Mc Graw Hill Education

世界著名计算机教材精选

C++面向对象程序设计（第4版）

E Balagurusamy 著

高 峰 等译

OBJECT ORIENTED PROGRAMMING WITH C++

Fourth Edition

清华大学出版社

E Balagurusamy
Object Oriented Programming with C++ ,Fourth Edition
EISBN：978-0-07-066907-9

图书在版编目(CIP)数据

C++ 面向对象程序设计(第 4 版)/(印)巴拉古路萨米著；高峰等译．—北京：清华大学出版社，2010.6
(世界著名计算机教材精选)
书名原文：Object Oriented Programming with C++ ,Fourth Edition
ISBN 978-7-302-22202-6

Ⅰ．①C…　Ⅱ．①巴…　②高…　Ⅲ．①C 语言－程序设计－教材　Ⅳ．①TP312

中国版本图书馆 CIP 数据核字(2010)第 035892 号

责任编辑：龙啟铭
责任校对：徐俊伟
责任印制：何　芊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京市人民文学印刷厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　　**印　张**：33　　**字　数**：826 千字
版　　次：2010 年 6 月第 1 版　　**印　次**：2010 年 6 月第 1 次印刷
印　　数：1～3000
定　　价：49.00 元

产品编号：034370-01

译 者 序

20 世纪 80 年代，贝尔实验室的 Bjarne Stroustrup 在 C 语言的基础上，加入了面向对象的一些特性，从而发明了一种“带类的 C 语言”，亦即今日在软件工程和 IT 业界大行其道的 C++ 编程语言。20 多年来，C++ 语言历尽改进，其标准分别在 1998 年和 2003 年制定并更新。

作为一门面向对象的程序开发语言，C++ 继承了 C 语言的运行高效，并具有和 Java 等语言类似的编写效率。无论是计算密集型的服务端后台开发，还是具有复杂业务背景的大型项目，C++ 语言都有其用武之地。由于在软件开发方面的优势和地位，C++ 语言业已成为当代计算机程序员和计算机专业科班学生的必修课之一。

国内外目前的C++ 书籍可谓汗牛充栋，不可计数。粗略来看，大体上可分速成类和经典大部头两类，前者往往号称 21 天学会C++，实际上所获却极为有限，而后者的学习曲线一般较为陡峭。令新手在自学过程中，往往心生畏惧。所以，对于编程还未入门的初学者而言，怎么入手学习C++ 经常是一个令人头痛的问题。

本书正是面向C++ 初学者的优秀读本，本书的作者 E. Balagurusamy 博士是印度国内 IT 培训教育方面最著名的专家，他所著的本书内容深入浅出，语言通俗，特别适合学习C++ 面向对象编程的初学者。本书成书于C++ 标准制定之后，其介绍的C++ 编程完全遵循标准。书中示例丰富，图示精彩，对C++ 面向对象特性的探讨全面又不显繁琐，简约而又不失华彩。对于编程新手来说，书中的叙述和程序例子能大大降低理解C++ 编程的难度，而从编写程序入手的教学风格，更容易帮助建立学习者的信心和兴趣。

另外，本书内容也涵盖了面向对象软件系统的介绍，并附有丰富的C++ 自测题库，供自学者自我测试，以检验对书中概念的掌握程度。附录中还专门有一章介绍了两个完整的 C++ 小型项目的开发，结合本书的学习，对示例项目代码的研读，相信会为读者带来更好的编程体验和学习效果。

虽然在本书中，有些地方将C++ 语言同 C 语言相作比较，但其目的主要是为了兼顾有 C 语言基础的学习者。如果读者尚无 C 语言的基础，直接学习本书也无障碍。

在本书的翻译过程中，陈金慧对部分书稿进行了仔细校对，并提出了很多宝贵的修改意见。在此对她的辛勤劳动表示感谢。在翻译过程中，我们力求忠实、准确地反映原著的风格和内容。但鉴于译者水平和时间所限，难免会有错误之处，敬请广大读者不吝指正。

高峰

2010 年 5 月

于北京 清华园

目　录

第 1 章　面向对象编程原理

1.1　软件危机

软件技术的发展日新月异，相关的工具和技术不断推陈出新。因此，软件工程师和业界被迫持续地寻求更新的软件设计和开发方案。随着软件系统越来越复杂，业界的竞争越来越激烈，软件设计便变得至为关键。IT 软件业快速发展，危机也相应而生，为应付软件危机，开发人员必须面临如下问题：

- 在系统设计中，如何表示现实实体的问题
- 如何设计具有开放接口的系统
- 如何保证模块的可重用性和可扩展性
- 如何开发能够适应将来任何变化的模块
- 如何提高软件开发效率，减少软件开发成本
- 如何提高软件质量
- 如何管理开发计划
- 如何让软件开发过程实现产业化

许多软件都没有开发完毕，或者很少使用，或者发布时还有重要隐患。图 1.1 显示了 20 世纪 70 年代实施的美国国防软件工程的最终结果。其中大约 50%的软件产品根本就没有最终发行，而在发行的软件中，有 1/3 从未有人用过。有意思的是，只有 2%的软件产品是发行后即可使用，而无须对软件作任何更改。这说明，软件业在发行软件产品方面，实在是口碑欠佳。

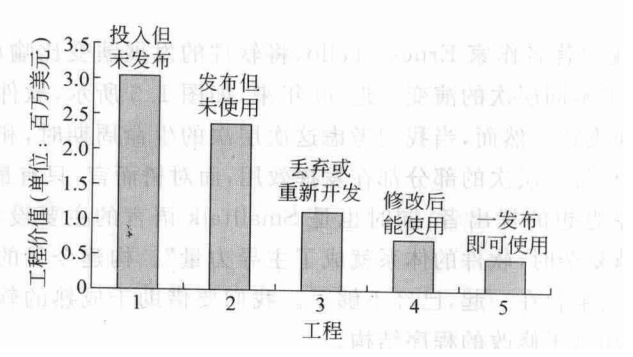

图 1.1　美国国防软件工程的状况（此数据由美国政府提供）

用户需求的变化一直是个大问题。另一份研究(图 1.2)表明,由于用户变更了需求和数据格式,50%的软件系统需要修改。这就说明,在业务变幻莫测的世界里,总是在变化的,所以系统必须得具有良好的适应性,以容忍日后的变化。

其他关于软件实现的研究报告表明,软件在发行和实现前,必须经过仔细的质量评估。在关键的评估中,涉及的一些软件质量包括如下:

1. 正确性
2. 可维护性
3. 可重用性
4. 开放性和互用性
5. 可移植性
6. 安全性
7. 完整性
8. 用户界面友好程度

选用一些合适的软件工具,可以帮助解决软件在这些方面的质量问题。

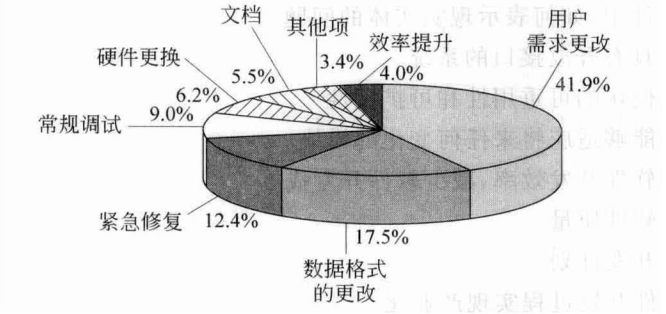

图 1.2 软件维护成本一览

1.2 软件的发展

人工智能领域的著名作家 Ernest Tello,将软件的发展演变比喻成树木的生长。和树一样,软件也经历了不同层次的演变。近 50 年来,如图 1.3 所示,软件逐层发展,每一层都是前一层的提升和飞跃。然而,当我们考虑这次层次的生命周期时,和树的类比就失效了。因为在软件系统中,每一层次的部分都在发挥效用,而对树而言,只有最外层是有用的。

作为面向对象思想的提出者,同时也是 Smalltalk 语言的主要设计者,Alan Kay 曾说过,“当程序越来越复杂时,软件的体系就成了主导力量”。构建今日的复杂软件,将一连串的语句,程序和模块组合在一起,已经不够了。我们要借助于成熟的软件构建技术,以及易于理解、易于实现和易于修改的程序结构。

自计算机发明以来,人们尝试过了许多编程方案,这其中包括模块化编程、自顶向下编程、自底向上编程和结构化编程。每一个方案的提出都出于同样的动机:应付日益增加的

软件复杂性，以保持程序的可靠性和可维护性。20 年间，这些编程方案在程序员群中，一度都非常流行。

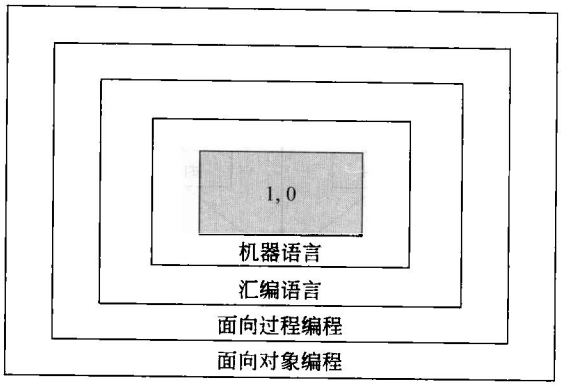

图 1.3　计算机软件的层次

而当 C 语言发明后，结构化编程大行其道，并在 20 世纪 80 年代成为了主流的编程技术。借助于结构化编程这一强大工具，程序员们可以相当容易地编写适度复杂的程序。然而，当程序规模变大后，对于无错、易于维护、可重用性高这些软件质量目标，即便结构化编程也可能无法达到。

为了克服传统编程方法的不足，面向对象编程(OOP)应运而生。面向对象编程方法利用了结构化编程的优点，并加入了一些新的特性，是一种组织和开发程序的新方案，它与具体的编程语言无关。尽管如此，不是所有编程语言都容易适用 OOP 的思想。

1.3　面向过程编程窥探

传统编程，使用诸如 COBOL、FORTRAN 和 C 语言等高级语言，通常是面向过程编程(POP)。在面向过程编程中，问题被分解成一系列的子处理任务，比如读取数据、计算和打印。为了实现这些任务，程序员需要编写函数，函数也是面向过程编程中的首要关注焦点。面向过程编程的典型程序结构如图 1.4 所示。在编程解决问题时，需要使用逐层分解任务的技术，以确定待完成的任务。

在面向过程编程中，基本上只需编写计算机执行的代码指令(或者功能)，并将这些指令组织成一个个函数。通常，我们使用流程图来组织各个代码功能，功能到功能之间即为流程。这里我们主要关注函数的开发，而被不同函数使用的数据还未提及。数据会被如何操作？当函数执行时，数据如何受到影响？

在一个多函数程序中，很多重要的数据项都被声明成全局变量，这样就可被所有函数访问。而每一函数亦有其局部数据。图 1.5 显示的即为面向过程编程中数据和函数的关系。

全局数据易受函数的不当访问而被更改。在大型程序中，很难确定数据被哪些函数使用。一旦更改了外部的数据结构，我们就需要修改所有访问该数据的函数，这将可能增加程

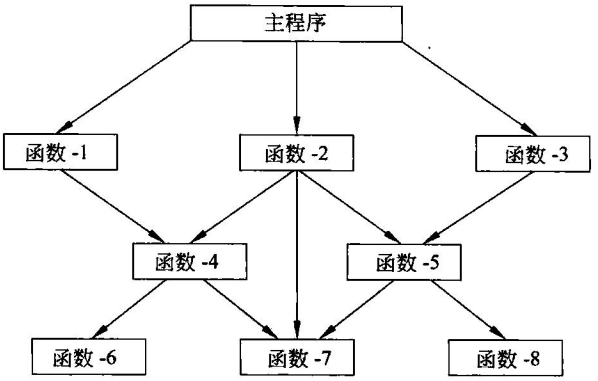

图 1.4 面向过程编程的典型程序结构

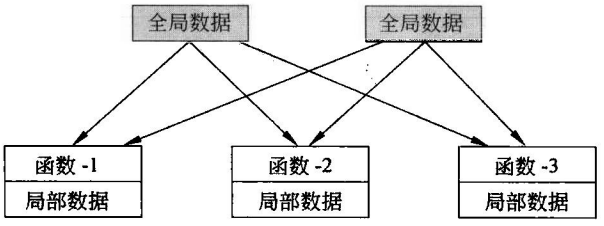

图 1.5 面向过程编程中数据和函数的关系

序的隐患。

面向过程编程还有一个严重不足,即它并没有对现实世界很好的建模。这是因为函数都是基于功能实现的,它们并不对应于问题要素。

面向过程编程的一些特点如下:

- 强调功能(算法)实现
- 大程序分解成函数等小模块
- 大多数函数共享全局数据
- 系统中数据可在函数间公开传递
- 函数加工处理数据,使其形式发生变化
- 程序设计采用自顶向下方案

1.4 面向对象编程方案

面向对象编程方案的兴起,很主要的动机是为了移除面向过程编程中的诟病。在面向对象编程中,数据成为程序的重要部分,而不再在系统中自由流转。面向对象编程将数据和函数捆绑得更为紧密,以防函数外的意外修改。它将问题分解成一些称为对象的实体,并围绕它们构建数据和函数。图 1.6 显示了面向对象编程中数据和函数的组织结构。其中对象中的数据只能由对象相关的函数访问。当然,对象的函数也可以调用其他对象的函数。

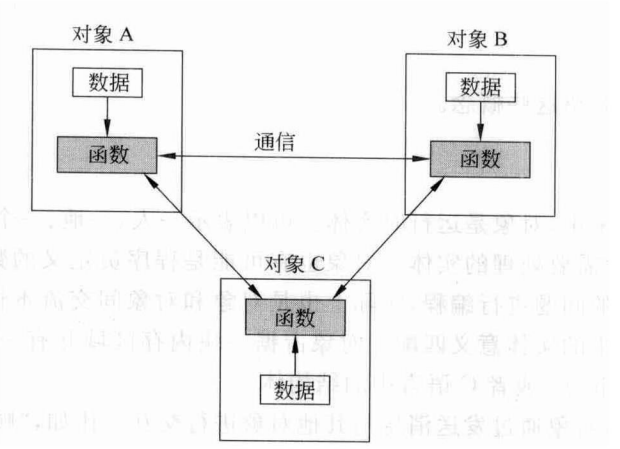

图 1.6　面向对象编程中数据和函数的组织结构

面向对象编程的一些核心特性如下：

- 更关注数据，而不是过程
- 程序分解成实体对象
- 设计数据结构以表示对象
- 作用于对象的函数也放于数据结构中
- 数据隐藏，不可由外部函数调用
- 对象间通过函数通信
- 必要时，容易添加新的数据和函数
- 程序设计采用自底向上方案

面向对象编程是最近才提出的编程方案，对它的理解不免因人而异。为了进一步讨论，有必要给出面向对象编程明确的定义。定义如下："作为一种编程方案，面向对象编程通过创建的分块内存区，提供了一种模块化程序的方式，内存块中存放数据和函数，它们可用作创建所需模块副本的模板"。所以，一个对象可视为开辟的一块计算机内存，其中存放了数据和访问数据的操作。由于内存分块之间互不影响，所以不经修改，对象就可在不同的程序中使用。

1.5　面向对象编程的基本概念

这里有必要理解一些概念，它们在面向对象编程中经常出现。它们包括：

- 对象
- 类
- 数据抽象和封装
- 继承
- 多态

- 动态绑定
- 消息传递

这一节将分别介绍这些概念。

对象

在面向对象系统中,对象是运行时实体。可以表示一人、一地、一个银行账号、一份数据表格,或者其他程序需要处理的实体。对象也有可能是程序员定义的数据,比如向量、时间和列表。对现实中的问题进行编程,实际上也是对象和对象间交流本性的分析。程序中的对象应尽量和现实中的实体意义匹配。对象占据一块内存区域并有一内存地址,对象相当于 Pascal 语言中的记录,或者 C 语言中的结构体。

当程序执行时,对象通过发送消息与其他对象进行交互。比如,"顾客"和"账号"是程序中的两个对象,顾客对象可能发送一条消息给账户对象,要求查看账户余额。每一对象包含数据,以及操作数据的代码。对象在交互时,无须知道其他对象数据或操作的细节,只要知道被接受和回复的消息类型就够了。虽然对象的表示因人而异,但是如图 1.7 所示的两种符号在面向对象分析和设计中颇为流行。

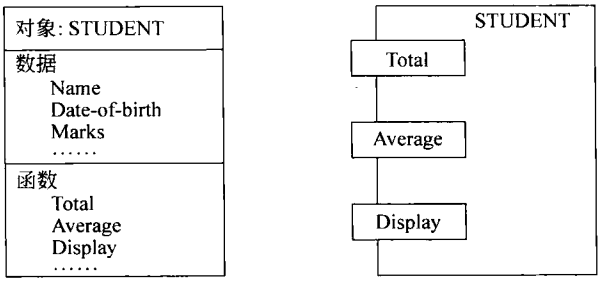

图 1.7 对象的两种表示方式

类

我们刚刚提到,对象包含数据和操作数据的代码。使用类的概念,对象的数据和操作代码可被定义成一个数据类型。实际上,对象即为某类的变量。一旦定义了一个类,我们就可以创建该类的任何数量的对象。每一对象都对应于一个类。所以类实际上是所有同类对象的集合体。比如,芒果、苹果和桔子都是水果类的成员。类是由程序员自定义的数据类型,和程序语言的内置数据类型一样被操作。创建一个类对象时,其语法和 C 语言声明整型变量时别无二致。比如 Fruit 被定义为类,则下列语句

```
Fruit mango;
```

将创建 Fruit 类对象 mango。

数据抽象和封装

将数据和函数合二为一(成为类)的过程,即为封装。数据封装是类最震撼人心的特性。