



美国最畅销“傻瓜”丛书

OBJECT-ORIENTED PROGRAMMING FOR DUMMIES®



“傻瓜”系列

面向对象的编程指南

〔美〕 Namir C. Shammas 著
宋炎 曹旭华 等译
彦君 校

- 介绍简单有趣的方法学习面向对象的编程（OOP）技术——全新
- 建立和使用C++的类来代表现实世界的对象的第一个帮助工具
- 揭示OOP的技巧，你可使用任意编译程序（DOS、Windows、UNIX或Macintosh）



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

OOP速查表

美国最畅销“傻瓜”丛书

面向对象的编程指南

[美] Namir C. Shammas 著

宋 炎 曹旭华等 译
彦 君 校

电子工业出版社

内 容 提 要

本书介绍了 OOP 的支柱并解释了连接面向对象编程技术的 C++ 的特性以及如何模拟和管理真实世界对象使用 C++ 类的方法。本书分为四部分,第一部分介绍了面向对象的程序设计、类以及对象的基本概念;第二部分介绍了 C++ 中主要 OOP 相关特性;第三部分介绍了一些高级 OOP 相关话题;第四部分列出了类的元件,提供了使用虚拟成员函数的指南。

本书为学习面向对象编程的读者提供了浅显易懂但又极为实用的内容。

Object-Oriented Programming For Dummies by Namir C. Shammas.

"Copyright(©) 1996 by Publishing House of Electronics Industry.

Original English language edition copyright(©) 1996 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

面向对象的编程指南

宋 炎 曹旭华等 译

彦 君 校

特约编辑:尚 凡 责任编辑:洋 溢

*

电子工业出版社出版(北京海淀区万寿路)

电子工业出版社发行 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:20.5 字数:515 千字

1996 年 9 月第一版 1996 年 9 月第一次印刷

印数:5000 册 定价:32.00 元

ISBN7-5053-3272-4/TP·1220

著作权合同登记号

图字:01-96-0638 号

译者序

本书介绍了 OOP 的支柱并解释了连接面向对象编程技术的 C++ 的特性以及如何模拟和管理真实世界对象使用 C++ 类的方法。面向对象程序设计方法(OOP)不仅受到程序员的喜欢,也受到用户的青睐。过去的十六年相继出现了 OOP 语言,如 SmallTalk, Eiffel 和 C++。即便是非 OOP 语言,如 Ada 和 COBOL,也正在把面向对象的设计特征作为最新的美国国家标准协会(ANSI)的一部分。由此看来 OOP 的影响力是不言而喻的。本书将把你带入一个使用 C++ 的面向对象程序设计世界。

为适应国内计算机应用的发展,我们翻译了《面向对象的编程指南》一书。参加本书翻译、校对工作的还有李斌、王伟亮、张华、程玮、丁世文、李庆林、马重妮、马英杰、樊开元、马志伟、赵军、郑小京、刘青。彦君同志进行了全书的统编及审校工作。本书的录入排版工作由刘文华、张卫、王芳、武超、许晓平、王娜、李琳琳、赵靖、董松圆、宋明华完成,在此表示感谢!

书中疏漏之处,欢迎广大读者给予批评和指正。

译者

一九九六年六月于北京

关于作者

Namir C. Shamma

Namir C. Shamma 是专职作者,他擅长于撰写面向对象的程序设计和 Windows 程序设计方面的书籍。他自己撰写和与他人合著的程序设计方面的书籍已多于 50 本,像《Foundations of C++ Programming》(IDG Books Worldwide, Inc.)。他获得了化学工程的学士和硕士学位。

目 录

前言	(1)
关于本书的说明	(1)
关于本书的代码	(1)
本书的编排顺序	(2)
第一部分 OOP 的基本概念	(2)
第二部分 舒适阶段的巡游	(2)
第三部分 高级 OOP 的特性	(3)
第四部分 十准则集粹	(3)
本书中所用的图标	(3)
敢问路在何方?	(4)
 第一部分 OOP 的基本概念	(5)
第一章 面向对象编程:基本状况	(7)
结构化程序设计:沿着回忆之路前行	(7)
面向对象的程序设计:此地此刻	(8)
OOP 的四大支柱	(9)
类:对象家族值	(9)
对象:火车、飞机以及..... ..	(10)
方法和消息:我们可以谈谈吗?	(11)
继承性:把财富留给子类	(12)
多态性:灵巧的 OOP 形态切换	(14)
小结	(14)
第二章 类:对象工厂	(15)
C++ 类:OOP 对人们的影响	(15)
类的部分:控制成员访问	(16)
使用数据成员来定义类实例的状态	(17)
谁说电视机没有类	(18)
箱子问题	(19)
使用成员函数:类允许者	(21)
选择成员函数(仔细地)	(23)
另一 TV 插曲	(23)
关于箱子更有趣的东西	(25)
存储类元件	(27)
理解成员函数的作用	(28)
初始化函数	(29)
数据代理函数	(32)
执行函数	(34)
访问函数	(37)

辅助函数	(39)
常量函数:你不可以碰它!	(42)
生成一个猜测游戏	(44)
小结	(48)
第三章 对象的生命周期:在…生活方式下的 scOOP	(49)
请给出一消息	(49)
电视机例子(更多次)	(49)
箱子问题(再一次!!!???)	(54)
过载成员函数:不要烧保险!	(58)
初始化对象:从哪儿着手呢?	(62)
使用缺省构造函数:都市中唯一的游戏	(63)
带拷贝构造函数的中性类实例	(68)
利用定制构造函数:独有的价值	(73)
毁坏对象:DOOM, OOP 方式!	(75)
与对象打交道(9 到 5)	(79)
小结	(80)
第二部分 舒适阶段的巡游	(81)
第四章 静态成员:属于它们自己的类吗?	(83)
静态数据成员:属于类的成员	(83)
声明静态数据成员:静态建立!	(84)
初始化静态数据成员:粉碎带静态的类	(85)
静态成员函数:静态调用!	(86)
实例计数:进行计数!	(87)
常见错误处理:情况很糟糕	(91)
共享信息:乐于共享!	(96)
专有成员	(102)
保护成员	(102)
公共成员	(102)
函数 main()	(103)
小结	(104)
第五章 类设计:好的、差的和劣质的	(105)
分析对象:你和你的父类关系如何?	(105)
第一流的类设计	(106)
内聚:函数中的法则	(106)
耦合:功能连接	(114)
低劣的类设计:真正的失去者?	(115)
改进我们的类设计:版本更新!	(122)
小结	(127)
第六章 继承:子类涉及的专题	(129)

单重和多重继承性: OOP 家族的领地	(129)
声明单继承类层次: 单亲类!	(131)
子类的构造函数: 与其层次相连	(132)
使用派生类: 继承和显示家族特征	(137)
非公共派生: 寂静成员	(142)
类 myArray	(146)
类 myStack	(146)
小结	(147)
第七章 多态性: 一致响应	(149)
总览多态性: 大 OOP 变形图画	(149)
虚拟(现实)函数	(151)
随机时间例子: 任何人都有正确时间吗?	(152)
类 RandomTime1	(156)
类 RandomTime2	(156)
类 RandomTime3	(156)
函数 main()	(156)
数字输入例子: 真实!	(157)
何时使用虚拟函数: \$ 64000 问题	(160)
虚拟解除程序: 真正的还是虚拟的破坏?	(160)
使用虚拟函数的规则: 调节关系	(163)
小结	(163)
第八章 类层次设计: 善于利用分层	(165)
设计类层次: 生成类行动的指南!	(165)
低效类设计: 未合理安排的数组	(166)
高效类层次: 节约!	(173)
类 myBasicArray	(179)
类 myArray	(179)
类 myOrderedArray	(179)
小结	(180)
第九章 多重继承性: OOP 类型家族成员	(181)
声明多重继承性: OOP 集成	(181)
不同世系的派生: 挽救 911!	(184)
样例输出	(185)
把你介绍给类	(185)
全局	(186)
类 Random	(187)
类 Distance	(188)
类 Rescue	(189)
函数 main()	(192)
共同世系的派生: 冷静计算!	(193)

样例输出	(194)
全局	(195)
类 NCScale01	(195)
类 NCScale100	(199)
类 NCScale200	(201)
类 NCScale120	(203)
函数 main()	(204)
小结	(205)
第十章 容器: 优于多重继承性吗?	(207)
容器: 它和有限破坏相关吗?	(207)
使用被包含类的规则: 使用它们或丢失它们	(209)
改进营救例子	(209)
全局	(211)
类 Random	(212)
类 Distance	(212)
类 Rescue	(214)
函数 main()	(217)
在容器中使用朋友类: 患难之交才是真朋友	(218)
另一个营救程序: 加入朋友关系	(219)
全局	(220)
类 Random	(221)
类 Distance	(222)
类 Rescue	(223)
函数 main()	(226)
小结	(227)
第三部分 高级 OOP 的特性	(229)
第十一章 抽象类: Trendsetters	(231)
抽象类仅仅是个哗众取宠的名词吗?	(231)
声明抽象类: 我确实想这么做!	(232)
声明一个有一个抽象类的简单层次: 孤单的一个!	(233)
类 AbstractArray	(238)
类 MemArray	(238)
类 DiskArray	(239)
函数 main()	(239)
声明一个有多个抽象类的层次: 越多越让人愉快	(240)
小结	(246)
第十二章 模板: 类的工厂	(247)
类模板是什么?	(247)
声明类模板: 设置规格	(248)
定义类模板的成员函数	(249)

例示类模板:一个类诞生了!	(250)
类模板 myArray	(257)
类 myString	(258)
函数 main()	(258)
声明类模板的层次:如果一个类模板不够的话	(259)
小结	(264)
第十三章 异常:运行错误,OOP 风格	(265)
异常是什么?(又是一个 Buzzword 吗?)	(265)
异常类:早已准备好承受责骂!	(266)
介绍标准异常	(266)
抛掷一个异常:让我们看看你的程序是多么粗暴	(267)
try 块:错误是人犯的!	(268)
catch 从句:原谅是神圣的!	(268)
想玩一玩 catch 吗?	(270)
类 Exception1	(274)
类 Exception2	(274)
类 Exception3	(274)
函数 main()	(274)
嵌套 try-catch 块:求助于计划 B	(275)
重新抛掷一个异常:是否抛球由你决定	(277)
函数 solver()	(279)
函数 main()	(279)
联系异常和函数	(279)
小结	(282)
第十四章 朋友类和嵌套类:另外的类关系	(283)
朋友类是什么?难道是敌人类的对立面吗?	(283)
声明朋友类:聚会时刻!	(283)
使用朋友类:并非如你想象!	(284)
类 myArray	(289)
类 myMatrix	(290)
函数 main()	(291)
嵌套类是什么?	(291)
声明嵌套类	(292)
使用嵌套类:在 C++ 中,此行为在道德上也 OK	(293)
嵌套类 Array	(296)
类 myFixedStack	(296)
函数 main()	(297)
小结	(297)
第四部分 十准则集粹	(299)
第十五章 十个类元件的清单(加 1)	(301)

父类·····	(301)
朋友类·····	(302)
类片断·····	(302)
构造函数·····	(302)
解除函数·····	(303)
静态成员·····	(303)
常数(const)成员函数·····	(304)
抽象成员函数·····	(304)
虚拟成员函数·····	(304)
运算符=·····	(305)
友元函数和友运算符·····	(305)
第十六章 虚拟成员函数的十个命令 ·····	(307)
何时声明虚拟成员函数·····	(307)
虚拟成员函数的语法·····	(307)
使用虚拟函数来覆盖非虚拟函数·····	(308)
覆盖虚拟成员函数·····	(308)
过载虚拟成员函数·····	(309)
继承虚拟成员函数·····	(309)
调用被继承的虚拟成员函数·····	(310)
速度和虚拟成员函数·····	(310)
虚拟成员函数和抽象类·····	(310)
虚拟成员函数和类模板·····	(311)
第十七章 十个最普通的 I/O 流元件 ·····	(313)
对象 cout 和运算符 <<·····	(313)
对象 cin 和运算符 >>·····	(313)
对象 cin 和成员函数 getline()·····	(313)
对象 cerr 和运算符 <<·····	(314)
操作符 dec·····	(314)
操作符 hex·····	(314)
操作符 oct·····	(314)
操作符 setfill()·····	(315)
操作符 setprecision()·····	(315)
操作符 setw()·····	(315)
词汇表·····	(317)

前言

面向对象程序设计方法(OOP)不仅受到程序员的喜欢,也受到用户的青睐。过去的十六年相继出现了OOP语言,如 SmallTalk, Eiffel 和 C++。即便是非OOP语言,如 Ada 和 COBOL,也正在把面向对象的设计特征作为最新的美国国家标准协会(ANSI)的一部分。由此看来 OOP 的影响力是不言而喻的。

关于本书的说明

本书将把你带入一个使用 C++ 的面向对象程序设计世界。暂且假定你熟悉 C++。如果不熟悉的话,我建议你还是找一本 Stephen R. Davis (IDG Books Worldwide 公司出版)写的 C++ For Dummies 看看。那本书把 C++ 讲解得很清楚,所以我不需要再多说了。相反,我着重强调 C++ 的面向对象方面。因此,我希望你已准备好了一个 C++ 编译器(它能在 DOS, Windows, UNIX 或 Macintosh 等操作系统下运行)并且知道怎样编写控制台应用程序(使用标准输入输出流的一类程序)。

关于本书的代码

本书含有类、成员函数等的基本语法。下面的这个小例子可以表明我的风格:

```
class className
{
    public;
        // public members
    protected;
        // protected members
};
```

本例表明非保留关键字(in this case, *className*)以斜体方式显示,而保留字(如 class, public 及 protected)则以标准字符方式出现。方括号表示可选项,当我讲解某一特定 C++ 特性的基本语法时,我会经常使用一些总体性字眼构成的注释来讲解此语法的含义。

本书的举例部分包含了我所提供的程序。在那些例子中,我使用黑体字用以强调用户所提供的任意输入。

有时你会在某些代码行的结尾看到一个曲线箭头,那就意味着此行在你的屏幕上是以一行的方式而出现。我们只是在一行中没有足够的空间来容纳所有的字符。

本书的编排顺序

本书分四部分共十七章,另附一个引导你在面向对象程序设计世界里艰难跋涉的词汇表。每章都有一些程序设计的例子,这些例子会告诉你不同面向对象编程特性的细节。很典型的情况是,程序员喜欢那些长得让人喘不过气来而且缺乏具体细节的文本的列表。从某种意义上讲,一个程序清单就像一幅画——顶得上千言万语。此外,我曾经被一些不准确的代码片断,尤其是在技术手册中害惨过多次。这就是我为什么喜欢给你一个能运行且完整的程序清单。我有时也用代码片断,但是我把它控制到最低限度。

第一部分 OOP 的基本概念

在第一部分中(包含一到三章)你将了解到面向对象程序设计、类以及对象的一些基本特征。

第一章将会给你展现面向对象设计的支柱:类、对象、方法、继承性以及多态性。本章将给你介绍一些面向对象程序设计的基本概念(以及生词)。

第二章着重讲解类及声明方法。本章考查类的所有组成部分(数据成员及成员函数)。此外本章还讲解成员函数的不同作用。

第三章讲解一个对象的生命周期并且告诉你怎样对其进行声明、初始化、控制以及最终的消除。

第二部分 舒适阶段的巡游

第二部分包含从第四章到第十章的内容,它为您提供面向对象的程序设计。

第四章主要讲解静态数据成员和成员函数,并且告诉你它和类在概念上是如何相关,而不是与类实例相关。本章提供了使用静态成员的例子。这些例子讲解了实例计数、共同错误管理以及共享信息。

第五章涉及到类设计并且着重强调内聚和耦合——与成员函数设计相关的两个标准。本章还会给出一个错误设计类的实例以及其改进版本。

第六章讲解 C++ 类里面的继承性。这章着重强调单一继承性并且告诉你怎样声明和使用后代类。此外本章还讲解子类的构造程序以及非公共后代的使用。

第七章讲解多态性——OOP 的特性之一,它能在类分层的成员中提供统一响应。本章还讲解虚拟成员函数是如何支持 C++ 中的多态性的。

第八章讲解类层次设计并且提供优秀设计指导。本章还提供一个非有效类层次例子,并且与它的改进版作比较。

第九章讲解多重继承性,并且告诉你怎样声明使用多重继承性模式的后代类。这一章还给你一个来自于独有父类的后代类例子,此外本章还提供一个来自于同一血统的父类的后代类的例子。

第十章比较容器和多重继承性。这一章告诉你在不使用一个多重继承性模式的情况下容器如何和不同的类相关。这一章含有一个编程的例子,此例子使用容器并把它和第九章中的多重继承性的例子相比较。

第三部分 高级 OOP 的特性

这一部分包括从第十一章到第十四章的内容,我给你提供一些面向对象的高级编程主题。

第十一章讲解抽象类并且告诉你怎样声明它们。本章提供了单一抽象类层次和多重抽象类层次的例子。

第十二章提供了模板,它是 C++ 较新的一个特性。这一章讲解模板的好处,并且告诉你怎样声明一个类模板,怎样定义一个类模板的成员函数,怎样具体声明一个类模板。此外,这一章还告诉你怎样生成一个类模板的层次。

第十三章讲解异常(运行错误)并且告诉你 C++ 是如何利用类和对象来管理异常的。这一章讲解标准异常并且提供了 throw、try 以及 catch 关键字。这一章还讲解嵌套 try-catch 块,重新投入一个异常以及结合函数和异常。

第十四章讲解相关类的其他方法,主要是用友元和嵌套类。这一章讲解使用友元的好处并且佐以实证。这一章也讲解嵌套类及其好处,同时还提供友元的嵌套类的例子。

第四部分 十准则集粹

为了和他人的风格保持一致... For Dummies 之类的书(或是 David Letterman?),第四部分是十准则集粹。

第十五章提供了类声明的核对列表。你也可以使用手头的列表以确保你没有遗忘珍贵的类组成元件。

第十六章讲解虚拟成员函数的十个命令(Featuring Charlton Heston as Moses)。

第十七章检查十个共同流 I/O 对象、成员函数以及控制器。这些组成元件一般都包含在 C++ 控制应用程序中。我在本书就提到过一个。

最后的,而且是相当重要的,那就是供你寻找难倒你的难词与技术词汇的词汇表。

本书中所用的图标

纵观全书,我使用了下列图标以指明特殊信息。



此图标指出了—个技术便条,它给我将要讲解的编程特性提供了额外注释。



此图标提醒你—个关键的编程特性。



此图标强调—个能加强你的代码的编程技巧。



此图标警告你有一些坏家伙,还帮你省了一些麻烦(给我省去了一些可恨的邮件)。

敢问路在何方?

现在我们已到了这里。你的面向对象程序设计世界之行即将开始,如果你感到压力太大,不要不好意思休息,你也许只是想要重新熟悉一些基本 C++ 词汇时才备上一本 C++ 的入门书(如 C++ For Dummies)。记住这是你的书,所以尽管放心大胆地勾重点,在空白处写批语,或在程序清单中作注释。

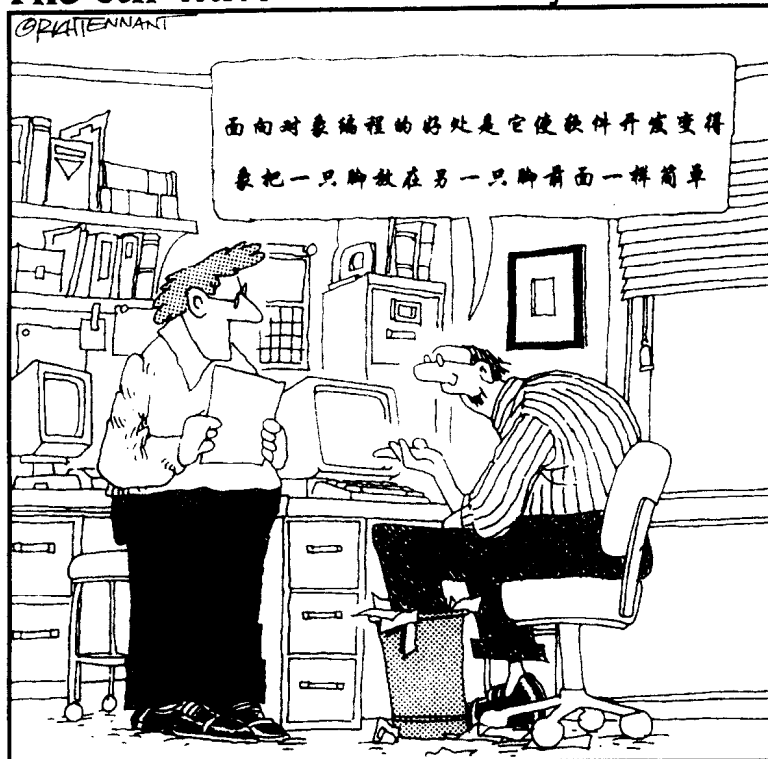
这就是书中写的东西!我希望这本书为你敞开通向面向对象程序设计的大门。祝你快乐!

第一部分

OOP 的基本概念

The 5th Wave

By Rich Tennant



在这部分中……

你首先接触到的是面向对象程序设计、类以及对象的基本概念。第一章讲解面向对象程序设计的支柱,第二章和第三章讲解与面向对象编程相关的类和对象。