	(199)
第十八章 继承	(201)
为什么需要继承?	(201)
继承怎样工作?	(203)
构造子类.....	(204)
“有一个”关系.....	(205)
结束语.....	(206)
第十九章 虚成员函数	(207)
为什么需要多态性?	(209)
多态性怎样工作?	(210)

用多态性方式制作 Nacho	(211)
什么时候虚函数不被多态地重载	(213)
什么时候定义为“虚”的	(215)
结束语	(216)
第二十章 类分解和抽象类	(217)
分解	(217)
抽象类	(222)
它们怎样工作?	(223)
由一个抽象类制造出一个实在类(honest class)	(224)
传递抽象类	(225)
为什么需要纯虚函数?	(225)
决定实时类型	(226)
结束语	(229)
用继承来改造 Budget 程序: BUDGET5. CPP	(231)
25 分钟练习	(237)
第六部分 可选的特征	(247)
第二十一章 多重继承	(249)
多重继承如何工作?	(249)
继承的模糊性	(250)
虚拟继承	(251)
构造多重继承的对象	(255)
结束语, 一个相反的建议	(255)
第二十二章 访问控制说明符	(257)
真正“专有的俱乐部”: 私有成员	(257)
什么时候用 private, 什么时候用 protected?	(257)
如何使用 private?	(258)
非公共继承	(258)
什么条件下不是子类?	(261)
结束语	(261)
第二十三章 运算符重载	(263)
为什么需要重载运算符?	(263)
如何操作一个运算符函数和一个函数?	(264)
这与重载运算符有什么关系?	(265)

重载运算符是如何工作的?	(265)
更详细看一看	(266)
作为成员函数的运算符	(268)
重载的另一个麻烦事情	(269)
转换运算符	(271)
结束语	(272)
第二十四章 赋值运算符:	(273)
为什么重载赋值运算符如此重要?	(273)
如何重载赋值运算符?	(274)
结束语	(275)
第二十五章 I/O 流	(277)
流 I/O 如何工作?	(277)
fstream 子类	(278)
stringstream 子类	(281)
控制符	(281)
一般插入运算符	(283)
灵活的插入运算符	(285)
结束语	(287)
第二十六章 对象有效性和标志域	(289)
用无效指针调用成员函数	(289)
结果怎么样?	(290)
如何对待无效指针?	(291)
isLegal()还能干什么?	(293)
结束语	(293)
第七部分 进一步的说明	(295)
第二十七章 避免在程序中出现错误的十种方法	(297)
使所有的警告信息和错误信息开关有效	(297)
使用 STRICT 编译	(298)
坚持“干净的”编译	(298)
采用清晰和一致的编码风格	(299)
限制可见性	(300)
使用标志域	(301)
写代码时给出注释	(301)
对每条路径至少单步跟踪一次	(301)

不要重载运算符.....	(302)
避免多重继承.....	(302)
第二十八章 本书前面未包括的近十个 C++ 特征	(303)
模板.....	(303)
异常处理.....	(304)
运行时的类型标识(RTTI)	(305)
名称空间.....	(305)
重载 new 和 delete	(305)
String 类.....	(305)
指向成员运算符的指针—>* 和.*	(305)
Intel 16 位指针	(306)
第二十九章 十个最重要的编译器开关(附加两个).....	(307)
内联(inline)函数的 outline 选项	(307)
定义预处理符号.....	(308)
在.obj 中包括调试信息	(308)
检查堆栈溢出.....	(309)
存储器模式.....	(309)
浮点类型的支持.....	(309)
编译器优化.....	(310)
使能异常/RTTI 处理	(311)
数据边界对准.....	(311)
处理器支持.....	(312)
标准堆栈结构.....	(312)
预编译头文件.....	(312)
附录 虚函数到底是如何工作的?	(313)
v_table	(314)
结束语.....	(316)
词汇.....	(317)

引言

欢迎阅读本书。这本书是充分考虑了读者掌握C++的需要而编写的，它给你所须知道的一切，而不会使你生厌。

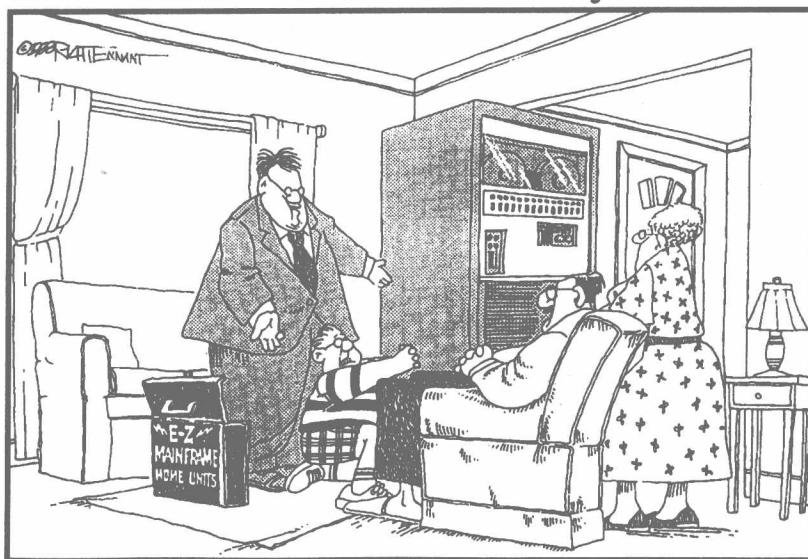
关于这本书

本书眼光集中于你和你的需要。如果你得到这本书，一定会令你非常兴奋的。

如同C的类似读物，这本书是一本教材。也就是说，要从头到尾读完它（当然不是一口气读完——否则你的家人会等急的）

The 5th Wave

By Rich Tennant



"OH SURE, \$1.8 MILLION SEEMS LIKE A LOT RIGHT NOW, BUT WHAT ABOUT RANDY? WHAT ABOUT HIS FUTURE? THINK WHAT A COMPUTER LIKE THIS WILL DO FOR HIS S.A.T. SCORE SOMEDAY."

这本书并不是要使你成为C++“语言律师”（就是一个熟悉计算机语言细节的人，一个象我这样的真正语言专家）。尽管我谈及了该语言的每一方面，但在C++的核心特征上用了更多时间，在你尚不需要的“强大用户”特征上用时较少（当你对这个语言适应以后，你可以从手册中发现它们）。

不同于其它的C++编程书籍，本书把“为什么”看得比“怎样”更重要，C++的特征就象拼图玩具的每一片配件，我不是仅仅告诉你这些特征，而是让你理解它们是如何相互适应，组合到一起的。

如果你没有弄明白语言中的某个特征是什么，你就不能真正理解它怎样工作。当你看完这本书，你将能够写出合理的C++程序，同样重要的，你将明白为什么

需要它和它是怎样工作的。

这本书不能告诉你怎样安装你的编译器——有太多不同的编译器,并且它们在不断地改变。另外,我不能进入如何录入一个程序、如何编译它等诸如此类事情的细节。当你学会编辑一个文件,编译、连接、运行结果程序,你会对此书有更深理解。(尽管如此,我还是拿出一章来讲编译器的选项)。

什么是 C++?

C++,顾名思义,就是 C 语言的下一代:即把学术界计算机语言学的新思想加到老的 C 语言上得到的结果。

C++是以 C 语言为核心建立的,它几乎完全向上兼容。C 能做到的任何事,C++都能做到,甚至可以用同样的方式做到。但是 C++ 决不仅是披上了光彩的新外衣的 C,从 C 到 C++ 的扩展是重大的,需要改变一些思想和习惯,但结果是值得的。这本书将帮助你尽可能轻松地从 C 转到 C++。

这本书讲解的是标准 C++,即得到了 ANSI 的 C++ 标准委员会认可的 C++。这是由 Borland C++,Turbo C++,Visual C++,Symantec C++和大多数其它公司实现的 C++ 版本。诸家卖主都使自己的产品尽可能地贴近这个标准,所以本书谈到的一切,在各种 C++ 版本中都同样得到很好的支持。当我确实提到一些 Borland 和 Microsoft 独有的变化形式时,都标记上“非标准”图标。

这本书适合你吗?

这本书假定你已经至少懂得一部分 C。如果你对 C 一点都不懂,那么马上停下来,把它放回,然后在书架上去找 C 语言教程(记住你把这本书放在哪里了,因为尽管你现在不需要它,将来你还是会回来买它)。

如果你对 C 语言的知识有些生疏了,但你基本上还懂或者至少在某段时间曾经懂 C,那就可以摸出您的信用卡付款。第一部分回顾了 C 的绝大多数的重要问题,它可以使你很快熟悉原来的知识。

这本书是怎样组织的?

每一个新特征都通过回答下面的三个问题来介绍:

- ✓ 这个新特征是什么?
- ✓ 为什么要把它引入到语言中?
- ✓ 它怎样工作?

每章中都有许多小段的代码,每一段代码都示例一些新介绍的特征或强调一些我认为精采的地方。这些小段代码可能不完整,自然也不能做什么有意义的事情。

在每一部分的结尾.....

为帮助读者掌握本书的知识,我在前五部分的每一部分后面,各安排了两组辅助内容:一个示例程序和一个练习。

我觉得弄明白 C++ 的各种特征在一个完整的程序中怎样一起起作用很重要。但是,若让我讲解许多不同的示例程序,我觉得很难办。那样我就必须花更多时间来说明每个程序能做什么,而用较少的时间来讲解它所包含的语言特征。另外,由于这些程序完成不同的功能,所以把它们作比较非常困难。

因此,我使用了一个简单的示例程序:预算程序(BUDGET)。这个程序在第一部分中的后面以函数式 C 程序出现,以后,我每次都重写它,以加入其后各部分提出的特征。在我们到达第五部分结尾时,预算程序发展成为一个完全的面向对象的 C++ 程序,更进一步就成为成熟的面向对象程序。有人可能认为这太浪费时间(如果是这样,可以跳过去)。但是,我希望你通过这个预算程序弄明白 C++ 的各种特征是如何一起起作用的。

第二个辅助措施是练习,用这些问题试一试(我给出了一些提示和警告,以使你保持正确的思路)。我在练习的后面给出了答案和解释。你解答完毕后,可与之作一比较。

第一部分 预备课程:C 语言回顾

在第一部分,我们要确保所有读者都处在同一起跑线上。如果你已经对你的 C 语言知识很满意,就可以快速浏览这部分内容。如果你确实感觉很自信,可以跳过它。但是一定要看看预算程序,并回答第一部分后的问题,以检查一下你的知识。

第二部分 C++ 的非面向对象的特征

在第二部分,向你介绍了一些 C++ 的非面向对象的特征,包括新的注释风格,内联函数和 I/O 流。这些特征使 C++ 成为更好的 C,而不是成为一门新语言。这些特征可以相当快地掌握,因为它们不要求你改变你的函数式方法。

第三部分 类的人门

在第三部分,你已深入 C++,开始了解它和 C 的区别。我介绍了面向对象程序设计概念,讨论了类。这两点都使 C++ 成为一个确实与 C 不同的语言。

第四部分 轻松自如地运用类

关于使用类的面向对象程序设计的介绍是一大步。这要求你重新测验你的函数式程序设计方式,重新思考你认为你在很久以前解决过的问题。第四部分给了你足够的时间,使面向对象的程序设计方法溶入你的思想。

第五部分 继承

正是继承将面向对象设计同结构化设计区分开了。理解这一最重要的概念,是进行有效的 C++ 程序设计的关键和第五部分的目标。现在已无路可退——完成这一部分后,你可以自称为面向对象的程序员,当然,是初级的。

第六部分 可选的特征

在进入第六部分之前,你已经知道了用 C++ 进行高效的程序设计所需的全部知识。我简略地论述了剩下的可选问题。你可以按你喜欢的任何顺序读本部分的各章,如果你乐意,可把它们作为参考。

第七部分 进一步的说明

什么? 这本书可以不用这几个关于十的列表结束? 在第七部分的第一章,你将找到避免在程序中出现错误的最好的方法。然后,根据大家的兴趣,我简单介绍了本书中未说明的特征(这些特征中的大多数仅在最近才被 ANSI C++ 标准委员会采纳,但还没有被编译器实现)。

你注意到现在的编译器有多少不同的选项吗? 怎样知道我的虚表指针是否跟在我的成员指针的后面? 怎样选用快速浮点指针? 慢速浮点指针呢? 在第七部分,我将指导你弄清这些选项,并指出哪些是重要内容,哪些最好先不要管它。

哪些你不必读

C++ 是一个大药丸,有好吃的部分,也有不太好吃的部分。为避免你被一时不感兴趣的信息打搅思路,我把技术资料用特殊图标标志出来(图标见下页)。

另外,有些背景信息插入到副框中。如果你觉得信息量太大,那么第一遍阅读可以跳过这些段落(记住有了时间还要读这些内容。在 C++ 里,你不懂的东西最终将会使你受到损害)。

在本书中使用的图标



这是技术资料,第一遍阅读可以跳过。



提示要点,可以节省你许多时间和精力。



记住它,它很重要。