

数据库与程序设计基础

SHUJUKU YU CHENGXU SHEJIJICHU

◇根据计算机基础教育课程体系编写

张彦玲 张 卉 主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

TP311.138
316

数据库与程序设计基础

根据计算机基础教育课程体系编写

张彦玲 张 卉 主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书贯彻全国高等院校计算机基础教育研究会制定的“中国高等院校计算机基础教育课程体系”的指导思想,在内容选取上突出基本理论和常用开发工具两个方面。前面章节介绍 Visual Basic 6.0 系统的常用内部控件、控制结构、数组、菜单和对话框设计等内容,叙述全面并配以丰富的实例;随后的章节介绍数据库的基本概念和 Access 数据库的应用,包括数据表、关系、查询、窗体和报表的设计,以及页、宏和 VBA 编程基础。通过学习,读者能充分利用 Visual Basic 的强大功能进行前台设计,用 Access 数据库做后台支持,二者前后呼应、完美结合,是开发应用程序的有效途径。

本书内容丰富、实用性强,既适合文科大学财经类专业使用,也可以作为计算机等级考试的参考书目。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

数据库与程序设计基础/张彦玲,张卉主编. —北京:电子工业出版社,2006.9
ISBN 7-121-02695-3

I. 数... II. ①张...②张... III. ①程序设计—高等学校—教材②关系数据库—数据库管理系统, Access—高等学校—教材 IV. ①TP311.1②TP311.138

中国版本图书馆 CIP 数据核字(2006)第 103714 号

责任编辑:施玉新 syx@phei.com.cn 特约编辑:朱 宇

印 刷:北京市海淀区四季青印刷厂

装 订:三河市万和装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:20.5 字数:525 千字

印 次:2006 年 9 月第 1 次印刷

印 数:5 000 册 定价:28.60 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077;邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

当今社会正步入以计算机和网络技术为代表的信息时代,世界各国对计算机基础教育给予了前所未有的关注。计算机已成为信息社会中必备的工具,操作与使用计算机已经成为一种基本能力,而且与人们的日常工作和生活密不可分。

教育部全国高等院校计算机基础教育研究会于2004年10月推出了“中国高等院校计算机基础教育课程体系”蓝皮书,分门别类地对不同专业的学生提出了计算机基础教育的水平要求与目标。结合我校的实际情况,针对不同专业学生的特点及实际水平,我们于2005年对计算机基础教育教学模式做出了重大改革,实施模块化教学方案,将传统的授课内容按专业科学地分成若干个教学模块,根据学生的专业特点,进行相应的模块组合。为了配合我校的计算机基础教学改革,我们组织多年从事计算机基础教育教学、具有丰富教学经验的一线教师编写了这本《数据库与程序设计基础》教材,该教材适用于财经类专业第二学期的计算机公共基础课程。

本书紧密结合财经院校的专业与教学特点,突出计算机在财经管理中的实际应用。在内容选取上突出基本理论和常用开发工具两个方面。全书共9章,第1~7章介绍程序设计方法、面向对象的程序设计、软件工程的基本概念和Visual Basic 6.0程序设计的实现方法;第8~9章介绍数据库的基本概念与Access数据库的应用。其中VBA程序设计部分,使读者能充分利用Visual Basic的强大功能进行前台设计,用Access数据库做后台支持,二者前后呼应、完美结合,而且较容易掌握,是开发应用程序的有效途径。

财经类院校承担着为各类企事业单位和政府经济管理部门培养中级以上财经管理人才的重要任务,而计算机在财经管理中又起着举足轻重的作用。因此,我们准备采用立体化教学模式,即教材+网上学习+实验,旨在提高学生的学习兴趣、培养学生自主学习的能力。本书在内容安排上突出理论与实际操作相结合、程序设计与数据库应用相结合的特点,以数据录入、数据查询、数据处理、数据分析与数据统计为主线,实例均选用学生感兴趣且与实际应用相关的内容,注重基础理论与编程技巧的合理搭配,使此书成为适合财经管理人员使用的指导教材。

本书在编写过程中得到了华斌教授、王新民教授的亲切关怀和大力支持,得到了计算机公共基础教研室全体教师的鼎力帮助。另外,刘国梁、孙宪、赵庚、郭洁、杨燕青等同志参与了收集资料、书稿校对、例题及习题的调试工作。在此一并表示衷心的感谢!

本书第1、2、6、7章由张卉编写,第3、4章及前言由张彦玲编写,第5、9章由陈钊编写,第8章由金钟、张磊编写。全书由张彦玲和张卉统稿。本书配有多媒体课件和全套实例源程序,需要者可直接与作者联系(E-mail: yanlingzh@126.com)。

由于编写时间仓促,作者水平所限,书中尚有不妥和疏漏之处,敬请同行专家、广大读者批评指正。

编 者
2006年7月

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

第 1 章 程序设计概述	(1)
1.1 程序设计语言的发展	(1)
1.2 程序设计方法与风格	(2)
1.2.1 源程序文档化	(3)
1.2.2 数据说明	(3)
1.2.3 语句结构	(3)
1.2.4 输入和输出	(4)
1.3 结构化程序设计	(4)
1.3.1 结构化程序设计概述	(4)
1.3.2 结构化程序的基本结构	(5)
1.3.3 结构化程序设计的原则和方法	(6)
1.4 面向对象的程序设计	(6)
1.4.1 面向对象的概念	(7)
1.4.2 Visual Basic 面向对象设计方法	(8)
1.5 软件工程的基本概念	(10)
1.5.1 软件的概念、特点和分类	(10)
1.5.2 软件危机和软件工程	(11)
1.5.3 软件生命周期	(11)
1.5.4 软件开发工具和软件开发环境	(11)
习题 1	(12)
第 2 章 Visual Basic 简单程序设计	(13)
2.1 Visual Basic 示例	(13)
2.2 Visual Basic 的基本特点和版本	(14)
2.2.1 基本特点	(14)
2.2.2 版本	(15)
2.3 Visual Basic 集成开发环境	(15)
2.3.1 主窗口	(15)
2.3.2 其他窗口	(17)
2.4 工程管理	(19)
2.4.1 工程的组成	(19)
2.4.2 工程的建立、打开和保存	(21)
2.4.3 工程内文件的添加、保存和删除	(21)
2.5 窗体和基本控件	(22)

2.5.1	窗体设计	(22)
2.5.2	控件	(24)
2.5.3	Visual Basic 常用内部控件简介	(26)
2.5.4	用户界面设计	(34)
2.6	调试	(34)
习题 2		(37)
第 3 章	Visual Basic 程序设计语言基础	(40)
3.1	数据类型	(40)
3.1.1	基本数据类型	(40)
3.1.2	用户自定义类型	(42)
3.2	常量和变量	(43)
3.2.1	常量	(43)
3.2.2	变量	(45)
3.3	表达式和运算符	(46)
3.3.1	算术运算符和算术表达式	(47)
3.3.2	字符串运算符	(49)
3.3.3	关系运算符和关系表达式	(49)
3.3.4	逻辑运算符和逻辑表达式	(50)
3.3.5	日期运算符	(51)
3.3.6	运算符的优先级	(52)
3.4	常用函数	(52)
3.4.1	算术函数	(52)
3.4.2	字符串函数	(53)
3.4.3	数据类型转换函数	(55)
3.4.4	日期和时间函数	(56)
3.4.5	测试函数	(56)
3.4.6	随机函数和随机数语句	(56)
3.4.7	格式函数	(57)
3.5	程序的控制结构	(58)
3.5.1	顺序结构	(59)
3.5.2	选择结构	(67)
3.5.3	循环结构	(76)
3.6	数组	(82)
3.6.1	数组的声明	(83)
3.6.2	静态数组和动态数组	(85)
3.6.3	数组的基本操作	(86)
3.6.4	数组应用实例	(89)
3.6.5	控件数组	(90)
3.7	过程	(95)

3.7.1 Sub 过程	(95)
3.7.2 Function 过程	(99)
3.7.3 过程间参数的传递	(102)
3.7.4 过程的嵌套调用	(104)
3.8 变量和过程的作用域	(105)
3.8.1 模块的划分	(105)
3.8.2 变量的作用域	(106)
3.8.3 过程的作用域	(109)
3.8.4 调用其他模块中的过程	(109)
3.9 多重窗体和多模块程序设计	(111)
3.9.1 设置启动对象	(111)
3.9.2 窗体加载和卸载	(112)
3.9.3 常用事件	(112)
3.9.4 多模块程序设计实例	(113)
习题 3	(115)
第 4 章 常用控件	(119)
4.1 单选按钮和复选框	(119)
4.1.1 单选按钮的常用属性和事件	(119)
4.1.2 单选按钮应用实例	(120)
4.1.3 复选框的常用属性和事件	(122)
4.1.4 复选框应用实例	(122)
4.2 框架	(124)
4.2.1 常用属性和事件	(124)
4.2.2 应用实例	(124)
4.3 列表框和组合框	(126)
4.3.1 常用属性	(126)
4.3.2 常用事件和方法	(128)
4.3.3 列表框应用实例	(129)
4.3.4 组合框应用实例	(131)
4.4 时钟	(132)
4.4.1 常用属性	(132)
4.4.2 事件	(133)
4.4.3 应用实例	(133)
4.5 图形控件	(134)
4.5.1 图片框和图像框	(134)
4.5.2 图片框和图像框的属性、事件和方法	(134)
4.5.3 图像框应用实例	(135)
4.5.4 形状控件和直线控件	(137)
4.5.5 形状控件应用实例	(138)

4.5.6 坐标系统	(139)
4.5.7 图形颜色	(140)
4.5.8 图形方法	(141)
4.5.9 图形方法应用实例	(142)
4.6 滚动条	(143)
4.6.1 常用属性	(143)
4.6.2 常用事件和方法	(144)
4.6.3 应用实例	(144)
4.7 内部控件应用实例	(145)
习题 4	(147)
第 5 章 常用算法	(150)
5.1 算法的概念和特征	(150)
5.2 算法设计要求	(150)
5.3 常用算法	(151)
5.3.1 常用数值算法	(151)
5.3.2 常用非数值算法	(153)
习题 5	(157)
第 6 章 对话框和菜单设计	(159)
6.1 菜单	(159)
6.1.1 下拉式菜单	(159)
6.1.2 建立菜单	(160)
6.1.3 弹出式菜单	(162)
6.2 通用对话框	(164)
6.2.1 通用对话框控件	(165)
6.2.2 对话框应用实例	(166)
习题 6	(168)
第 7 章 文件	(169)
7.1 文件概述	(169)
7.1.1 文件结构	(169)
7.1.2 文件类型	(169)
7.2 文件的打开和关闭	(170)
7.3 相关函数	(171)
7.4 顺序文件	(172)
7.5 随机文件	(174)
7.6 文件系统控件	(178)
7.6.1 驱动器列表框	(179)
7.6.2 目录列表框	(179)
7.6.3 文件列表框	(179)

7.6.4 应用实例	(179)
习题 7	(181)
第 8 章 数据库的基本概念和 Access 数据库	(182)
8.1 数据库基础知识	(182)
8.1.1 数据库基本概念	(182)
8.1.2 关系数据库基本概念	(185)
8.1.3 SQL 基本命令	(187)
8.1.4 Access 系统简介	(190)
8.2 数据库和表的基本操作	(192)
8.2.1 创建数据库	(192)
8.2.2 表的建立	(197)
8.2.3 表间关系的建立和修改	(208)
8.2.4 表的维护	(210)
8.3 查询的基本操作	(215)
8.3.1 基本概念	(215)
8.3.2 创建选择查询	(219)
8.3.3 创建交叉表查询	(226)
8.3.4 创建参数查询	(230)
8.3.5 创建操作查询	(231)
8.3.6 创建 SQL 查询	(236)
8.3.7 操作已创建的查询	(239)
8.4 窗体的基本操作	(240)
8.4.1 认识窗体	(241)
8.4.2 创建窗体	(243)
8.4.3 创建其他窗体类型	(260)
8.4.4 使用窗体操作数据	(264)
8.5 报表的基本操作	(265)
8.5.1 报表概述	(265)
8.5.2 使用向导创建报表	(267)
8.5.3 使用报表设计器编辑报表	(272)
8.5.4 其他报表类型	(276)
8.6 页的基本操作	(280)
8.6.1 数据访问页的概念	(280)
8.6.2 创建数据访问页	(280)
8.7 宏	(285)
8.7.1 基本概念	(286)
8.7.2 基本操作	(286)
8.8 模块	(290)
8.8.1 基本概念	(290)

8.8.2 创建模块	(291)
8.8.3 宏和模块	(291)
8.8.4 VBA 程序设计基础	(292)
习题 8	(293)
第 9 章 程序设计实例	(295)
9.1 Access 数据库设计——学生学籍管理系统	(295)
9.1.1 系统分析与设计	(295)
9.1.2 数据库设计	(296)
9.1.3 创建数据表和关系	(299)
9.2 Visual Basic 界面设计——学生学籍管理系统	(303)
9.2.1 创建主界面	(304)
9.2.2 创建“录入学生基本信息”窗体	(308)
9.2.3 创建“学生基本信息查询”窗体	(311)
参考文献	(315)

第 1 章 程序设计概述

完成一项复杂的任务通常需要进行一系列具体工作。这些按照一定顺序安排的工作即操作序列，就称为程序。

程序设计就是设计、书写及检查程序的过程。要设计出一个好的程序，首先必须了解利用计算机解决实际问题的过程，其次应该掌握程序设计的基本技术，最后需要熟练掌握一种程序设计语言。本章将对程序设计的基本原理进行简要的介绍。

1.1 程序设计语言的发展

人们在使用计算机解决实际问题时，需要用某种特定的“语言”与计算机进行交流和沟通。计算机语言是人类与计算机交流信息的主要途径，它通过语法、语义、描述记号等表述各种运算和处理过程，能够被计算机所识别、理解、执行，最终完成某项任务。这些语言统称为计算机程序设计语言。

计算机语言大体上分为机器语言、汇编语言和高级语言。只有用机器语言编写的程序才能被计算机直接执行，而其他任何语言编写的程序则需要通过中间的编译过程。

1. 机器语言

机器语言是指由 0, 1 二进制代码组成的，能被计算机直接识别的机器指令的集合。机器语言能直接针对计算机的硬件结构描述各种算法，因此不需要翻译就能够被计算机直接执行，速度快。但是，用机器语言编程非常烦琐，程序的可读性极差，程序的修改、调试都极不方便。另外，机器语言是面向机器的，不同机器的指令系统也不同，不便于计算机的推广应用。

2. 汇编语言

汇编语言采用一定的助记符表示机器语言中的指令和数据，所以也称为符号语言。它用便于识别的符号，例如英语单词或其缩写作为助记符，来代替机器指令编写程序，然后由专门的转换程序将这些符号转换为机器语言指令代码。用汇编语言也能够针对计算机硬件编写程序，执行速度快。目前，针对一些实时性要求较高的实际问题，仍可以采用汇编语言编写程序。但是，与机器语言一样，它对机器的依赖性较强，且语言的通用性等问题还没有得到根本解决。

3. 高级语言

高级语言采用一组通用的英语单词、数学式和规定的符号，按照严格的语法规则和逻辑关系表述各种运算和处理过程。由于采用这种表达方式编写程序，接近于自然语言和数学语言，符合人们的习惯，为此称之为高级语言。高级语言具有较强的通用性，用高级语言编写

的程序能够在不同的计算机系统上运行。

高级语言的种类很多,适用于不同的应用领域。例如 Fortran, Basic, Cobol, Pascal, Ada, C 语言等。其中有些高级语言只适用于大、中型计算机。在 PC 不断发展、广泛应用的过程中,一些高级语言的编译程序才被移植到 PC 上。

最早的高级语言是 Fortran,是用于编制数值计算程序的主要语言,主要适用于科学计算方面。Pascal 语言是一种典型的结构化程序设计语言,具有丰富的数据类型,语句简明,程序结构严谨,控制语句简单灵活等特点,已经成为标准的计算机程序设计教学语言。C 语言具有数据结构丰富、表达力强、数据流控制灵活、通用性强和便于移植等特点,因此 C 语言既适合系统软件设计,也适合应用软件设计。C 语言是目前国际上公认的优秀程序设计语言。

高级语言又分为三类:面向过程的语言、面向问题的语言和面向对象的语言。

(1) 面向过程的语言

用计算机能够理解的逻辑来描述需要解决的问题和解决问题的具体方法、步骤。用这种语言编写程序时,程序不仅要说明做什么,而且还要告诉计算机如何做。如 Fortran, Basic, Pascal, C 语言等。

(2) 面向问题的语言

面向问题的语言又称非过程化的语言或第四代语言(4GL)。用该类语言解决问题时不用考虑计算机内部的逻辑问题,也不必关心求解问题的算法和求解过程,只需指出要计算机做什么,以及数据的输入和输出形式,就能得到结果。如 SQL 数据库查询语言。

(3) 面向对象的语言

面向对象的语言能够更直接地描述客观世界中存在的事物,以及它们之间的关系。它提供了简单的类机制及动态的接口模型,在对象中封装状态变量及相应的方法,实现了模块化和信息隐藏。并且通过继承机制,子类可以使用父类所提供的方法,实现了代码的复用。如 C++, Java, Visual Basic 语言等。

1.2 程序设计方法与风格

程序设计是一门技术,需要相应的理论、技术、方法和工具支持。就程序设计方法和技术的发展而言,主要经过了结构化程序设计和面向对象的程序设计阶段。

以前许多人认为程序只是给机器执行的,而不是供人阅读的,只要程序逻辑正确,能够被机器理解并执行就足够了,至于程序设计风格无关紧要。但是随着软件规模不断地增大,复杂性也在增加,人们逐渐认识到,在软件生存期中需要经常阅读程序。尤其是在软件测试阶段和维护阶段,编写程序者和测试、维护者都需要阅读程序。因此,除了好的程序设计方法和技术之外,程序设计风格也是很重要的。因为程序设计风格会影响软件的质量和可维护性,良好的程序设计风格可以使程序结构清晰合理,使程序代码便于维护。

程序设计风格是指编写程序时所表现出的特点、习惯和逻辑思路。程序设计的风格总体上应该强调简单和清晰,程序必须是可以理解的。当今主导的程序设计风格可以归结为“清晰第一,效率第二”。

在这一节里,将简要地介绍程序设计风格的四个方面,包括源程序的文档化、数据说明的方法、语句的结构、输入和输出方法中的一些问题。

1.2.1 源程序文档化

源程序文档化包括选择标识符的名字、程序注释，以及程序视觉组织等。

1. 符号名的命名

符号名即标识符，包括模块名、变量名、子程序名等。符号名的命名应具有实际含义，便于对程序功能理解。

2. 程序注释

程序中的注释是程序员与日后的程序读者之间沟通的重要手段。正确的注释能够帮助读者理解程序，为以后测试和维护提供明确的指导。

注释一般分为序言性注释和功能性注释两种。序言性注释通常位于每个程序的开始部分，给出程序的整体说明，主要内容可以包括：程序标题、程序功能说明、主要算法、接口说明、程序位置、开发简历、程序设计者、复审者、复审日期、修改日期等。功能性注释一般嵌在源程序之中，主要描述它后面的程序的功能。

3. 视觉组织

为了使程序的结构清晰，可以在程序中利用空格、空行、缩进等技巧，使程序具有层次感。恰当地利用空格，可以突出运算的优先性；程序段之间可以用空行隔开；缩进是指程序中的各行不必都在左侧对齐，因为这样会使程序缺乏层次关系。因此，对于选择语句和循环语句，把其中的程序段语句向右做阶梯式缩进，可以使程序的逻辑结构清晰，层次分明、一目了然。

1.2.2 数据说明

在编写程序时，需要注意数据说明的风格，使程序中的数据说明易于理解和维护。应注意以下方面。

首先是数据说明的顺序规范化。原则上，数据说明的顺序可以是任意的。但出于阅读、理解和维护的需要，最好使其规范化，使说明的先后次序固定。这样可使数据属性容易查找，也有利于测试、查错和维护。

其次是说明语句中变量安排有序化。当一个说明语句说明多个变量时，把变量按照字母顺序排序更好。

再次，使用注释说明复杂数据的结构，可以使今后的维护更加方便。

1.2.3 语句结构

编写出的程序应该简单易懂，语句的构造应该简单直接，不能单纯为了提高效率而使语句复杂化。编写程序时应注意以下方面。

① 在一行内只写一条语句，并采取适当的缩进格式，使程序的逻辑清晰、功能明确，提高程序的可读性。

② 编写程序应优先考虑清晰性，不应刻意追求技巧性，使程序难以理解。先要保证程序正确，然后才要求提高速度，除非对效率有特殊的要求。

- ③ 确保所有变量在使用前都进行初始化，避免使用临时变量而使程序的可读性下降。
- ④ 避免采用复杂的条件语句，尽量少使用“否定”条件的条件语句。
- ⑤ 数据结构要有利于简化程序；程序要模块化，模块功能要尽可能单一化；利用信息隐蔽，确保每一个模块的独立性。
- ⑥ 避免不必要的转移。
- ⑦ 尽可能使用库函数。
- ⑧ 不要修补写得不好的程序，应重新编写。

1.2.4 输入和输出

系统能否被用户接受，常常取决于输入/输出（I/O）的风格。无论是批处理的输入/输出方式，还是交互式输入/输出方式，都应以便于用户的使用为标准。因此，在设计和编程时应该考虑以下原则。

- ① 输入格式要简单，尽量简化用户的操作步骤。应该允许用户使用自由格式输入。
- ② 对所有输入数据都要检验数据的合法性，同时应允许默认值。
- ③ 输入一批数据时，要使用输入结束标志，不要让用户指定输入数据数目。
- ④ 在以交互式输入/输出方式进行输入时，应在屏幕上使用提示信息以提示用户输入合法数据。并且在数据输入过程中和输入结束时，在屏幕上给出状态信息。
- ⑤ 给所有输出加注释，并设计输出格式。

总之，要从实践中积累编制程序的经验，培养良好的程序设计风格，使编写的程序清晰易懂，易于测试和维护。

1.3 结构化程序设计

20 世纪 60 年代，许多大型软件的开发工作都遇到了严重的困难，软件开发的日程安排经常被拖延，花费大大超过预算，但最终产品并不可靠。此时，人们认识到软件开发比他们想象的复杂得多。此时的研究使结构化程序设计方法得到了发展。按照结构化规则编写程序，比用非结构化编程方法编写的程序结构更清楚，更容易测试和调试，也更易于修改。

1.3.1 结构化程序设计概述

Nicklaus Wirth 于 1971 年开发出了 Pascal 语言。它是以 17 世纪的数学家和哲学家 Blaise Pascal 命名的。设计 Pascal 语言的目的是在大学中教授结构化程序设计方法，但这种语言缺少许多特性，使其不能用于商业和工业的应用中，因此它在这些领域并没有被广泛接受。

Ada 编程语言是美国国防部（DOD）于 20 世纪 70~80 年代早期支持开发出来的。Ada 语言的一个重要功能是支持多任务，允许程序员指定许多并行发生的动作。

结构化程序设计是指仅使用三种基本控制结构实现程序的设计方法，它们是顺序结构、选择结构和循环结构。其基本设计原则是：模块化原则、自顶向下原则和逐步求精及限制使用 GOTO 语句原则。

结构化方法也称为功能分解法 (functional decomposition)。结构化程序设计方法的主要思想是对问题进行功能分解, 如果分解后得到的功能过大, 那么再对这些功能进行分解, 直到分解后的功能可以被方便地处理和理解为止。

1.3.2 结构化程序的基本结构

结构化程序设计方法是程序设计的先进方法和工具。采用结构化程序设计方法编写程序, 可以使程序结构良好, 且易读、易维护。

1. 程序流程图

程序流程图也称为程序框图。程序流程图的基本图符如图 1-1 所示。

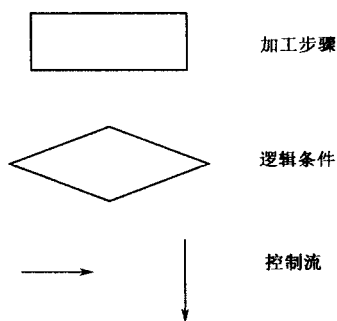


图 1-1 程序流程图的基本图符

2. 顺序结构

顺序结构就是按照程序语句行的自然顺序, 一条语句一条语句地执行程序。它是最基本、最常用的结构, 如图 1-2 所示。

3. 选择结构

选择结构又称为分支结构, 它包括简单选择结构和多分支选择结构, 这种结构可以根据设定的条件, 判断应该选择哪一条分支来执行相应的语句序列, 如图 1-3 所示。

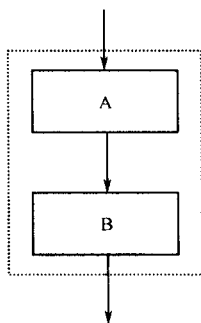


图 1-2 顺序结构

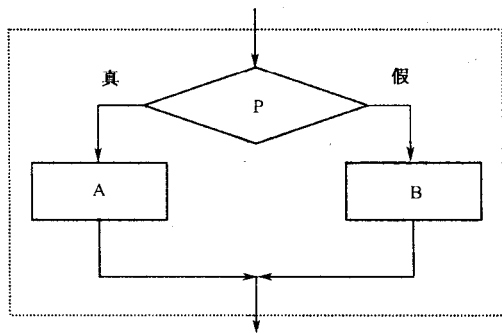


图 1-3 选择结构

4. 循环结构

循环结构又称为重复结构，它根据给定的条件，判断是否需要重复执行某一部分操作。循环结构有两类循环语句，包括当型循环结构和直到型循环结构，当型循环结构是先判断后执行循环体，如图 1-4 所示。直到型循环结构是先执行循环体后判断，如图 1-5 所示。

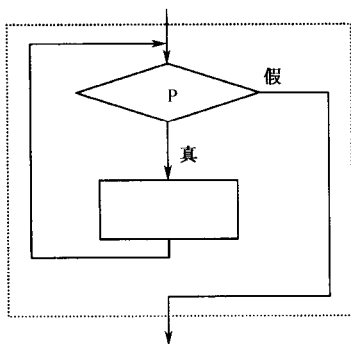


图 1-4 当型循环结构

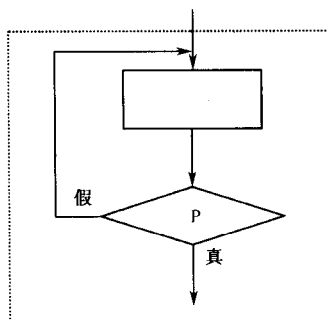


图 1-5 直到型循环结构

总而言之，遵循结构化程序的设计原则，按照结构化程序设计方法设计出的程序易于理解、使用和维护。程序员采用结构化程序设计，能够进行逐步求精，从而降低了程序的复杂性。由于结构化程序是由一些基本结构模块组成的，一些模块可以由机器自动生成，所以大大地减轻了程序员的工作量。从而提高了编程工作的效率，降低了软件开发成本。对于用户来说，结构化程序清晰易读、容易理解、便于使用和维护。

1.3.3 结构化程序设计的原则和方法

在了解和掌握了结构化程序设计原则、方法及结构化程序基本结构之后，在结构化程序设计的具体实施中，应注意以下原则。

- ① 使用程序设计语言中的顺序、选择、循环等控制结构表示程序的控制逻辑。
- ② 选用的控制结构只准许有一个入口和一个出口。
- ③ 程序语句组成容易识别的块，每一个块只允许有一个入口和一个出口。
- ④ 复杂结构应该用嵌套的基本控制结构进行组合嵌套来实现。
- ⑤ 严格控制 GOTO 语句的使用。

1.4 面向对象的程序设计

面向对象（Object-Oriented, OO）技术充分体现了分解、抽象、模块化、信息隐蔽等思想，可以有效地提高软件生产率、缩短软件开发时间及提高软件质量，是控制软件复杂性的有效途径。

与传统的结构化程序设计相比，面向对象的程序设计在描述和理解问题域时采用了不同的方法。其基本思想是，对问题域进行自然分割，以更接近于人类思维的方式建立问题域模