



教育部高等学校文科大学计算机课程教学指导分委员会立项教材  
普通高等教育计算机类专业“十三五”规划教材

# 计算机程序设计基础

(VB 版)

张增良 侯 申 编著



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



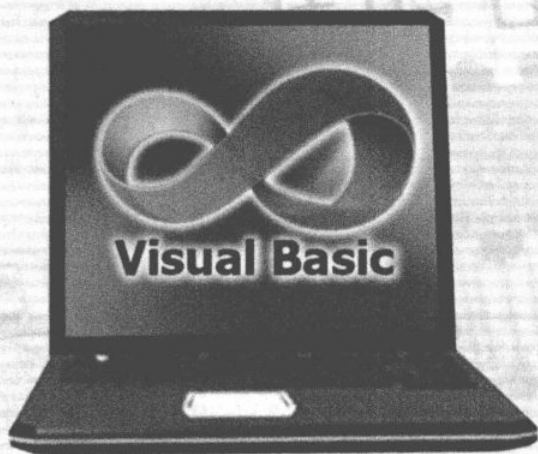
教育部高等学校文科大学计算机课程教学指导分委员会立项教材  
普通高等教育计算机类专业“十三五”规划教材

计算机专业·计算机类课程教材系列

# 计算机程序设计基础

(VB 版)

张增良 侯 申 编著



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

## 内容简介

本书是针对高等学校文科类专业本科学生的“计算机程序设计基础”课程而编写的通用教材。该教材注重理论与实践相结合,以 Visual Basic 6.0 为开发平台,系统讲授程序设计语言的数据、运算、语法、控制结构、过程与函数、文件等理论知识,并在介绍程序设计技术和典型算法的同时,通过大量有趣的实例详细讲解程序设计的方法和技巧。通过学习本教材,学生可牢固掌握程序设计相关知识,迅速有效地提高编程能力。

本书在内容的安排上力求循序渐进、由浅入深,在语言风格上力求简练、通俗、直观、生动,适合作为高等院校非计算机专业,特别是文科类专业计算机课程的教科书,也可作为工程技术人员和自学者的参考书。

---

### 图书在版编目(CIP)数据

计算机程序设计基础:VB版/张增良,侯申编著. —西安:西安交通大学出版社,2016.12  
教育部文科大学计算机教学改革项目立项教材·普通高等教育计算机类专业“十三五”规划教材  
ISBN 978-7-5605-9328-9

I. ①计… II. ①张… ②侯… III. ①程序设计-高等学校-教材 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2016)第 326037 号

---

书 名 计算机程序设计基础(VB版)  
编 著 张增良 侯 申  
责任编辑 贺峰涛 屈晓燕

---

出版发行 西安交通大学出版社  
(西安市兴庆南路10号 邮政编码710049)  
网 址 <http://www.xjtupress.com>  
电 话 (029)82668357 82667874(发行中心)  
(029)82668315(总编办)  
传 真 (029)82668280  
印 刷 虎彩印艺股份有限公司

---

开 本 787 mm×1 092 mm 1/16 印张 19.5 字数 470 千字  
版次印次 2017年9月第1版 2017年9月第1次印刷  
书 号 ISBN 978-7-5605-9328-9  
定 价 39.00 元

---

读者购书、书店添货或发现印装质量问题,请与本社发行中心联系、调换。  
订购热线:(029)82665248 (029)82665249  
投稿邮箱:eibooks@163.com

版权所有 侵权必究

# 前 言

本书是由教育部规划立项,针对高等学校文科类专业“计算机程序设计基础”课程而编写的教材。该教材以教育部《高等学校文科类专业大学计算机教学要求》为编写依据,并兼顾了《2016年全国计算机等级考试二级VB语言程序设计考试大纲》的要求,以“增强学生的信息技术素养,培养程序设计技能,提高实际应用能力”为教学的总体目标。

本书具有如下几个方面的特点:

(1)符合文科计算机教育需求。文科计算机教育从根本上说是隶属于宽泛意义上的“计算机科学与技术”专业名称下计算机应用型人才培养的一种计算机基础教育。它面向的是文科学生,既不同于计算机专业教学,也有别于理科(包括理工、农林、医药)专业教学,主要目的是培养学生的计算思维,掌握 Windows 应用程序的开发技能和技巧,能独立编写中小型应用程序,并能解决本专业的实际问题。

(2)内容充实,讲授生动。本书全面系统地介绍了程序设计的基本概念、数据与运算、程序基本控制结构、过程与函数、可视化程序设计、数据文件、程序访问数据库等内容;并以 Microsoft 公司的 Visual Basic 6.0 为开发平台,详细介绍了 Windows 应用程序的开发步骤及方法。本书针对文科学生的知识结构和思维习惯,在内容的呈现方式上,尽量做到图文并茂、形象直观,将难以理解的抽象内容转为容易理解的形象化内容,实现直观生动的讲授。

(3)理论结合实践。本书注意采用“任务驱动”和“案例式”教学方法,实现程序设计理论与实践的紧密结合。本书通过大量有趣的示例来介绍程序设计基本知识与基本方法,避免枯燥空洞的理论阐述,使学生在学懂理论知识的基础上,获得程序设计能力的提高,还可反向帮助理解某些较抽象的理论知识。本书所提供的程序代码都在计算机上调试通过。

(4)结构布局合理。全书内容经过精心安排和设计,结构清晰,逻辑性强,符合一般的教学规律和教育心理学的要求,围绕“程序设计”这个主题,通过大量实例深入浅出地介绍程序设计与算法的基本概念、程序设计语言、程序设计方法、程序开发步骤等程序设计相关知识。

(5)语言精炼、通俗易懂。本书采用精炼的语言和直观生动、循序渐进的讲授方法实施教学,使学生感受到学习程序设计并不是一件困难和乏味的事情。

本书选择 Visual Basic 6.0 作为语言和程序设计环境。之所以如此,是因为它具有易学、易用、程序开发周期短,并具有可视化设计界面等特点。对于初学者来说,采用 Visual Basic 6.0 可迅速进入角色并较容易地理解和掌握所学知识;对于很多程序员来说,Visual Basic 6.0 也是开发 Windows 应用程序的首选工具,能快速开发高质量的应用程序。

Visual Basic 6.0 自 1998 年发布以来,一直具有旺盛的生命力,可谓长盛不衰,有人开玩笑地说它就像 Windows 环境中“杀不死的小强”。不久前 Microsoft 公司将其 Windows 系统对 Visual Basic 6.0 程序的完全兼容支持延长到了 Windows 8 的整个产品周期。相比之下,.NET Framework 1.1 (2002 年发布)则仅仅被支持了 7 年,到 Windows 7(2009 年发布)时便被抛弃了。

本书是在作者长期从事教学和科研工作的基础上编写完成的。为方便教学,书中编排了大量例题和习题。本书可作为高等院校非计算机专业,特别是文科类专业计算机课程的教科书,也可作为工程技术人员和自学者的参考书。全书内容分为如下 13 章。

- 第 1 章 程序设计基础知识
- 第 2 章 Visual Basic 概述
- 第 3 章 数据类型与表达式
- 第 4 章 程序流程控制
- 第 5 章 数组
- 第 6 章 过程与函数
- 第 7 章 窗体及常用控件
- \* 第 8 章 绘图
- \* 第 9 章 ActiveX 控件
- \* 第 10 章 多窗体程序设计
- 第 11 章 文件操作
- \* 第 12 章 API 函数
- \* 第 13 章 数据库编程

其中,第 1 章由侯申编写,第 2 章~第 13 章由张增良编写。全书由张增良教授统稿、审稿和定稿。编写过程中,张绘宏教授、张志勇教授提出了建设性意见;顾宇涵、谢雨含、吕茜、赵喜强、李金超等老师给予了很大帮助,在此深表谢意。

教学实施建议:课时数应不少于 64 学时,理论课与实践课的课时比为 1:1,上文各章标题前标有“\*”号的章节内容可根据不同专业的需求灵活掌握。

由于作者水平有限,书中缺点和疏漏之处在所难免,望广大读者多提宝贵意见。联系方式:wyzzl@126.com。

编 者

2016 年 3 月

# 目 录

前言	
第 1 章 程序设计基础知识	(1)
1.1 程序与程序设计语言	(1)
1.1.1 程序设计的概念	(1)
1.1.2 程序设计语言	(1)
1.2 程序设计方法与过程	(4)
1.2.1 结构化程序设计	(4)
1.2.2 面向对象程序设计	(8)
1.2.3 程序设计的一般过程	(10)
1.3 算法	(11)
1.3.1 算法概述	(11)
1.3.2 算法的表示	(11)
1.3.3 常用算法及应用	(14)
习题 1	(18)
第 2 章 Visual Basic 概述	(21)
2.1 Visual Basic 的版本及特点	(21)
2.1.1 Visual Basic 的版本	(21)
2.1.2 Visual Basic 的特点	(21)
2.2 Visual Basic 6.0 开发环境	(22)
2.2.1 Visual Basic 6.0 的启动	(22)
2.2.2 Visual Basic 6.0 的界面布局	(23)
2.2.3 联机帮助	(26)
2.3 Visual Basic 程序的结构及开发步骤	(27)
2.3.1 一个简单而完整的 VB 程序	(27)
2.3.2 Visual Basic 程序的几个基本术语	(28)
2.3.3 Visual Basic 应用程序的结构	(30)
2.3.4 Visual Basic 应用程序的开发步骤	(31)
习题 2	(36)
第 3 章 数据类型与表达式	(38)
3.1 数据类型	(38)
3.1.1 基本数据类型	(38)
3.1.2 复杂数据类型	(41)
3.2 标识符与关键字	(43)
3.2.1 标识符	(43)
3.2.2 关键字	(43)
3.3 常量与变量	(44)
3.3.1 常量	(44)
3.3.2 变量	(46)
3.4 运算符与表达式	(49)
3.4.1 运算符	(49)
3.4.2 表达式	(52)
习题 3	(52)
第 4 章 程序流程控制	(55)
4.1 顺序结构	(55)
4.1.1 语句的书写规则	(55)
4.1.2 几个简单语句	(55)
4.2 分支结构	(59)
4.2.1 单行结构条件语句	(59)
4.2.2 块结构条件语句	(60)
4.2.3 If 函数	(62)
4.2.4 多分支选择结构语句	(62)
4.3 循环结构	(64)
4.3.1 For...Next 循环	(64)
4.3.2 While...Wend 循环	(68)
4.3.3 Do...Loop 循环	(70)
4.3.4 闲置循环	(72)
习题 4	(74)
第 5 章 数组	(78)
5.1 数组的声明	(78)
5.1.1 一维数组的声明	(78)
5.1.2 二维数组的声明	(79)
5.1.3 动态数组的声明	(79)

5.2 关于数组的操作 .....	(80)	7.2.2 命令按钮 .....	(153)
5.2.1 对数组的访问 .....	(80)	7.2.3 标签 .....	(153)
5.2.2 数组的刷新 .....	(81)	7.2.4 文本框 .....	(154)
5.2.3 有关数组的函数 .....	(81)	7.2.5 复选框 .....	(155)
5.3 控件数组 .....	(82)	7.2.6 单选按钮 .....	(155)
5.3.1 手工建立控件数组 .....	(82)	7.2.7 分组框 .....	(156)
5.3.2 通过程序建立控件数组 .....	(84)	7.2.8 图片框与图像框 .....	(157)
5.4 数组应用举例 .....	(84)	7.2.9 列表框与组合框 .....	(160)
习题 5 .....	(98)	7.2.10 滚动条 .....	(164)
		7.2.11 计时器 .....	(165)
<b>第 6 章 过程与函数</b> .....	(101)	7.3 菜单设计 .....	(170)
6.1 过程的概念 .....	(101)	7.3.1 下拉菜单 .....	(170)
6.1.1 什么是过程 .....	(101)	7.3.2 弹出式菜单 .....	(174)
6.1.2 过程的分类 .....	(103)	7.4 工具栏设计 .....	(176)
6.2 Sub 过程 .....	(104)	7.4.1 工具栏外观设计 .....	(176)
6.2.1 Sub 过程的声明 .....	(104)	7.4.2 编写工具栏事件过程代码 .....	(179)
6.2.2 Sub 过程的调用 .....	(105)	习题 7 .....	(180)
6.3 Function 函数 .....	(107)		
6.3.1 Function 函数的声明 .....	(107)	<b>第 8 章 绘图</b> .....	(183)
6.3.2 Function 函数的调用 .....	(108)	8.1 Visual Basic 中的坐标系统 .....	(183)
6.3.3 常用内部函数 .....	(111)	8.1.1 标准规格坐标系 .....	(183)
6.4 过程参数的传递 .....	(122)	8.1.2 自定义规格坐标系 .....	(184)
6.4.1 参数的传递方式 .....	(122)	8.2 绘图控件 .....	(185)
6.4.2 可选参数 .....	(128)	8.3 绘图方法 .....	(187)
6.4.3 数组参数 .....	(128)	8.3.1 画点和线 .....	(187)
6.4.4 不定量参数 .....	(131)	8.3.2 画矩形 .....	(188)
6.5 过程的作用域 .....	(132)	8.3.3 画圆、圆弧和椭圆 .....	(188)
6.6 递归调用 .....	(133)	8.4 图的调整和保存 .....	(189)
习题 6 .....	(135)	8.4.1 图的调整 .....	(189)
		8.4.2 图的保存 .....	(193)
<b>第 7 章 窗体及常用控件</b> .....	(138)	8.5 综合实例 .....	(194)
7.1 窗体 .....	(138)	习题 8 .....	(199)
7.1.1 窗体的创建 .....	(138)		
7.1.2 窗体的属性 .....	(138)	<b>第 9 章 ActiveX 控件</b> .....	(201)
7.1.3 窗体的方法 .....	(141)	9.1 CommonDialog 控件 .....	(201)
7.1.4 窗体的事件 .....	(142)	9.1.1 CommonDialog 控件概述 .....	(201)
7.1.5 窗体的生命周期 .....	(151)	9.1.2 CommonDialog 控件的属性和 .....	(202)
7.2 常用控件 .....	(152)	方法 .....	(202)
7.2.1 控件分类 .....	(152)		

9.2 TabStrip 控件 .....	(207)	习题 11 .....	(252)
9.2.1 TabStrip 简述 .....	(207)	<b>第 12 章 API 函数</b> .....	(253)
9.2.2 TabStrip 控件的属性 .....	(208)	12.1 API 函数的概念 .....	(253)
9.2.3 Tabs 集合的方法 .....	(209)	12.2 API 函数的声明 .....	(254)
9.2.4 TabStrip 控件应用实例 ..	(209)	12.2.1 API 的声明位置和格式	
9.3 多媒体控件 .....	(212)	.....	(254)
9.3.1 MMControl 控件 .....	(212)	12.2.2 API 浏览器 .....	(255)
9.3.2 Windows MediaPlayer 控件		12.3 API 函数的参数传递 .....	(257)
.....	(215)	12.3.1 句柄 .....	(257)
9.4 其他控件 .....	(217)	12.3.2 字符串参数 .....	(261)
9.4.1 StatusBar 控件 .....	(217)	习题 12 .....	(264)
9.4.2 ProgressBar 控件 .....	(219)	<b>第 13 章 数据库编程</b> .....	(265)
习题 9 .....	(221)	13.1 数据库概述 .....	(265)
<b>第 10 章 多窗体程序设计</b> .....	(222)	13.1.1 数据模型 .....	(265)
10.1 多重窗体 .....	(222)	13.1.2 关系型数据库的结构 ..	(265)
10.1.1 相关语句和方法 .....	(222)	13.1.3 SQL 简介 .....	(266)
10.1.2 多重窗体程序设计 .....	(223)	13.2 数据库访问技术 .....	(268)
10.2 多文档界面 .....	(227)	13.2.1 数据库应用程序的结构	
10.2.1 MDI 窗体的概念 .....	(227)	.....	(269)
10.2.2 有关 MDI 的属性、方法和事件		13.2.2 数据访问接口 .....	(269)
.....	(228)	13.3 使用 ADO 数据控件编程 .....	(270)
10.2.3 如何设计 MDI 应用程序		13.3.1 建立 Access 数据库 .....	(270)
.....	(229)	13.3.2 ADO 数据库控件 .....	(273)
10.3 窗体间的数据传递 .....	(231)	13.3.3 ADO 控件编程实例 .....	(278)
习题 10 .....	(233)	13.4 使用 ADO 对象编程 .....	(289)
<b>第 11 章 文件</b> .....	(235)	13.4.1 使用 ADO 对象编程的步骤	
11.1 文件的概念 .....	(235)	.....	(289)
11.2 文件的读写 .....	(236)	13.4.2 使用 ADO 对象编程实例	
11.2.1 文件读写相关函数 .....	(236)	.....	(292)
11.2.2 顺序文件 .....	(237)	习题 13 .....	(298)
11.2.3 随机文件 .....	(242)	<b>附录 标准 ASCII 码表与控制符含义表</b>	
11.2.4 二进制文件 .....	(245)	.....	(299)
11.3 文件操作命令与控件 .....	(246)	<b>参考文献</b> .....	(301)
11.3.1 文件操作命令 .....	(246)		
11.3.2 文件操作控件 .....	(248)		

# 第 1 章 程序设计基础知识

计算机系统由硬件系统和软件系统两大部分组成。硬件系统是计算机的躯体,为计算机的运算、存储和通信提供物质基础;而软件系统则是计算机的大脑,使计算机具有了智能。本章主要介绍软件的基本概念、程序设计语言、程序设计方法及常用算法等与程序设计相关的一些基本概念和基础知识。

## 1.1 程序与程序设计语言

本节介绍软件、程序和程序设计的概念,以及程序设计最重要的工具——程序设计语言。

### 1.1.1 程序设计的概念

计算机是依靠硬件和软件的配合进行工作的,硬件是计算机系统的基础,而软件则附着在硬件之上,指挥和控制硬件工作。

#### 1. 软件

软件是计算机程序、程序所用的数据以及相关文档资料的集合。软件的文档资料通常包括软件安装说明书、用户使用手册和其他相关技术资料、服务信息等。软件的核心是程序。

#### 2. 程序

所谓计算机程序,就是计算机为完成某一任务所必须执行的一系列指令的集合。一个程序应包括两方面的内容:一是对数据的描述,即数据结构;二是对操作步骤的描述,即算法。瑞士著名计算机科学家尼克劳斯·沃思(Niklaus Wirth)提出了一个公式:

$$\text{程序} = \text{数据结构} + \text{算法}$$

实际上,随着程序设计技术的发展,一个程序除了有数据结构和算法两个主要要素之外,还应涉及程序设计方法、计算机语言和运行环境等。因此,程序还可用下面的公式表示:

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计方法} + \text{语言工具和环境}$$

其中,算法是灵魂,数据结构是加工对象,语言是工具。同时,程序设计还需采用合适的方法。

#### 3. 程序设计

编排程序的过程即为程序设计。这个过程是将需要解决的问题转化成引导计算机运行的指令序列,并确保得到预期的结果。程序设计需借助于某个开发平台,利用某种程序设计语言,并采用合适的程序设计方法来完成。

### 1.1.2 程序设计语言

程序设计语言是程序员编写计算机程序不可缺少的工具。它主要经历了机器语言、汇编语言、高级语言和第四代语言(4GL)等几个发展阶段,图 1-1 直观表示了前 3 个阶段的程序设计语言。

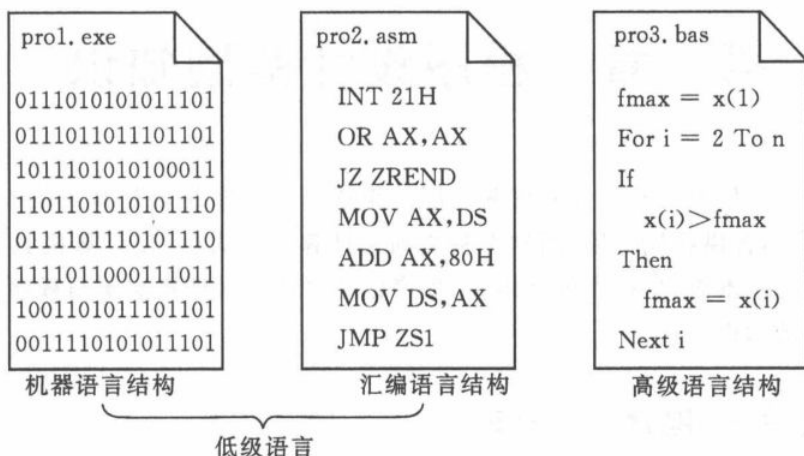


图 1-1 低级语言和高级语言

### 1. 机器语言

能被计算机直接识别和执行的二进制编码所表示的命令,称为机器指令;而机器语言直接由机器指令构成。机器指令是计算机的处理器可直接解读的数据,是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能,其具体的表现形式和功能与计算机系统的硬件结构相关。因此,不同的计算机具有不同的机器语言。例如,下面是用某种处理器的机器语言编写的程序,其功能是在屏幕上显示字符串“Hello”。

操作码	操作数	程序注释
11100000	01001000	/* 输出字符 H */
11100000	01100101	/* 输出字符 e */
11100000	01101100	/* 输出字符 l */
11100000	01101100	/* 输出字符 l */
11100000	01101111	/* 输出字符 o */
00000000		/* 停机 */

其中,每一行代表一条指令,每条指令由一个操作码和若干个操作数构成。操作码规定了指令的功能,操作数指明了被操作的对象。上述程序中的操作码 11100000 表示向屏幕输出字符,后面的操作数表示要输出字符的 ASCII 码;操作码 00000000 表示停止指令,没有操作数。

机器语言是最底层的程序设计语言,也是计算机能识别的唯一语言。由机器语言编写的程序不需转换就可直接被计算机系统识别并运行,执行速度快、效率高,但也存在着难记忆、难书写、编程困难、可读性和可移植性不好等缺点。使用机器语言适于编写底层软件,用它很难高效地编写出高质量的复杂程序。

### 2. 汇编语言

为了克服机器语言的缺点,人们采用助记符与符号地址来代替机器指令中的操作码与操作数。例如,用“add”表示加法操作,用“push”表示压栈操作,用“pop”表示弹栈操作等;操作数可用二进制、八进制、十进制或十六进制数表示,这种表示计算机指令的语言称为汇编语言。

下面是一段用汇编语言编写的显示“Hello”字符串的程序代码。

```
Start: jmp short Begin
Message DB ' Hello $ '
Begin:  mov     DX, OFFSET Message
        push   AX
        mov     AH, 09H
        int    21H
        pop    AX
        int    20H
```

上述程序用助记符代替了二进制的操作码,显然比机器语言程序的可读性和编程效率都有所提高,且保留了执行效率高的特点。

由于计算机只能识别和执行二进制形式的机器语言,因而用汇编语言编写的程序(源程序)必须经过汇编程序(assembly,即汇编器)“翻译”成机器语言程序(目标程序)后才能计算机上执行,其翻译过程称为“汇编”,如图 1-2 所示。

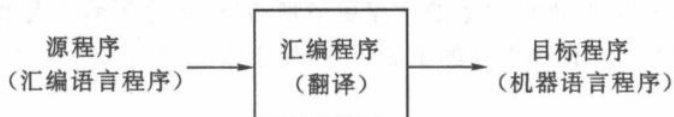


图 1-2 汇编语言源程序翻译为目标程序

汇编语言是 20 世纪 50 年代中期诞生的,也是一种面向机器的语言。由于它比机器语言的可读性好,又比其他语言执行效率高,所以现在许多系统软件的核心程序仍采用汇编语言编写。

尽管汇编语言较机器语言前进了一步,但差别仅仅体现在表示形式上,每条指令基本上与机器指令是一一对应的;虽然“助记符”的使用提高了程序的易读性,但编写的程序和机器语言程序一样依赖于具体的机器,不能方便地移植到另一种机器上;而程序员仍需要熟悉和记忆具体机器的硬件特征和相应的指令,且存储器也需人工分配。

总之,从机器语言到汇编语言并没有实质性的进步,它们共同被称为低级语言或面向机器的语言,属于第一代语言(1GL)。

### 3. 高级语言

面向机器的语言要求程序员必须对计算机硬件结构及工作原理十分熟悉,这对非计算机专业人士来说非常困难,为此,人们开发了高级语言,也称为算法语言。高级语言很接近人类的自然语言,通用易学,程序可读性好,可维护性强,可靠性高。例如,用 Visual Basic 高级语言编写显示“Hello”字符串的程序,只需 Print “Hello”一句即可。

1954 年,IBM 公司以约翰·巴科斯(John Backus)为主的研究小组开发出了第一种高级语言——FORTRAN。随后的 50 多年,共有几百种高级语言相继出现。根据程序设计方法的不同,目前主要有面向过程(Procedure-Oriented)和面向对象(Object-Oriented)这两类高级语言。

### 1) 面向过程的高级语言

面向过程语言与机器语言和汇编语言相比是个巨大的进步,它强调功能的抽象和程序的模块化,将问题的求解过程分解为对数据的一系列运算过程;程序员不必关心具体计算机的硬件特征,只需对解题的过程进行设计。软件从此摆脱了硬件的束缚,成为一个独立的产业。

采用面向过程的语言编写程序,关键是确定数据的表示形式及算法。20世纪70年代出现的高级语言(如PASCAL语言和C语言等)大多有此特征,通常被称为面向过程的语言,属于第二代语言(2GL)。

### 2) 面向对象的高级语言

面向对象程序设计解决问题的方法与面向过程程序设计完全不同,它把计算机求解问题的方案,设计为既相互独立又相互联系的若干对象间的相互协作,即每个程序的功能是通过对象的相互协作来完成的。基于这种方法的高级语言被称为面向对象的程序设计语言,如Visual Basic、Visual C++、C#和Java等。其中,Visual Basic具有开发界面友善、开发维护成本低、与Windows操作系统结合紧密的优点,很适于应用系统的开发,本书就是以Visual Basic 6.0作为开发环境来介绍程序设计的相关内容。

与面向过程的程序相比,面向对象的程序更清晰易懂,更适于编写大型复杂的程序。面向对象的高级语言已成为当今程序设计的主流语言,属于第三代语言(3GL)。

## 4. 第四代语言

第四代语言(4GL)出现于20世纪80年代,具有界面友好、简单易学、非过程化程度高、面向问题等特点。4GL是按计算机科学理论指导设计出来的结构化语言,如ADA、MODULA-2、SMALLTALK-80等。严格地讲,4GL并不只是语言,还是交互式程序设计环境。

4GL以数据库管理系统所提供的功能为核心,进一步构造了开发高层软件系统的良好用户环境,如报表生成、多窗口表格设计、菜单生成系统、图形图像处理系统及决策支持系统等。4GL提供了功能强大的非过程化问题定义手段,用户只需告知系统“做什么”,而不需说明“怎么做”,因此可大大提高软件生产效率,缩短软件开发周期。

4GL可分为查询语言与报表生成器、应用生成器、图形语言、可执行规格说明语言等4类。它们的代表性软件系统分别有:SQL、ADF;PowerBuilder、Oracle、Informix-4GL、Fox-Pro、NATURAL、FOCUS、RAMIS、MAPPER、UFO、NOMAD、SAS、MANTIS、IDEAL、LINC、FORMAL、SQL Windows、NPL、SPECINT等。

## 1.2 程序设计方法与过程

程序设计的实质是将人工求解问题的过程转换成计算机的算法语言源程序。为了设计具有可靠、易读、高效、可维护等特点的程序,需采用科学的程序设计方法。根据思维方式的不同,程序设计方法分为面向过程的结构化程序设计方法和面向对象的程序设计方法。

本节将结合实例,介绍结构化程序设计方法和面向对象程序设计方法,并给出程序设计的一般步骤。

### 1.2.1 结构化程序设计

结构化程序设计,是为解决最初程序结构比较随意、程序难读懂的问题而提出的一种程序

设计方法。它强调程序设计的风格和规范化,提倡清晰的程序结构。

结构化程序设计的概念最早由荷兰科学家艾兹格·迪科斯彻(E. W. Dijkstra)提出,1965年他在一次会议上指出:程序的质量与程序中包含的 GOTO 语句的数量成反比;如果程序中 GOTO 语句数量过多,则会严重破坏程序的可理解性和可维护性,人们难以读懂和修改这样的程序;应尽量少地使用 GOTO 语句,甚至可将它从高级语言中取消。

1966年,科拉多·伯姆(Corrado. Bohm)和朱塞佩·贾可皮尼(Giuseppe. Jacopini)证明了只用顺序、选择和循环3种基本控制结构就能实现任何单入口、单出口的程序,这为结构化程序设计方法的产生奠定了理论基础。

### 1. 结构化程序设计的原则

结构化程序设计方法引入了工程思想和结构化思想,使大型软件的开发和编程得到极大改善。其主要原则可概括为:自顶向下、逐步求精、模块化。

(1)自顶向下。设计程序时应先从最上层总目标开始,逐步使问题具体化。即先考虑总体,后考虑细节;先考虑全局目标,后考虑局部目标。

(2)逐步求精。对复杂的问题,应设计一些子目标作为过渡,逐步细化。

(3)模块化。把程序要解决的总目标分解为若干分目标,再进一步分解为具体的小目标,把每个小目标称为一个模块。

**【例 1.1】**输入一个班所有学生的“计算机基础”课程的成绩,求最高分、平均分和不及格生所占的比例并输出。

◆解题分析:按“自顶向下,逐步求精”的原则,首先须获得所有学生的成绩,然后进行成绩统计,最后输出统计结果。其中,成绩统计包括求最高分、平均分和不及格学生所占的比例。计算平均分可分解为计算总分和统计班级总人数两个子任务;计算不及格学生比例可分解为计算不及格学生数和统计班级总人数两个子任务。该问题自顶向下的功能分解结果,可用图 1-3 表示。

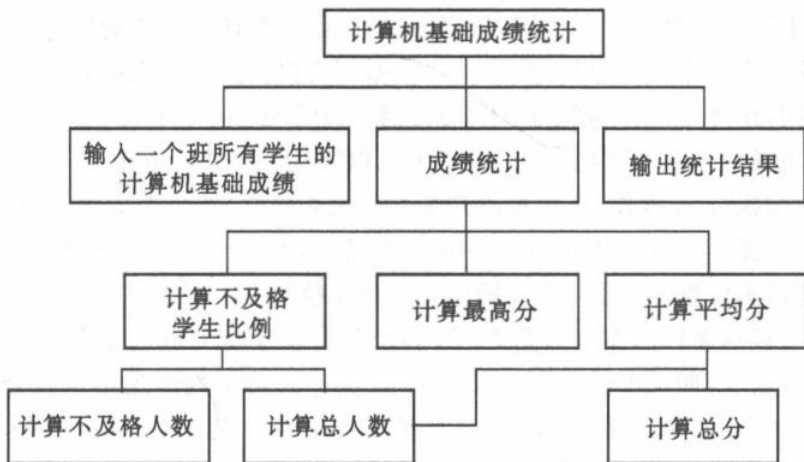


图 1-3 自顶向下功能分解图

### 2. 结构化程序设计的基本结构

采用结构化程序设计方法编写的程序,具有结构良好、易读、易理解、易维护等特点。结构

化程序设计仅仅使用顺序、选择和循环等三种基本控制结构。

1) 顺序结构

顺序结构是最简单的一种结构,计算机在执行顺序结构的程序时,按语句出现的先后顺序执行。在如图 1-4(a)所示的顺序结构中,计算机先执行 A 操作,再执行 B 操作。

2) 选择结构

当程序需要根据某种条件的成立与否有选择地执行一些操作时,应使用选择结构。这种结构包含一个判断框,根据给定的判定条件,从两个分支路径中选择执行其中的一个。从图 1-4(b)中可以看出,无论执行哪一个分支路径,程序都要通过汇合点 b。这个 b 点是选择结构的出口点。

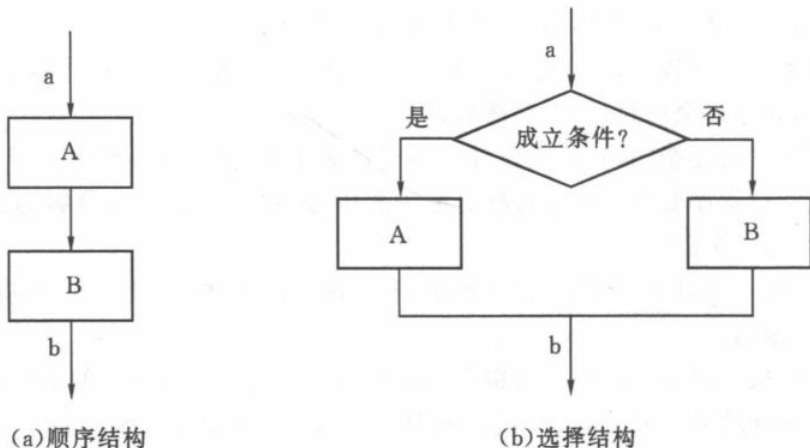


图 1-4 顺序结构与选择结构流程图

3) 循环结构

循环结构用于描述重复执行一些相同或相似的操作。要使计算机能够正确地完成循环操作,就必须使循环在执行有限次数后退出。因此,循环的执行要在一定的条件下进行。根据循环条件的位置不同,循环结构分为“当循环”和“直到循环”两种。

(1) 当循环:当循环的流程如图 1-5(a)所示。程序从 a 点进入循环,首先判断循环条件是否成立,如果成立则执行 A 操作;之后再次判断条件是否成立,若仍然成立,则再执行 A 操作,如此反复,直到某次循环条件不成立而不再执行 A 操作,而是从 b 点退出循环。显然,进入当循环后,如果一开始条件就不成立,则 A 操作一次都不执行。

(2) 直到循环:直到循环的流程如图 1-5(b)所示。程序从 a 点进入循环,执行 A 操作后判断退出条件是否成立,如果不成立则再次执行 A 操作并再次判断退出条件是否成立,若仍然不成立,则再次执行 A 操作,如此反复,直到某次退出条件成立不再执行 A 操作,而是从 b 点退出循环。显然,在进入直到循环后,A 操作至少执行一次。

顺序、选择和循环三种基本结构有一些共同的特点:只有一个入口;只有一个出口;基本结构中的每一部分都有机会被执行。也就是说,对每一个框来说,都应当有一条从入口到出口的路径通过它。**注意:**循环结构内不能存在“死循环”(即无终止的循环)。

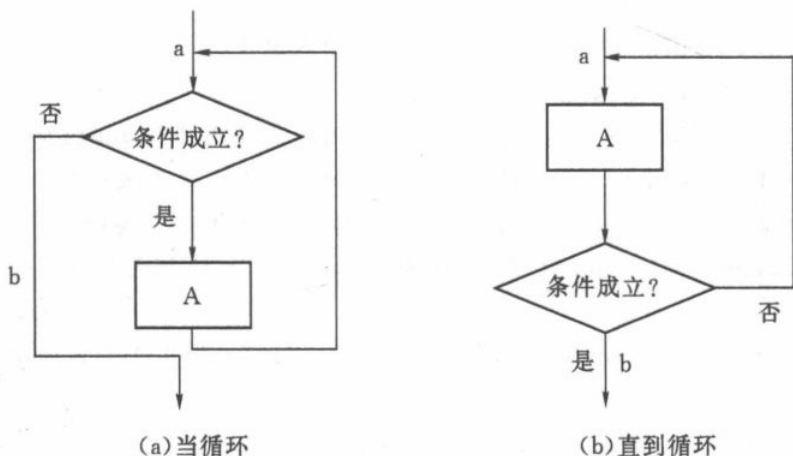


图 1-5 循环结构流程图

### 3. 程序设计的应用

初学者往往把程序设计理解为简单的编制一个程序。实际上,程序设计包括多方面的内容,而编制程序只是其中一个方面。下面结合例子,简要说明如何使用结构化程序设计方法进行程序设计。

**【例 1.2】**假如汽车 1 的极限行驶速度为 60 km/h,每百公里油耗为 5 L;汽车 2 的极限行驶速度为 80 km/h,每百公里油耗为 7 L。若 A、B 两地相距 240 km,两车都以极限速度行驶,求两车由 A 地行驶到 B 地分别需要的时间和耗油量。要求使用结构化程序设计方法。

#### ◆ 解题分析:

(1) 根据“自顶向下,逐步求精”的原则,将问题分解为如图 1-6 所示的 4 个过程。

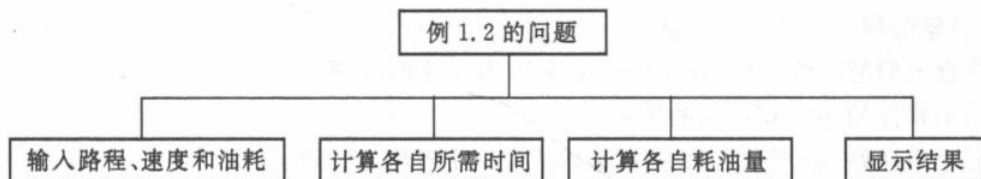


图 1-6 问题分解

第 1 个过程:输入路程和汽车 1、汽车 2 速度的值。

第 2 个过程:时间 1=路程/速度 1,时间 2=路程/速度 2。

第 3 个过程:耗油量 1=路程/100 \* 百公里油耗 1,耗油量 2=路程/100 \* 百公里油耗 2。

第 4 个过程:显示结果。

这个例子比较简单,经过第一轮分解,各过程的任务已很具体、明确,因此不必再分。对于复杂问题,可在第一轮分解的基础上继续分解,直到不需再分为止。

(2) 采用 3 种基本控制结构编写程序,用伪代码表示如下。

```
BEGIN                                     /* 程序开始 */
    Input Distance = 240                    /* 输入路程 Distance */
```

```

Input Speed1 = 60
Input Speed2 = 80          /* 输入速度 Speed */
Input FuelEco1 = 5
Input FuelEco2 = 7        /* 输入百公里油耗 FuelEco */
Time1 = Distance / Speed1 /* 计算汽车 1 所需要的时间 Time1 */
Time2 = Distance / Speed2 /* 计算汽车 2 所需要的时间 Time2 */
FuelNeed1 = Distance/100 * FuelEco1 /* 计算汽车 1 所需要的时间
                                     Time1 */
FuelNeed2 = Distance/100 * FuelEco2 /* 计算汽车 2 所需要的时间
                                     Time2 */
Output Time1,Time2,FuelEco1,FuelEco2 /* 显示结果 */
END                             /* 程序结束 */

```

### 1.2.2 面向对象程序设计

尽管结构化程序设计方法已得到广泛使用,但它是面向过程的,当程序较复杂时,容易出错且难以维护。目前,结构化程序设计方法已不能满足现代软件开发的要求,取而代之的是面向对象的程序设计(Object-Oriented Programming, OOP)。

#### 1. 面向对象程序设计概述

面向对象的程序设计方法是 20 世纪 80 年代初出现的。采用面向对象的方法解决问题,是将复杂系统抽象为一个个“对象”,以“对象”为思考问题的出发点,涉及哪个对象的功能,便由哪个对象自己去处理;不同对象之间通过消息或事件发生联系,各对象依据接收到的消息或事件进行工作。目前,这种“对象+消息”的面向对象的程序设计模式有取代“数据结构+算法”的面向过程的程序设计模式的趋向。

面向对象的程序设计有如下优势:

- (1)符合人们的思维习惯,便于分析复杂而多变化的问题。
- (2)易于软件的维护和功能的扩展。
- (3)可重用性好,能用继承的方式减少程序开发所花的时间。
- (4)与可视化技术相结合,改善了工作界面。

#### 2. 面向对象程序设计的基本概念

关于面向对象程序设计方法,人们对其概念有许多不同的看法和定义,但都涵盖了对象及其属性、方法,以及类和实例等几个基本要素。

##### 1) 对象

对象(Object)是面向对象程序设计方法中最基本的概念。对象是现实世界中可以独立存在、可以区分的实体,也可以是一些概念上的实体。例如,一本书、一个人、一所学校,甚至一个地球,这些都是对象。还有一些抽象事件,例如一次演出、一次球赛、一次借书等,也都可以看作是对象。这些实体和抽象事件通常都既有静态的属性,又有动态的行为(即“方法”)。对象是将描述这些实体属性的数据及施加其上的所有操作封装在一起所构成的统一体。

例如,一辆汽车是一个对象,其属性包括颜色、型号、载重量、极限速度、油耗等,其方法包括启动、刹车等;一个窗口也是一个对象,其属性包括大小、颜色、位置等,其方法包括打开、关

闭等。

## 2) 类和实例

类(Class)是具有共同属性和共同方法的对象的集合,是同类对象的抽象,它描述了属于该对象类型的所有对象的性质,而一个对象则是其对应类的一个实例(Instance)。

要注意的是,当使用“对象”这个术语时,既可以指一个具体的对象,也可以泛指一般的对象,但使用“实例”这个术语时,必然是指一个具体的对象。例如,Integer是一个整型类,它描述了所有整数的性质。因此,任何整数都是整数类的对象,而一个具体的整数123则是Integer类的一个实例。

类是关于对象性质的描述,它同对象一样,包括一组数据属性和作用于数据上的合法操作。例如,窗体中有3个圆,一个是半径为3 cm的红色圆,一个是半径为4 cm的绿色圆,一个是半径为1 cm的黄色圆,它们的位置、半径和颜色均不相同,可看成是3个不同的对象,但由于它们都有相同的属性(圆心坐标、半径、颜色)和操作(显示自己、放大或缩小半径、在屏幕上移动位置等),因此是同一类事物,可用Circle类来定义。

## 3. 面向对象程序设计的特点

面向对象程序设计具有封装、继承和多态性等特点。

### 1) 封装

封装(Encapsulation)是把对象的属性和操作结合成一个独立的系统单位,并尽可能隐藏对象的内部细节。封装是一种信息隐藏技术,目的是将对象的使用者和对象的设计者分开,用户只能看到对象封装界面上的信息,不必知道实现的细节。封装可降低开发过程的复杂性,提高效率和质量,也能保证程序中数据的完整性和安全性。

### 2) 继承

继承(Inheritance)是表示类之间相似性的机制,可以从一个类生成另一个类。子类(也称派生类)继承了父类和祖先类的数据和操作。例如,把“车”抽象为一个类,则“汽车”“摩托车”“自行车”都继承了“车”的性质,因而是“车”的子类。父类是所有子类的公共属性的集合,而子类则是父类的一种特殊化,可增加新的属性和操作。使用继承的主要优点是提高软件复用性、降低编码和维护的工作量。

### 3) 多态性

多态性(Polymorphism)是指同样的消息被不同的对象接受而产生完全不同的行为。例如,“启动”是“车”类都具有的操作,如“汽车”的“启动”是“发动机点火—启动引擎”,而“自行车”的“启动”是踩脚踏板。多态的特点可大大提高程序的抽象程度和简洁性,降低类和模块之间的耦合性,有利于程序的开发和维护。

## 4. 面向对象程序设计的应用

采用面向对象技术的开发过程一般分为面向对象分析(Object-Oriented Analyzing, OOA)、面向对象设计(Object-Oriented Designing, OOD)和面向对象编程(Object-Oriented Programming, OOP)三个阶段。下面结合实例,简要说明面向对象的程序设计方法。

**【例 1.3】**用面向对象程序设计方法编程实现例 1.2 的问题求解。

◆**解题分析:**根据面向对象程序设计的思想,汽车1和汽车2都是对象,具有多种属性和方法,同属于“汽车”这个类。本例用到“汽车”类的“极限速度”和“百公里油耗”这两个属性以