

21 世纪高等学校规划教材

C++ 程序设计

刘丽华 主 编



中国电力出版社
www.infopower.com.cn

21 世纪高等学校规划

TP312/2559

2007

C++ 程序设计

刘丽华 主 编
崔玲玲 张秀芝 副主编
张佃龙 参 编



中国电力出版社
www.infopower.com.cn

内容提要

本书由从事 C 与 C++ 程序设计课程教学多年的老师编写,共分 11 章,主要包括: C++ 语言概述、基本数据类型与表达式、控制语句、类和对象、继承和多态、特殊成员函数、运算符重载、I/O 流、模板、异常处理和面向对象的程序设计,各章均附有与内容相对应的习题,强化应用实践,是学生学习 C++ 面向对象程序设计的理想教材。本书注重基础,强调实用,培养能力,在满足教学大纲和计算机等级考试要求的同时,还专门开辟一章“面向对象程序设计”来对 C++ 语言的不易掌握之处和灵活性加以重点说明。同时,传授实用程序的编制方法及大程序的设计方法。

本书不仅可以作为各类本专科、高职高专学生的教材,也可以作为社会上相关培训班的教材,还可供广大程序开发爱好者自学之用。

图书在版编目(CIP)数据

C++ 程序设计 / 刘丽华主编. —北京: 中国电力出版社, 2007

21 世纪高等学校规划教材

ISBN 978-7-5083-5666-2

I. C… II. 刘… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 103211 号

丛 书 名: 21 世纪高等学校规划教材

书 名: C++ 程序设计

出版发行: 中国电力出版社

地 址: 北京市三里河路 6 号

邮政编码: 100044

电 话: (010) 68362602

传 真: (010) 68316497, 88383619

服务电话: (010) 58383411

传 真: (010) 58383267

E-mail: infopower@cepp.com.cn

印 刷: 北京市同江印刷厂

开本尺寸: 185mm×260mm 印 张: 14.5 字 数: 330 千字

书 号: ISBN 978-7-5083-5666-2

版 次: 2007 年 8 月北京第 1 版

印 次: 2007 年 8 月第 1 次印刷

印 数: 0001—4000 册

定 价: 22.00 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

面向对象方法的产生，是计算机科学发展的需要。与传统的程序设计方法相比，面向对象的程序设计具有抽象、封装和多态性等特征。现今，“面向对象”不仅仅作为一种语言，更作为一种方法论贯穿于软件设计的各个阶段。面向对象的技术在系统程序设计、数据库及多媒体应用等领域都得到广泛应用。专家们预测，面向对象的程序设计思想将会主导今后程序设计语言的发展。

C++语言是为适应 20 世纪 90 年代开发和维护复杂的应用软件的需要而开发的。它的目标是为程序员的程序开发提供优良的程序设计环境，以便能产生模块化程度高、重用性和可维护性好的程序。同时，C++语言非常强调代码的有效性和紧凑性，它是程序员的语言，允许程序员决定如何实现特定的操作。因此，C++语言已经在各个领域得到了广泛应用，尤其适用于中等和大型的程序开发项目。许多事实已经证明，C++应用于 C 语言曾经使用过的所有场合，其效果比 C 语言要好得多，从开发时间、开发费用到形成的软件的可重用性、可扩充性、可维护性和可靠性等方面，都显示出 C++的优越性。在程序设计方法方面，C++既支持传统的面向过程的程序设计方法，也支持新的面向对象的程序设计方法，因此，C++是一种混合语言。由于 C++的这种特性，就使得 C++保持与 C 语言兼容，从而使许多 C 语言代码不经修改就可以为 C++所用，用 C 语言编写的众多的库函数和实用软件也可以用于 C++中，从而方便了 C 语言用户向 C++过渡。不过，用 C++编写的程序的可读性更好，代码结构更为合理，可以直接在程序中映射问题空间的结构。

确实，C 程序员仅需学习 C++语言的特征，就可以用 C++编写程序。这并不意味着程序员要先学习 C 语言，然后学习 C++。相反，没有学过 C 语言的程序员，学习 C++更容易。因为已经对 C 语言熟悉的程序员，很容易穿着 C++的鞋子，却走着 C 语言的老路。没有学过 C 语言的人，反而不受 C 语言框框的影响。这是因为 C 语言是面向过程的程序设计语言，而 C++是面向对象的程序设计语言。虽然编译器是一个，但我们不是学习使用编译器，而是学习 C++面向对象的编程方法。因此，学习 C++的关键是面向对象的思维方法。

根据高等学校的教学计划，学生已经学习过 C 语言，所以不再重复 C 语言的知识。因为本书的重点是强调面向对象的程序设计方法，涉及少量 C 语言的知识，所以也可以作为直接学习 C++的教材。为了掌握这些基本概念及设计方法，我们还单独开辟“面向对象程序设计”一章，结合实例对面向对象设计方法进行综合归纳，并给出几个实用的例子以增强理解。全书共分 11 章。第 1 章是 C++语言概述，重点是介绍面向对象的基本概念，并从 C++的观点出发，介绍许多 C 语言所没有的概念。第 2 章是基本数据类型与表达式，重点是介绍 C++中基本数据类型和表达式。第 3 章是 C++的控制语句，重点是介绍 C++兼容面向结构化程序设计的方法。第 4 章是类和对象，重点是介绍面向对象的程序设计知识及定义和使用类的方法。第 5 章是继承和多态，介绍单一继承、多重继承、多态性、虚函数、虚函数的多态性及虚析构函数。第 6 章是特殊成员函数，介绍各种常用成员函数的特征。第 7 章是运算符重载，介

绍运算符重载的必要性及增量、转换和赋值运算符的重载。第8章是I/O流，介绍流类库及流应用。第9章是模板，简要介绍函数模板、类模板及模板与继承的关系。第10章是异常处理，介绍流的错误和处理。第11章是面向对象程序设计，本章先介绍面向对象的设计方法，然后讨论对类的整体认识、类的组织、类和函数的设计与说明及继承和面向对象设计等问题，最后给出几个设计实例以加深理解。

本书主要面向本专科的学生，注重基础、强调实用、培养能力，在满足教学大纲和计算机等级考试要求的同时，还专门开辟一章“面向对象程序设计”，对C++语言的不易掌握之处和灵活性加以重点说明。同时，传授实用程序的编制方法及大程序的设计方法，另外，每章配有习题，使学生巩固所学知识，强化他们的实践应用。

各章最后附有精心挑选的按题型分类的习题，包括多选题及编程题，既加深对概念的理解又加强了编程能力的训练。

本书虽然作为各类本科、高职高专的教材，但也适合作为社会上各种培训班的教材，还可供广大程序开发爱好者自学之用。

本书编写过程中参考了大量书籍及文献资料，在此向被引用资料的作者及给予帮助的所有同志再次表示感谢。

限于作者水平，加之时间仓促，不妥之处在所难免，恳请同行及读者指正。

刘丽华

2007年6月

目 录

前 言

第 1 章 C++语言概述	1
1.1 C++语言的起源和特点	1
1.2 面向对象基础	2
1.3 C++对面向对象程序设计的支持	3
1.4 C++语言与 C 语言的关系	4
1.5 面向对象程序设计的意义	6
1.6 简单的 C++程序实例	8
1.7 C++程序的编辑、编译和运行	10
小结	12
习题 1	13
第 2 章 基本数据类型与表达式	14
2.1 标识符	14
2.2 数据类型的初步知识	15
2.3 常量	16
2.4 变量	18
2.5 运算符	19
2.6 表达式	30
2.7 运算符优先级和结合性	31
2.8 求值次序与副作用	31
小结	33
习题 2	33
第 3 章 控制语句	37
3.1 条件语句	37
3.2 多分支的选择语句	39
3.3 循环语句	41
3.4 跳转语句	47
3.5 程序举例	48
3.6 对输入输出的初步认识	52
3.7 关于注释的一些说明	62
3.8 预处理	63
小结	66
习题 3	66

第4章 类和对象	70
4.1 定义类	70
4.2 使用类和对象	72
4.3 内联的成员函数	76
4.4 成员函数的重载及其默认参数	78
4.5 this 指针	79
4.6 结构和联合	79
4.7 有关类的其他知识	81
4.8 构造函数与析构函数	85
小结	101
习题4	101
第5章 继承和多态	104
5.1 类的继承	104
5.2 单一继承	104
5.3 多重继承	106
5.4 多态性和虚函数	108
5.5 类的应用示例	122
小结	126
习题5	126
第6章 特殊成员函数	132
6.1 静态成员	132
6.2 友元函数	135
6.3 const 对象和 volatile 对象	138
6.4 转换函数	141
6.5 内联函数	143
6.6 函数重载	144
6.7 默认参数的函数	149
6.8 指向类成员的指针	151
6.9 数组与类	154
小结	156
习题6	156
第7章 运算符重载	159
7.1 运算符重载	159
7.2 如何重载运算符	160
7.3 值返回与引用返回	163
7.4 运算符作成员函数	164
7.5 重载增量运算符	167
7.6 转换运算符	170
7.7 赋值运算符	172

小结	174
习题 7	174
第 8 章 I/O 流	176
8.1 I/O 标准流类	176
8.2 文件流类	177
8.3 串流类	179
8.4 控制符	180
8.5 使用 I/O 成员函数	183
小结	186
习题 8	186
第 9 章 模板	188
9.1 模板的概念	188
9.2 函数模板	189
9.3 类模板	192
小结	194
习题 9	194
第 10 章 异常处理	195
10.1 异常的概念	195
10.2 异常的实现	196
10.3 异常的规则	198
10.4 异常处理机制	200
10.5 使用异常的方法	202
小结	204
习题 10	204
第 11 章 面向对象程序设计	205
11.1 面向对象的设计	205
11.2 改变思维方式更好地理解面向对象的程序设计	208
11.3 类设计的几点注意	220
11.4 继承和面向对象设计	221
小结	222
参考文献	223

第 1 章 C++语言概述

计算机软件开发一直被两大难题所困扰：一是如何超越程序复杂性障碍；二是如何在计算机系统中自然地表示客观世界，即对象模型。面向对象的程序设计理念是软件工程学中的结构化程序设计、模块化、数据抽象、信息隐藏、知识表示、并行处理等各种概念的积累与发展，是解决上述两大难题的 20 世纪 90 年代最有希望、最有前途的方法。

面向对象程序设计是软件开发方法的一场革命，它代表了新颖的计算机程序设计的思维方法。该方法与通常结构程序设计十分不同，它支持一种概念，即旨在使计算机问题的求解更接近人的思维活动，使人们能够利用 C++ 语言充分挖掘硬件的潜在能力，在不增加开销的前提下，提供更强有力的软件开发工具。

面向对象程序设计是软件系统的设计与实现的新方法。这种新方法是增加软件的可扩充性和可重用性，来提高程序员的生产能力，并能控制维护软件的复杂性和软件维护的开销。当使用面向对象程序设计方法时，软件开发的设计阶段更加紧密地与实现阶段相联系。在软件设计与实现的诸多方法中，面向对象方法在实践中是极具发展潜力的方法，并且在各个应用领域中都获得巨大成功。

从目前现状来看，C++ 和面向对象程序设计，不仅在尖端技术应用领域（如金融和通信）已立稳脚跟，而且 C++ 已逐渐地为企业开发人员所接受。C++ 产品可以给用户一个很好的起点，用户可在此基础上按照接近人的思维活动，按不同的对象类向前推进。

面向对象方法包含了分析、设计和实现三个方面，这部分是当今软件开发最弱的部分，面向对象方法对软件系统开发起着关键作用。本书不仅要在面向对象方法方面，而且还要在面向对象模型和设计方面加以介绍。面向对象模型和设计能够更好地加速对问题需求的了解，使得设计更加简洁清晰。特别是分析和设计过程中产生的高质量的产品，能极大地减少在开发后期发现的错误，并能显著地改善系统质量。

C++ 是目前最流行的面向对象程序设计语言。它在 C 语言的基础上进行了改进和扩充，增加了面向对象程序设计的功能，更适合编制复杂的大型软件。这一章我们将引入面向对象的概念，并通过一个简单的 C++ 程序来加深对面向对象程序设计方法的理解。

1.1 C++语言的起源和特点

顾名思义，C++ 语言是从 C 语言继承来的，但这种继承只是表现在语句形式、模块化程序设计等方面。如果从更重要的方面——概念和思想方面来看，C++ 源于早期的 SIMULA 语言，因为 C++ 语言的最大特征是支持“面向对象的程序设计”（面向对象的程序设计的概念见 1.2 节）。SIMULA 语言被广泛地用于系统仿真，设计它的主要目的是模仿现实世界的真实个体，而使用的主要手段是构造计算机领域的对象来表述现实的客体。由于 SIMULA 语言的应用领域并不十分广阔，更重要的一点是它缺乏强有力的开发工具支持，因而并没有得到很

高的重视。随后推出的另外一种面向对象语言 Smalltalk 也没有取得太大的成功，很多人认为它没有提供给自己足够的灵活性和如同 C 或 BASIC 语言那样丰富的功能，关键原因还在于它和人们早已得心应手的语言并不兼容。比如说，一个 C 程序员可能会对它的新特性退避三舍，因为他对 C 的特性是十分熟悉的，同时 C 的确是功能强大的，大多数人不愿放弃这些。

C++的产生正是为了解开这样的一个“情结”。面对越来越大、越来越复杂的系统，使用 C 语言编写程序已经感到力不从心了，但 C 语言作为应用领域最为广泛的程序设计语言之一，又不能轻易放弃。必须有一种面向对象的程序设计语言，它对 C 语言有很高的兼容性，使得 C 程序员只需在原有的知识基础上进行一定的扩充，就能够方便地进行面向对象的程序设计。

1980 年起，Bell 实验室的 Bjarne Stroustrup 博士及其同事开始为这个目标对 C 语言进行改进和扩充。由于这种被扩充和改进的 C 语言的大量特性与类（class）相关，它最初被开发者称为“带类的 C”。但很快人们就认识到这个称呼太片面了，这个“扩展了的 C”不仅以标准化 C（ANSI C）作为子集保留了 C 语言的全部精华，同时又吸收了 SIMULA 67、ALGOL 68 和 BCPL 语言的许多特性，它已远远超过了 C 语言。随着这个语言的广泛应用和在各个领域取得成果的增多，它给程序设计带来了全新理念，表现出远大前景，因此它的开发者将 C++ 这一名字赋予了它。

与过去的面向过程的程序设计语言相比，C++ 的最大特征在于它是面向对象的程序设计语言。所谓对象是现实世界中的实体，例如，桌子、电视接收机、张三等等。具有共同行为和特征的实体的集合，可以被归纳成一类，因此每个对象都是属于某个类的对象，例如，人是一个类，而每个具体的人则是人这个类中的一个对象。面向对象的程序设计是程序设计的一种新思想，该思想认为程序是相互联系的离散对象的集合。面向对象的程序设计语言即是支持这种思想的程序设计语言。

1.2 面向对象基础

在面向对象的程序设计方法出现之前，占据主流的是结构化程序设计方法。对于复杂的问题，结构化程序设计采用模块化、自顶向下逐步求精的设计原则，因此结构化的程序往往清晰、易读。典型的结构化程序设计语言有 C 语言、PASCAL 语言等，著名的 UNIX 操作系统的大部分代码就是用 C 语言编写的。

随着软件技术的发展，需要开发的系统越来越复杂。人们逐渐发现，对于大型软件系统来说，如果采用结构化的设计方法，设计、编程、测试和维护等工作都非常困难，而且有许多问题是结构化设计自身无法解决的。在这种背景下，产生了面向对象的方法，而面向对象的程序设计语言（Object Oriented Programming Language, OOP）也应运而生。

那么，什么是面向对象的方法呢？

面向对象方法的出发点和基本原则，就是使分析、设计和实现一个系统的方法尽可能地接近认识一个系统的方法。形象一点来说，就是使得描述问题的问题空间和解决问题的方法空间在结构上尽可能地一致。这样说可能太抽象，但随着对 C++ 语言学习的深入，会逐步体会到这一点。

下面介绍面向对象方法的几个重要概念，作为后面学习 C++ 语言的基础。对象（Object）：

是由信息和对它进行处理的描述所组成的包，其结构如图 1-1 所示。

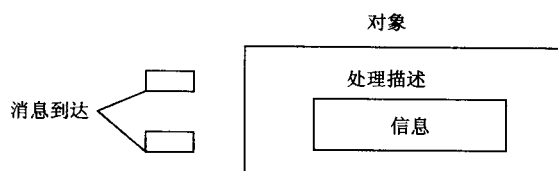


图 1-1 对象的图解

消息 (Message): 是对某个对象进行处理的说明。

方法 (Method): 是类似于过程的一个实体，是对对象接受了某一消息后所采取的一系列操作的描述。

类 (Class): 是对具有共同特征的对象描述。

类的封装 (Encapsulation): 封装是把一类对象的状态（用数据表示）和方法（用函数来表示）封闭起来，装入对象体中，形成一个能动的实体。OOPL 的封装机制模仿现实生活中的封装技术，把一类对象的数据和函数封闭起来，并提供访问它们的机制。外界只有调用对象的共有成员函数才能和对象交换信息，这样就达到了封装的目的。

类的继承 (Inheritance): 是指新的类继承原有类的全部数据、函数和访问机制，并可以增添新的数据、函数和访问机制。这样产生的新类叫子类或派生类，原来的类叫父类或基类，这种产生新类的方法叫类的派生，也叫类的继承。如“汽车”是一个类，“轿车”就是它的子类，而某辆实际的小轿车就是这个子类的一个对象。

多态性 (Polymorphism): 是指相似而实质不同的操作可以有相同的名称。例如，“和”的操作，可以是“整数和”，也可以是“矢量和”，在 C++ 中，这两种“和”的操作都可以简单地称为“和”。C++ 的多态性使得 C++ 与人的思维习惯更趋一致。用 C++ 编制的程序也更方便人的阅读。面向对象的方法还有许多特征和概念，如虚函数等，我们会在后面具体介绍。

1.3 C++对面向对象程序设计的支持

C 语言产生于 1972 年，最早用来编写 UNIX 操作系统，经过完善与改进后，发展成为了一种通用的计算机语言。1983 年进行标准化 (ANSI C) 后，其发展更为迅速，几乎所有操作系统都提供对 C 语言的支持。

1985 年，AT&T 的 Bell 实验室在 C 语言的基础上，吸收了 OOPL 的特点，形成了面向对象的程序设计语言 C++。

C++ 是一种灵活高效、可移植的面向对象程序设计语言。C++ 诞生以后，发展极其迅速，很多公司都研制了自己的 C++ 版本，并推出了多种 C++ 的集成开发环境。

C++ 支持基本的面向对象的概念，如对象、类、方法、消息、子类、继承性以及多态性等。表 1-1 给出了 C++ 对这些概念的命名约定。

C++ 程序设计语言在 C 语言的基础上扩充了类、内联函数、运算符重载、变量类型、引用和自由存储管理运算符等语法。我们在后面通过简单的例子来了解 C++ 语言的特点，在第 2 章、第 3 章再系统地学习 C++ 语言的语法与编程方法。

表 1-1 C++的概念及命名约定

面向对象的概念	C++的命名的约定
对象	对象
类	类
方法	成员函数
实例变量	成员
消息	函数调用
子类	派生类/子类
继承	派生/继承

1.4 C++语言与 C 语言的关系

C 语言也诞生在 AT&T 的 Bell 实验室，1972 年由 Dennis Ritchie 在 B 语言的基础上开发出来的一个高级语言，今天 C 语言的使用已遍及计算机的各个领域。

C 语言有以下几个显著的特点：

(1) 它是一种结构化语言，要求一个程序由众多的函数组成，程序的逻辑结构由顺序结构、选择结构和循环结构三种基本结构组成，适宜于大型程序的模块化设计。

(2) 它可以部分取代汇编语言，同时具有很高的可移植性，这使得 C 语言程序在保证支持不同硬件环境的前提下，具有较高的代码效率。

(3) 它提供了丰富的数据类型和运算，具有较强的数据表达能力，因而在许多不同的场合得到广泛应用。

总之，C 语言反映了设计者追求高效、灵活，支持模块化程序设计，从而支持大规模软件开发的愿望。

C++语言保留了 C 语言设计者的良好愿望，并使得 C 语言语句成为 C++语言的一个子集。一般，用 C 语言编写的程序可直接在 C++编译器下编译。

1.4.1 C++语言与 C 语言的主要区别

首先，C++提出了类（class）的概念。类是数据和函数的集合，数据用来描述类所属对象的状态，函数用来描述此类对象的行为。例如，大学生代表在大学读书的一类人，即大学生是个类，每个具体的大学生都是这个类中的对象。大学生这个类中的数据可以是学生的姓名、性别、年龄、学校、专业或入学时间等等，描述此类对象的行为可以是入学、改换专业或毕业等。

C 语言中的结构只是数据的集合，这种结构也可在 C++语言中使用。C++语言沿用了 C 语言中的结构，概念上没有变化。不同的是，C++语言将 C 语言中的“结构”概念扩充成近似于上述“类”的概念，即 C++语言中的结构既可以有数据，也可以有函数。下列关键字是 C++语言新增的：class、private、protected、public、this、new、delete、friend、inline、virtual、volatile。

1.4.2 C++语言与 C 语言的细小区别

为较完整地讲述 C++语言与 C 语言的关系，在此预先介绍它们之间的细微差别，读者可

以在学习完本书各章节后，再来阅读这部分。

(1) C++语言在保留 C 语言原有注释方式的同时，增加了行注释。以“//”起始，以换行符结束的部分是行注释。

(2) **const** 关键字。用这个关键字修饰的标识符为常量。它的引入可以替代 C 语言中的符号常量定义。比较下面两个语句：

```
define Number 1
const Number=1
```

它们的功能相同，但后一语句在编译时进行了类型检查。

(3) 说明结构、联合和枚举变量时不必在结构类型名、联合类型名和枚举类型名前加关键字 **struct**、**union**、**enum**。例如：

```
/*C 语言的说明*/
struct semployee a0,b0;
union uemployee a1,b1;
enum eemployee a2,b2;
// C++语言的说明
semployee a0,b0;
uemployee a1,b1;
eemployee a2,b2;
```

(4) 变量的说明可以放在程序的任一位置上，例如：

```
for (int I =0;I <100;I++)
```

(5) 提供了作用域运算符“::”，当其中一个变量被一个局部变量阻挡时，运用作用域运算符仍可以操作该全局变量。

(6) 标准输入输出一般不再使用 C 语言的 **printf** 和 **scanf**，而使用三个标准 I/O 流，它们是：**cout**（与标准输出设备相连）加 **cin**（与标准输入设备相连）和 **cerr**（与标准错误输出设备相连）。在计算机上，各输入输出设备一般分别为显示器、键盘和显示器。“<<”和“>>”分别是定义为流的插入和提取操作，例如：

```
cout<<"welcome";
cin>>a;
cerr<<"There is an error";
// 向显示器输出"welcome"
// 从键盘取得数据至 a
// 在显示器上立即显示错误信息
```

自从 1983 年 AT&T 的 Bell 实验室推出了它的 C++标准之后，仅仅经历了 9~10 年，C++ 的应用已遍及计算机技术的各个领域，并且取得了很大的成功。面向对象的概念越来越多地被人们所应用，例如，新一代的具有革命性的微机和工作站操作系统 Winndows NT 就大量运用了这一概念。

面对越来越复杂的系统，越来越大型的软件，面向对象的程序设计从一个新的角度使得问题空间和解题空间保持更为有效的一致性，使得计算机软件更加易于理解。作为一种成功的面向对象程序设计语言，C++成为人们对于客观世界进行概括和准确描述的有效手段，C++使得软件开发者能方便地仿真客观世界。所以我们可以想见，今天的 C++正如当年 C 语言一

样, 会给软件设计带来一次巨大进步, 成为新的更为有效工具。

需要指出的是, C++只是作为面向对象的程序设计方法的一种实现, 还有其他的程序设计语言也能支持面向对象程序设计, 例如, 面向对象的 PASCAL 语言及前面已经提到的 SMALL TALK 语言等。因为面向对象方法学的提出是程序设计的革命, 很多的专家学者都在为这种新思想的更完美实现而努力着。不仅仅是软件产生巨大的变革, 计算机硬件也受到这个思想的影响而发生了变化。

1.5 面向对象程序设计的意义

在前面我们提到, C++语言的最大特点是支持面向对象的程序设计。在 C++语言基本为读者了解之后, 我们来看看面向对象程序设计的意义, 希望能帮助读者接受一种良好的程序设计思想, 从而使程序的分析、设计和实现一致而有效, 使 C++语言能得以更好的应用。

面向对象的程序设计, 是继模块化的面向过程的程序设计思想之后发展起来的表现出强大生命力的新的程序设计思想。它的产生和发展, 大大提高了软件的开发质量。衡量一个软件的质量高低, 主要从以下几个方面考虑:

- (1) 易读性;
- (2) 易修改性, 易维护性;
- (3) 易扩展性, 代码可重用性;
- (4) 可移植性。

一个好的程序设计思想, 首先应是提高软件开发的质量, 节省软件开发时间。

一般来说, 一种程序设计思想, 总是与一定的程序设计语言相互促进、共同发展的。面向对象的程序设计的好处, 应从程序设计思想的发展历史中寻找。计算机诞生之初, 硬件昂贵, 内存少, 运算速度低, 因此, 这个时期对于软件的要求主要是, 尽可能少的空间占用和时间消耗。关于软件的这种要求便成了当时的程序设计思想。这个时期的程序设计语言, 主要是汇编语言。汇编语言的易读性只比机器语言强一些, 其修改性和维护性都很差, 代码重用率低, 可移植性也很差, 因为它是与具体的计算机硬件直接对应的。这个时期产生的某些计算机高级语言, 主要也只是相对于汇编语言, 谋求在程序的易读性和可移植性方面有所进步。

随着计算机硬件水平的显著提高, 内存空间的运算速度已不成为困难。此时, 程序设计思想已不受计算机硬件的限制, 相继产生了面向过程的程序设计思想和面向对象的程序设计思想, 这两种程序设计思想都包含了结构化和模块化的程序设计思想。

面向过程的程序设计认为, 程序是过程的集合。具体说来, 这种程序设计思想类似于图论和运筹学中的网络工程——程序也可以被看成网络图。

一般网络图中的一个箭头表示一个过程, 每个过程完成一定的操作, 有一定的输入和输出, 各过程之间相互联系, 协同运作, 完成软件预定的功能。这里的过程都被看作是“算法+数据结构”, 每个过程都是采用某种算法对特定结构的数据的操作, 并且, 几个过程可以对同一种结构的数据进行操作。

在面向过程的程序设计思想指导下开发的程序, 弊端主要有以下几个方面:

(1) 程序的各个部分不便于修改。原因有: ①各过程联系紧密, 任何一个过程的修改, 都将可能引起其相关联过程的修改; ②一个过程的修改部可能涉及数据结构的改变, 这将影

响对此数据操作的所有过程。

(2) 不便于系统功能的修改和扩展。这主要是因为网络工程图结构本身就是基于功能的。从设计伊始,面向过程的程序设计所关心的,就是程序所要完成的功能(所要达到的输出)。一个程序的功能又被分解成一系列的子功能,它们分别对应于各个过程。这种基于功能的做法,导致了系统功能扩展的困难。因为系统功能是被分解的,改变的系统功能或新增的系统功能可能需要重新分解,这种分解可能涉及相当多的过程,导致许多过程的重写,或是需要删除和增添一些过程。

(3) 与上面第(2)点相似,代码的重用率低,从而程序的寿命短,不能适应不断发展的用户需要。

(4) 程序的开发周期不够快。产生这个弊端的原因在于,这种设计思想与人们认识世界的习惯不一致。因为人们对于世界的认识习惯是基于对象的,在认识事物的过程中,已有许多关于各种对象的描述。面向过程的程序设计不能利用已有的这些描述,须重新认识系统,将系统理解成个网络,这增加了软件系统的分析和设计时间。

面向对象的程序设计之所以流行,其原因是:这种思想较为成功地克服了面向过程的程序设计的上述弊端,较大地提高了软件质量,缩短了软件的开发周期。

面向对象的程序设计把程序看作是离散对象的集合,这里的对象是人们所理解的现实世界中的对象的抽象,它具有以下特征:

- (1) 对象是具有一定属性和行为的实体;
- (2) 对象是可以分门别类的;
- (3) 对象是相对封闭的,对象的内部情况对外界不透明;
- (4) 对象及对象间的相互联系构成系统;
- (5) 对象间的相互联系依赖于消息,对象既可向外界发出消息的消息;
- (6) 一个对象的属性和行为可从别的对象继承和发展得到,也可接收和响应外界;
- (7) 对象和由对象所构成的系统都是可变的,对象的属性和运动过程都可能变化,但对于由对象所构成的系统,对象是相对稳定的。

面向对象的程序设计,要尊重现实世界中对象的特征,特别是对对象的封闭性、继承性,以及对象间的联系依赖于消息这三个方面,从而在这种思想指导下开发的程序具备了以下几个方面的优点:

(1) 程序是易于修改的。因为许多修改都可局限于一个对象的内部或一类对象的内部,这种修改不再像面向过程的程序设计那样,涉及联系广泛的过程、数据及其结构;易于修改和增加系统的功能,可以方便地通过改变对象间的消息传递,或是继承和发展新的对象来进行;程序的代码重用率人为提高,从而程序的寿命得以延长。特别地,面向对象的程序设计思想,与人们认识世界的习惯一致,从而有效地节省了软件的开发时间。

(2) 从前面关于 C++语言的讲述可知 C++由于引入“类”与“对象”,并且在类的基础上引入继承机制,从而取得了支持面向对象的程序设计思想的能力。按照面向对象的程序设计思想,使用 C++语言,可望较快地得到高质量的程序。

(3) C++语言的多态性等,使得用 C++语言开发的程序,更加方便人们的阅读和理解,更加方便程序的修改和扩充。

关于 C++语言进行面向对象的程序设计方面的成功例子,读者可以参阅有关书籍。

1.6 简单的 C++ 程序实例

C++是一个功能强大、方便灵活的程序开发语言，适合于编制复杂的大型软件系统。但若想熟练掌握它，则需要由浅入深、循序渐进地进行学习。

下面先给出两个简单的 C++ 程序代码，然后仔细解释了每行代码的含义，通过这两个例子初步介绍 C++ 的编程思想。

程序清单 1.1 计算圆的面积和周长。

```
// ex1-1.cpp
#include <iostream.h>
const double pi=3.14;
double calculate_area(int r)
{return pi*r*r;
}
double calculate_per(int r)
{return 2*pi*r*r
}
void main()
{
int input_radius;
cout<<"请输入圆的半径:";
cin>>input_radius;
cout<<" 圆的面积为:"<<calculate_area(input_radius)<<"\n";
cout<<" 圆的周长为:"<<calculate_per(input _radius) <<"\n";
}
```

程序运行时显示：

请输入圆的半径：

如果从键盘输入 3 并按回车键，程序会继续显示：

圆的面积为：28.26

圆的周长为：18.84

Press any key to continue

这时，程序运行已结束，按任意键可继续下面的操作。

程序中，如果首行有“//”的语句，均为 C++ 的注释行，程序执行时将会忽略注释行。

第 2 行的 `#include <iostream.h>` 是预处理包含指令，其中的 `iostream.h` 是 C++ 提供的标准类库，它提供了输入输出流 `cin` 和 `cout`，以及输入输出运算符 `>>` 和 `<<` 的定义。

第 3 行说明了一个浮点型符号常量 `pi`，该常量的固定值为 3.14。

第 4~6 行是函数 `calculate_area` 的定义，该函数可以根据输入参数 `r` 计算圆的面积。

第 7~9 行是函数 `calculate_per` 的定义，它可以根据输入参数 `r` 计算圆的周长。

第 10~11 行定义了主函数 `main()`，其中 `main` 是函数名，它是程序执行的起点。

第 13 行是一个输出语句，`cout<<` 表示从显示器输出后面的“请输入圆的半径：”的内容。

第 14 行是一个输入语句，其中的 `cin>>` 表示从键盘输入一个整数给变量 `input_radius`。

第 15 行是一个输出语句，先输出字符串“圆的面积为：”，再输出由 `calculate_area` 计算

出的半径为 r 的圆的面积，最后输出 “\n”，表示换行。

第16行也是一个输出语句，先输出字符串“圆的周长为：”，再输出由 `calculate_per` 计算出的半径为 r 的圆的周长，最后输出换行符。

现在再解释 `cin` 和 `cout` 的用法：`cin` 称作字符流输入，`cout` 称作字符流输出，这两个函数都是在文件 `iostream.h` 中定义的。若想在程序中使用 `cin` 和 `cout`，就必须用预处理包含指令 `#include` 将文件 `iostream.h` 嵌入到程序中。

运算符 `<<` 和 `>>` 的原意是表示算术左移和右移，在 `iostream.h` 文件中被重新定义（在面向对象中称作运算符重载），作为输入输出运算符使用。

`cout` 中使用 `<<` 运算符，它将控制语句中的字符串或变量输出到标准设备上。`<<` 运算符可以在一个语句中使用多次，比如，`cout<<"请输入姓名："<<endl`，该语句表示先输出字符串“请输入姓名：”，再输出 `endl`。`endl` 是换行符号，等效于“\n”。

`cin` 中使用 `>>` 运算符，它可将键盘输入的信息存于变量中，当遇到空格便立即停止输入。一个 `cin` 语句中可以多次使用 `>>` 符号，比如：`cin>>ai>>a2`

注意：当使用 `cin` 输入语句时，与标准 C 语言的输入语句不同，它输入信息的变量可以是任意类型的，而且不要求带有地址指示符 `&`。

程序清单 1.2 使用类封装计算圆的面积和周长。

```
// ex1-2.cpp

#include <iostream.h>
const float pi=3.14;
class circle
{
public:
    int r;
    double calculate_area( )
    {
        return pi*r*r;
    }
    double calculate_per( )
    {
        return 2*pi*r;
    }
};

void main()
{
    circle thecircle;
    cout<<"请输入圆的半径:";
    cin>>thecircle.r;
    cout<<"圆的面积:"<<thecircle.calculate_area( )<<"\n";
    cout<<"圆的周长:"<<thecircle.calculate_per( )<<"\n";
}
```

该程序的运行结果与程序 1.1 完全相同，但是程序结构却大不一样。程序的 4~16 行定义了一个类 `circle`，其中有公共类型的变量 `r` 和两个函数 `calculate_area( )` 和 `calculate_per( )`。主程序中的第1行生成了一个类 `circle` 的对象 `thecircle`，然后将我们输入的半径值存入该对象的 `radius` 变量，再利用该对象的两个函数分别计算圆面积和周长。可能用户还有不清楚的地方，因为这个例子涉及了 C++ 中 OOP 的一些基本概念，这些概念在后面还要详细讨论。