

程序设计方法与优化

Programming: Methodologies and Optimization

覃 征 王志敏 等 编著

Qin Zheng

Wang Zhimin

et al



西安交通大学出版社

XI'AN JIAOTONG UNIVERSITY PRESS

程序设计方法与优化

Programming: Methodologies
and Optimization



编著

王向华
Wang Xianghua

杨利英
Yang Liying

黄 茹
Huang Ru



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容提要

本书系统讲述了计算机程序设计的基本概念、基本方法和常用程序语言的优化设计思想,用大量的程序实例说明了常用程序设计方法的实际应用和编程技巧。全书分 10 章,以三个部分介绍了程序设计的基础知识、基本方法及其优化方法。第一部分概要介绍了程序设计方法的发展、程序设计的一般方法和表示方法,并描述了算法的概念和图灵机模型;第二部分结合具体程序实例详细讲述了结构化程序设计方法、面向对象程序设计方法、组件化程序设计方法、递归程序设计方法、嵌入式程序设计方法和程序的正确性证明;第三部分介绍了程序计算复杂度的分析方法,对程序设计进行了定量的表示,并举例说明了 C/C++ 程序、Java 程序、ASP 程序、Prolog 逻辑程序、32 位汇编指令常用的优化内容、原则与方法。

本书可作为高等院校程序设计方法课程的教科书,也可以作为从事计算机程序设计的研究人员和从事软件系统设计、开发及应用工作的相关技术人员的参考书。

图书在版编目(CIP)数据

程序设计方法与优化 / 覃征,王志敏等编著. —西安:西安交通大学出版社,2004. 1
ISBN 7-5605-1801-X

I. 程… II. ①覃… ②王… III. 程序设计:最优设计
IV. TP311

中国版本图书馆 CIP 数据核字(2003) 第 119313 号

书 名	程序设计方法与优化
编 著	覃 征 王志敏等
责任编辑	贺峰涛 屈晓燕
文字编辑	葛赵青 钱次余
出版发行	西安交通大学出版社
地 址	西安市兴庆南路 25 号(邮编:710049)
网 址	http://unit.xjtu.edu.cn/unit/jtupress
电 话	(029)82668357 82667874(发行部) (029)82668315 82669096(总编办)
电子信箱	eibooks@163.com
印 刷	陕西宝石兰印务有限责任公司
版 次	2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷
开 本	727mm×960mm 1/16
印 张	24
字 数	442 千字
书 号	ISBN 7-5605-1801-X/TP·362
定 价	36.00 元

版权所有 侵权必究

前 言

电子计算机的发展是 20 世纪科学发展史上最伟大的事件之一。自从 1946 年世界上第一台电子计算机 ENIAC 诞生以来,在短短的 50 多年里,计算机科学迅猛发展,计算机的应用已经渗透到社会的各个领域,成为当今信息社会的最显著的特征。之所以如此,其中一个很重要的原因就是计算机软件系统的高速发展。软件系统发展的关键在于程序设计方法的发展。

程序设计方法研究程序设计的基本思想、原理、技术和优化,使程序代码能有效地描述用于解决特定问题的算法。程序设计方法已成为计算机科学中内涵丰富而深刻的一个重要分支,涉及程序理论、控制结构、开发技术、运行环境和工程规范标准等内容。程序设计方法的研究是计算机科学中的一个新兴领域。近年来,这一领域发展非常迅速,同时也取得了很多研究成果。为了介绍这一领域的一些基本思想方法和实际应用,我们在总结多年研究成果的基础上,撰写完成本书。本书着重讨论程序设计方法中最基本和最成熟的方面,并在一定程度上反映国内外的当前工作。

与国内外同类书比较,本书系统性强、层次分明、通俗易懂、便于自学,并结合作者的理解和体会来阐述基本概念和特定问题,同时引入近年来在程序设计领域出现的新的思想和方法。另外,本书没有采用统一的语言来描述程序,这样可以使读者接触到更多的程序控制结构和设计风格,有利于读者阅读其他相关专著。

本书系统讲述了计算机程序设计的基本概念、基本方法和常用程序语言的优化设计思想,用大量的程序实例说明了常用程序设计方法的实际应用和编程技巧。本书中的完整程序均在 PC 机上调试通过,希望能对读者起到抛砖引玉的作用。全书共分 10 章,以三个部分介绍了程序设计的基础知识、基本方法及其优化方法。

第一部分:基础篇(第 1,2 章)

该部分概要介绍了程序设计方法的发展、程序设计的一般方法和表示方法,并描述了程序算法的概念和图灵机模型。

第二部分:方法篇(第 3~8 章)

该部分结合具体程序实例详细讲述了结构化程序设计方法、面向对

象程序设计方法、组件化程序设计方法、递归程序设计方法、嵌入式程序设计方法和程序的正确性证明。

第三部分:优化篇(第9,10章)

这一部分介绍了程序计算复杂度的分析方法,对程序设计进行了定量的表示,并举例说明了C/C++程序、Java程序、ASP程序、Prolog逻辑程序、32位汇编指令常用的优化内容、原则与方法。

本书的所有内容都经过了作者的精心策划和安排。在本书的编写过程中,得到了西安交通大学电子商务研究所和计算机系很多教授和青年教师的支持和指教,同时也得到西安交通大学出版社的大力支持,我们在此表示衷心的感谢。在编写本书的过程中,参考了大量的中外文献,作者对这些文献著作者表示真诚的谢意。由于本书所涉及的内容广,加之程序设计方法的发展非常迅速,限于作者的水平与时间,难免存在错误和不妥之处,恳请专家和广大读者批评指正。

作者

2003年6月

目 录

第一部分 基础篇

第 1 章 绪论	(3)
1.1 程序设计方法的发展	(3)
1.2 程序设计的一般方法	(9)
1.2.1 程序设计语言简介	(9)
1.2.2 三种基本的程序结构	(10)
1.2.3 程序设计的基本方法要素	(11)
1.2.4 程序设计风格	(15)
1.3 程序设计的表示方法	(18)
1.3.1 程序流程图	(18)
1.3.2 判定表	(20)
1.3.3 过程设计语言(PDL)	(22)
小结	(27)
第 2 章 程序算法与图灵机模型	
2.1 算法概念	(29)
2.2 图灵机模型	(31)
2.2.1 图灵机概念	(31)
2.2.2 二进位码的数据表示	(39)
2.2.3 非自然数的表示	(43)
2.3 通用图灵机	(44)
2.4 希尔伯特问题的不可解性	(48)
小结	(53)

第二部分 方法篇

第 3 章 结构化程序设计方法	
3.1 结构化程序设计的基本思想	(57)

3.1.1 结构化程序设计的概念与标准结构·····	(57)
3.1.2 结构化程序设计的判别·····	(60)
3.1.3 结构化程序设计的步骤与原理·····	(61)
3.2 逐步求精的方法·····	(62)
3.3 改进的 N-S 图 ·····	(66)
3.4 非结构化程序到结构化程序的转化·····	(70)
3.4.1 非结构化程序转化为结构化程序的一般方法·····	(70)
3.4.2 非结构化程序转化为结构化程序的实例·····	(72)
小结 ·····	(75)

第4章 面向对象程序设计方法

4.1 面向对象程序设计的基本思想·····	(76)
4.1.1 面向对象程序设计的概述·····	(76)
4.1.2 面向对象方法的理论基础·····	(82)
4.1.3 面向对象程序设计的方法与步骤·····	(86)
4.2 面向对象程序设计中的继承机制·····	(91)
4.3 面向对象程序设计中的多态性·····	(99)
4.3.1 多态性的实现方式·····	(99)
4.3.2 多态性在程序设计中的应用 ·····	(107)
4.4 面向对象方法与结构化方法的比较 ·····	(116)
4.5 面向对象技术的未来发展 ·····	(121)
小结·····	(125)

第5章 组件化程序设计方法

5.1 组件化程序设计的基本思想 ·····	(126)
5.1.1 组件化程序的标准 ·····	(126)
5.1.2 组件技术与面向对象技术的比较 ·····	(133)
5.1.3 组件化程序的开发方法 ·····	(134)
5.2 CORBA 组件模型 ·····	(136)
5.2.1 CORBA 的相关概念 ·····	(136)
5.2.2 CORBA 中面向对象分析的方法与 Java IDL 程序实例 ·····	(142)
5.2.3 CORBA 技术的新发展 ·····	(150)
5.3 COM 组件对象模型 ·····	(153)

5.3.1 COM 的相关概念	(153)
5.3.2 COM 组件开发方法与程序实例	(157)
5.3.3 DCOM 与 COM+ 技术	(175)
5.4 EJB 组件模型	(177)
5.4.1 EJB 的相关概念	(177)
5.4.2 EJB 组件的开发方法和程序实例	(181)
5.4.3 EJB 和其他技术的比较	(189)
5.5 组件技术与软件体系结构	(192)
小结	(196)

第 6 章 递归程序设计方法

6.1 递归程序设计的基本思想	(198)
6.1.1 递归算法的分析与设计方法	(198)
6.1.2 递归程序的公式化方法与程序实例	(200)
6.1.3 递归方法的应用领域	(204)
6.2 递归方法与树型结构	(206)
6.3 递归方法与栈结构	(209)
6.4 递归算法到非递归算法的变换	(211)
小结	(217)

第 7 章 嵌入式程序设计方法

7.1 嵌入式程序设计的基本思想	(218)
7.1.1 嵌入式系统的定义与特点	(218)
7.1.2 嵌入式程序设计的关键技术和方法	(220)
7.2 嵌入式实时操作系统分析	(226)
7.3 嵌入式 C/C++ 语言程序设计方法	(229)
7.3.1 嵌入式 C 语言程序设计方法	(229)
7.3.2 嵌入式 C++ 语言程序设计方法	(234)
7.4 嵌入式 J2ME 程序设计方法	(239)
7.4.1 嵌入式 J2ME 概述	(239)
7.4.2 MIDP 在仿真器上的程序设计与实例	(244)
小结	(248)

第 8 章 程序的正确性证明

8.1 证明程序正确性的 Floyd 断言方法	(250)
8.1.1 基本概念与定义	(250)
8.1.2 部分正确性的证明	(251)
8.1.3 终止性证明	(253)
8.2 证明程序正确性的 Hoare 公理系统	(255)
8.2.1 Hoare 逻辑	(255)
8.2.2 Hoare 公理系统中的简单公理和证明实例	(256)
8.2.3 循环不变式的建立	(258)
8.3 面向对象程序设计的正确性证明	(260)
8.3.1 面向对象程序验证的特点	(260)
8.3.2 类级正确性	(261)
8.3.3 系统级正确性	(264)
8.3.4 类级正确性与系统级正确性的关系与证明	(265)
8.4 组件程序设计的正确性证明	(266)
8.4.1 组件的描述	(266)
8.4.2 组件的正确性验证	(271)
8.5 用广义数学归纳法证明递归程序的正确性	(273)
8.5.1 普通数学归纳法及应用实例	(274)
8.5.2 广义数学归纳法在递归程序证明中的应用	(275)
小结	(278)

第三部分 优化篇**第 9 章 程序计算复杂度的分析方法**

9.1 程序结构复杂度的分析	(281)
9.1.1 程序结构复杂度的度量与建模	(281)
9.1.2 结构复杂度度量的自动实现	(283)
9.2 程序嵌套结构复杂度的分析	(285)
9.3 递归函数时间复杂度的分析	(289)
9.3.1 渐进算法分析	(289)
9.3.2 递归函数的分析方法	(290)
9.4 简化法则对程序算法时间复杂度的估算	(293)
小结	(295)

第 10 章 程序设计优化的方法

10.1 程序优化的内容与基本方法·····	(296)
10.1.1 程序优化的内容与原则·····	(296)
10.1.2 程序结构优化的基本方法·····	(298)
10.1.3 程序代码优化的基本方法·····	(299)
10.2 算法剖析与程序优化·····	(303)
10.3 常用高级程序语言的优化·····	(305)
10.3.1 C 程序的常用优化方法·····	(305)
10.3.2 C++ 程序的常用优化方法·····	(314)
10.3.3 Java 程序性能的优化方法·····	(329)
10.3.4 ASP 程序性能的优化方法·····	(343)
10.3.5 Prolog 逻辑程序的优化方法·····	(349)
10.4 32 位汇编指令的常用优化方法·····	(358)
小结·····	(365)

主要英文缩写索引**参考文献**

第一部分 基础篇

这一部分概要介绍了程序设计方法的发展、程序设计的一般方法和表示方法,并描述了程序算法的概念和图灵机模型。

本部分内容包括:

第1章 绪论

第2章 程序算法与图灵机模型

第1章 绪论

1.1 程序设计方法的发展

自从1946年世界第一台电子计算机ENIAC诞生以来,在这短短的50多年里,计算机科学得到了迅猛的发展,计算机的应用已经渗透到社会的各个领域。计算机之所以能有如此强大的功能,除了计算机硬件系统的功能日益强大之外,另一个很重要的原因就是计算机的软件系统的高速发展。软件系统发展的关键在于程序设计方法的发展。

所谓程序设计方法,就是使用在计算机上可执行的程序代码来有效地描述解决特定问题算法的过程。程序设计方法经历了如下几个阶段的发展:

(1) 面向计算机的程序设计

计算机系统包括硬件系统和软件系统。硬件是由运算器、控制器、存储器、输入设备和输出设备组成,其中运算器和控制器合称中央处理器,它是计算机的核心,由大规模数字集成电路组成。软件包括了使计算机运行所需的各种程序及有关的文档资料。计算机的工作是由程序来控制的,程序是指令的集合。软件工程师将解决问题的方法、步骤编成由一条条指令组成的程序,输入到计算机中,计算机执行这一指令序列,便可完成预定的任务。所谓指令,就是计算机可识别的命令。计算机的指令相当于人与计算机之间交流的语言。由于计算机的中央处理器是由数字电路组成的,因此它只能识别简单的“0”和“1”组合的二进制指令。人类最早的编程语言是由计算机可以直接识别的二进制指令组成的机器语言。显然机器语言便于计算机识别,但对人类来说却是晦涩难懂。这一阶段,在人类的自然语言与计算机编程语言之间存在着巨大的鸿沟,这一时期的程序设计属于面向计算机的程序设计,设计人员关注的重心是使程序尽可能地被计算机接受并按指令正确地执行,至于计算机的程序能否让人理解并不重要。软件开发的人员只能是少数的软件工程师,因此软件开发的难度大、周期长,而且开发出的软件功能也很简单,界面也不友好,计算机的应用仅限于科学计算。随后出现了汇编语言,它将机器指令映射为一些能读懂的助记符,如ADD, SUB等。此时的汇编语言与人类的自然语言之间的鸿沟略有缩小,但仍与人类的思想相差甚远。因为它的

抽象层次太低,程序员需要考虑大量的机器细节。此时的程序设计仍很注重计算机的硬件系统,它仍属于面向计算机的程序设计。面向计算机的程序设计的基本思想可归纳为:注重机器,逐一执行。

(2) 面向过程的程序设计

随着计算机应用范围的扩大,人们感觉到机器语言和汇编语言的不足,机器语言太注重计算机的硬件,而汇编语言也不太适合人类的思维习惯。因而更接近人类思维习惯的高级语言被设计出来。它撇开了计算机硬件的细节,提高了语言的抽象层次,程序中可采用具有一定含义的数据命名和容易理解的执行语句,这使得在写程序时可以联系到程序所描述的具体事物。20世纪60年代末开始出现的结构化程序设计的思想便是面向过程的程序设计思想的集中表现。结构化程序设计的思想是:自顶向下,逐步求精。其程序结构是按功能划分为若干个基本模块(基本程序),这些模块形成一个树状结构,各模块之间的关系尽可能简单,在功能上相对独立;每一个模块内部均是由顺序、条件、循环三种基本结构组成,其模块化实现的具体方法是使用子程序。结构化程序设计由于采用了模块分化与功能分解,自顶向下、分而治之的方法,因而可将一个较复杂的问题分解为若干个子问题,各子问题分别由不同的人员解决,从而提高了速度,并且便于程序的调试,有利于软件的开发和维护。结构化程序设计的自顶向下、逐步求精的思想可用图1-1的树状结构表示(其中P表示程序, P_1, P_2, P_3 等表示子程序,以此类推; P_{211}, P_{212} 等表示基本程序)。

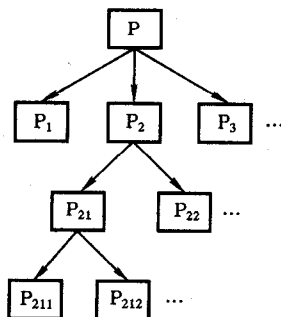


图 1-1 结构化程序的设计思想示意

结构化程序设计思想的核心是功能的分解。当程序员用 C 或 Pascal 语言来设计程序解决一个实际问题时,首先要做的工作就是将一个问题按功能的不同分解成若干个模块,然后根据模块的功能来设计一系列用于存储数据的数据结构,最后编写一些过程(或函数)对这些数据进行操

作。最终的程序就是由这些数据和操作构成的。显然,这种方法将数据结构和操作过程作为两个独立的实体来对待,设计人员编程之前首先考虑如何将功能分解,在每个过程中又要着重安排程序的操作序列,并且程序员在编程的同时又必须时时考虑数据结构。客观世界中的问题是错综复杂和不断变化的,软件开发人员开发的软件往往不是一成不变的。随着社会的发展,用户对软件提出了更多的要求,因此软件的更新日益加快。而面向过程的程序设计中由于数据与操作的分离,降低了程序的可重用性,增加了维护代价。为了克服这一缺点,人们提出了面向对象的程序设计思想。

(3) 面向对象的程序设计

面向对象的程序设计思想是:注重对象,抽象成类。在程序系统中,将客观世界中的事物看成对象,对象是由数据及对数据的操作构成的一个不可分离的整体。对同类型的对象抽象出其共性,形成类。类中的大多数数据,只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系,对象与对象之间通过消息进行联系。为了进一步弄清面向对象的程序设计的思想,我们先来解释几个概念。

对象。对象是系统中描述客观事物的实体,它是由描述其属性结构的数据和定义在数据上的一组操作系统组成的实体。它是数据结构和操作序列的组合物(其中数据描述对象的静态特征,操作描述对象的动态特征)。它是构成系统的一个基本单位。

类。类是一组对象的抽象,是具有相同的属性结构和操作行为的一组对象的集合。类与对象的关系犹如模具与铸件之间的关系。类是用来创建对象实例的样板,它包含所创建对象的属性特性和操作行为的定义。类是一个型,而对象是这个型的一个实例。

封装。封装是面向对象程序设计的一个重要的特性。它是指对象在把数据与操作结合为一个整体时,其数据的表示方式及对数据的操作细节是尽可能地被隐藏的。用户只是通过操作接口对数据进行操作,至于其内部细节则一无所知,这样既能与外部发生联系,又保证了数据的安全性。

继承。继承是面向对象程序设计的又一个重要的特性。它是指特殊类的对象拥有其一般类的全部属性结构的操作行为。如果B类继承了A类,就称A类为父类,B类为子类。在一般情况下,要定义一个新类,只需继承一个父类,再描述一下它与父类的不同之处就行了。这样就大大地减少了设计人员的重复操作,极大地提高了编程效率。

多态性。多态性也是面向对象程序设计的一个重要特性。它是指在一般类中定义的属性或行为,被特殊类继承之后,可以具有不同的数据类型或不同的行为,这使得同一个属性或行为在一般类及各特殊类中具有不同的语义。

面向对象的程序设计的结构特点,一是定义类或继承父类,二是定义各对象并规定它们之间传递消息的规律,三是程序中的一切操作都是通过对象发送消息实现。面向对象的程序设计的过程可以用图 1-2 表示。

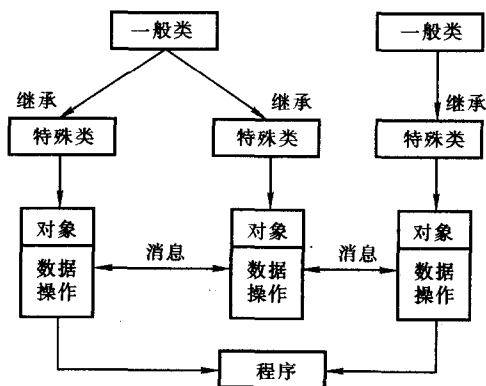


图 1-2 面向对象的程序设计的过程

通过以上讨论可看出,面向对象的程序设计,由于数据与操作封装在对象这个统一体中,使编程人员在编程过程中能够将数据与操作联系在一起,便于程序的修改和调试,并且由于类的继承性使得编程人员可以在可视化的环境中进行组件化的编程,从而使设计人员能够从单调、重复的编程过程中解放出来,去进行创造性的总体设计工作。

(4) 面向组件的程序设计

计算机硬件技术的飞速发展使硬件实现了“即插即用”的功能——只要将硬件的各个部分通过接口有机地连在一起,便可更快更便宜地组装计算机。而大多数软件开发组织仍然把每一个软件开发项目看成是必须完全从头开始的新任务,导致大多数软件项目或是推迟交付时间,或是超出预算。目前,在软件开发领域由日趋成熟的组件技术引起的一场革命正在悄悄兴起。基于组件的开发能改变软件开发过程中的被动局面,使开发者能够到组件市场购买所需组件,组装成应用程序,这将使软件产业发生革命性变化。基于组件的开发与面向对象和客户机/服务器等有着本质的区别,它不只是一种分布式计算的新方法,而是一种广泛的体系结

构。

由于组件自身固有的特性,目前人们对组件这一概念还没有一个统一的定义。下面是关于组件的一些具有代表性的观点:

- 组件是一个独立的可传递的操作的集合。
- 组件是软件开发中一个可替换的软件单元,它封装了设计决策,并作为一个大单元的一部分和其他组件组合起来。
- 组件是由一些对象类组成的物理意义上的包。
- 组件是具有特定功能,能够跨越进程的边界实现网络、语言、应用程序、开发工具和操作系统的“即插即用”的独立的对象。
- 组件在通常意义上是指任何可被分离出来、具有标准化的和可重用性的公开接口的软件(子)系统。

基于组件的开发(Component Based Development,CBD)是一种利用可重用的软件组件构建应用程序的技术。这些组件由三部分组成,即:

- 组件实现的功能的详细说明书;
- 组件是如何工作的实现设计;
- 在指定开发平台上可行的传递方法。

组件的开发工程,主要是一个组装和集成的过程,其基本活动是:

- 收集组件

这一活动是指对从本地或远程资源可获得的组件进行仔细分析,从中发现并收集对自己有用的组件。在这一过程中,可能对组件的特征所知甚少,只是通过组件的名字、参数和所需的操作环境获得一些信息,而组件与其他组件的交互信息可能还隐藏着。通过收集组件获得的组件接口信息,不仅包含有与应用程序的接口信息,而且还包括与其他所有组件进行交互的信息。

- 改善组件质量

这一活动是指在仔细分析组件的文档或详细说明书的基础之上,与组件开发商和用户进行必要的讨论,并在不同的环境设置下运行组件,发现组件中可能存在的不足之处,并加以改进,使之达到改善组件质量的目的。

- 使组件能相互适应

这一活动是指通过编写一些简单的程序作为用户需求与组件产生的相应动作之间的缓冲区,这个缓冲区可给组件提供缺省的信息,减少不希望发生的动作。此缓冲区就像组件之间的“绝缘层”。这是一种包装组件的方法,还有其他的方法,如使用代理、翻译器等。这些方法都可用来解决