



新编 21 世纪高等院校计算机系列规划教材

Visual FoxPro 6.0

中文版

程序设计

北京希望电子出版社 总策划

董晓华 主 编

姜 雷 刘晓燕 王家伟 副主编

董晓华 陈浩杰 高 丽 编 著



新编 21 世纪高等院校计算机系列规划教材

Visual FoxPro 6.0

中文版

程序设计

北京希望电子出版社 总策划

董晓华 主 编

姜 雷 刘晓燕 王家伟 副主编

董晓华 陈浩杰 高 丽 编 著

 科学出版社
www.sciencep.com

内 容 简 介

本教材是多年从事语言教学的教师和科研人员根据基础教学的特点和应用型人才培养的基本要求,精心组织编写。

书中主要介绍了软件的基本概念和软件技术的发展过程;然后介绍了数据库的基本概念, Visual FoxPro 基础知识、基本操作, Visual FoxPro 程序设计和应用实例的编写方法;最后讲述了软件工程的基本概念和工程化观点,及软件生命周期、软件开发模型和软件过程等内容。

本书适合作为应用型人才培养的院校计算机及相关专业的教学用书,也可作为专业人士的开发参考书,还可作为全国计算机等级考试二级(Visual FoxPro)培训教材或自学参考书。

需要本书或技术支持的读者,请与北京清河6号信箱(邮编100085)发行部联系,电话:010-82702660 010-82702658 010-62978181 转103 或者238, 传真010-82702698, E-mail: tbd@bhp.com.cn。

图书在版编目(CIP)数据

Visual FoxPro 6.0 中文版程序设计/董晓华主编—北京:
科学出版社, 2006.1
(新编21世纪高等院校计算机系列规划教材)
ISBN 7-03-016289-7

I. V... II.董... III.关系数据库—数据库管理系统,
Visual FoxPro 6.0—程序设计 IV.TP311.138

中国版本图书馆CIP数据核字(2005)第110179号

责任编辑:李秉真 / 责任校对:孙红
责任印刷:双青 / 封面设计:梁运丽

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2006年1月第一版 开本:787×1092 1/16
2006年1月第一次印刷 印张:14 1/4
印数:1—3 000 字数:326 000

定价:22.00元

新编 21 世纪高等院校计算机系列规划教材编委会

主 任： 陈火旺 全国工科院校计算机专业教学指导委员会主任
中国工程院院士

副主任： 李国杰 中国计算机学会理事长
中科院计算技术研究所所长

杨芙清 中国计算机学会副理事长
中国科学院院士

沈复兴 全国高等师范学校计算机教育研究会副理事长
北京师范大学信息科学学院院长

何炎祥 武汉大学计算机学院院长

桂卫华 中南大学信息科学与工程学院院长

李仁发 湖南大学计算机与通信学院院长

陆卫民 中国科学出版集团北京希望电子出版社社长

委 员： （按姓氏笔画为序）

王江晴	王行恒	甘 玲	邓志华	孙中胜	刘晓燕	匡 松
任达森	李华贵	李超锋	李节阳	李新国	李龙澍	李建平
何婷婷	何登旭	张友生	张洪瀚	罗 琳	杨 波	杨宪泽
武兆辉	陈浩杰	陈 庄	郑明红	赵振华	洪汝渝	徐建军
徐 谡	唐光海	唐霁虹	唐 雁	高 丽	阎怀志	曹永存
覃 俊	董玉萍	董晓华	谢秉元	詹国华	戴上平	

秘 书： 徐建军

前 言

近年来,随着我国教育的不断深入,对计算机教材提出了新的要求。特点是重视高等教育中应用型人才的培养,为了编写适合应用型人才培养特点的计算机教材,经过精心组织和策划编写了本教材。

本教材由多年从事Visual FoxPro语言教学的教师和科研人员根据基础教学,以及最新的国家计算机等级考试二级Visual FoxPro考试大纲而组织编写的。结合初学者的特点,本书的内容首先介绍了软件的基本概念和发展,然后从介绍数据库的基本概念开始,深入浅出地介绍了Visual FoxPro的基础知识、基本操作,Visual FoxPro程序设计和应用实例的编写方法;最后还讲述了软件工程的基本概念和工程化观点,以及软件生命周期、软件开发模型和软件过程等内容。具体说来,本教材有以下特点:

(1) 内容编写思路新颖、注重适用。以实际问题引出概念,在例题中讲解语法及注意问题,便于初学者学习;同时强调实用性,在讲解语言中给出应用建议,使读者在掌握语法的同时明确它的实际用途。

(2) 注重程序设计能力的培养。在介绍Visual FoxPro基本知识的同时,结合实例着重介绍程序设计方法,使读者逐步建立起程序结构的概念,掌握程序设计的一般思路和方法,培养学生独立解决问题的能力。

(3) 以适用于初学者为目地进行编排。知识难度控制在初学者能接受的范围内,对于哪些内容可以了解、哪些必须掌握、哪些是较深入地应用等都给出了明确的说明。语法介绍简明扼要、条理清楚、例题丰富。程序例题尽量简单易学,以适合初学者。本书所有程序都运行通过。

本书由重庆大学的董晓华博士主编和完成全书的统稿工作,由姜雷、辽宁警官高等专科学校的刘晓燕老师及王家伟老师担任副主编。参加本书编写的还有:何清太、熊心志、林勇、罗琳,以及山东日照曲阜师范大学的陈浩杰老师和华中师范大学的高丽老师,在此一并表示感谢。

本书在编写过程中,参考了大量的国内外文献资料,在此,谨向文献资料的作者表示感谢。

本书配有便于教学用的电子教案,并配有四套模拟试题以及习题参考答案。由于篇幅有限,书中没有列出这些内容,以上所有内容可到<http://www.b-xr.com>网站下载。

由于作者水平有限,缺点和错误在所难免,欢迎读者和专家批评指正。

编 者

目 录

第1章 概论.....	1
1.1 计算机系统.....	1
1.1.1 计算机系统的发展.....	1
1.1.2 软件与软件的分类型.....	2
1.2 计算机语言和语言处理程序.....	3
1.2.1 计算机语言概述.....	3
1.2.2 计算机语言处理程序概述.....	4
1.3 计算机软件技术概述.....	7
1.3.1 程序设计.....	7
1.3.2 数据结构.....	8
1.3.3 操作系统和计算机网络.....	9
1.3.4 数据库管理系统.....	15
1.3.5 软件工程.....	22
1.3.6 面向对象方法学.....	23
1.4 软件开发环境.....	24
1.4.1 软件开发方法.....	24
1.4.2 软件开发模型.....	24
1.4.3 软件开发环境.....	24
1.5 小结.....	25
1.6 习题.....	26
第2章 数据库系统及 Visual FoxPro 6.0 简介.....	27
2.1 数据库与数据库管理系统简介.....	28
2.1.1 数据.....	28
2.1.2 数据库.....	29
2.1.3 数据库管理系统.....	29
2.1.4 数据库系统.....	30
2.2 关系数据库系统的性质和特点.....	30
2.2.1 数据模型.....	30
2.2.2 关系模型的有关术语.....	32
2.2.3 关系模型的性质.....	34
2.2.4 关系操作.....	34
2.3 中文版 Visual FoxPro 6.0 简介.....	35
2.3.1 Visual FoxPro 6.0 中文版的特点.....	35
2.3.2 系统运行的软硬件环境.....	36
2.3.3 Visual FoxPro 6.0 中文版的安装.....	36

2.3.4 Visual FoxPro 6.0 系统的启动与退出.....	37
2.4 小结.....	37
2.5 习题.....	38
第3章 Visual FoxPro 6.0 中文版的工作环境.....	40
3.1 用户界面介绍.....	40
3.1.1 菜单栏.....	40
3.1.2 工具栏.....	41
3.1.3 命令窗口.....	41
3.1.4 结果显示区.....	41
3.1.5 状态栏.....	41
3.1.6 标题栏.....	41
3.2 操作 Visual FoxPro 6.0.....	41
3.2.1 Visual FoxPro 6.0 的操作方式.....	41
3.2.2 Visual FoxPro 6.0 的工具.....	42
3.2.3 文件扩展名及文件类型.....	44
3.3 配置 Visual FoxPro 6.0 中文版.....	44
3.4 项目管理器的使用.....	48
3.4.1 什么是项目管理器.....	48
3.4.2 创建新的项目文件.....	48
3.4.3 项目管理器中包含的内容.....	50
3.4.4 项目管理器的各种操作.....	51
3.5 小结.....	52
3.6 习题.....	52
第4章 程序设计基础知识.....	54
4.1 程序设计的方法.....	54
4.1.1 程序设计的优点.....	55
4.1.2 命令窗口的使用方法.....	55
4.1.3 命令格式与书写规则.....	56
4.1.4 程序的操作过程.....	56
4.2 程序设计的基础知识.....	59
4.2.1 数据类型.....	59
4.2.2 运算符和表达式.....	61
4.2.3 函数.....	68
4.3 结构化程序设计.....	73

4.3.1 顺序结构程序.....	75	6.2.2 控制字段中重复值的输入.....	119
4.3.2 选择结构程序.....	76	6.2.3 创建多个索引.....	119
4.3.3 循环程序结构.....	80	6.2.4 控制访问记录的顺序.....	120
4.3.4 多重循环程序结构.....	84	6.3 重建活动索引文件.....	121
4.4 用户自定义函数和过程.....	85	6.4 通过改变记录的物理顺序进行排序.....	121
4.4.1 子程序及调用.....	86	6.5 小结.....	122
4.4.2 过程及过程调用.....	87	6.6 习题.....	123
4.4.3 用户自定义函数.....	89	第7章 数据库的操作	124
4.5 小结.....	90	7.1 数据库的设计.....	124
4.6 习题.....	90	7.1.1 数据库的设计过程.....	124
第5章 数据表的使用	94	7.1.2 确定建立数据库的目的.....	125
5.1 有关表的基础知识.....	94	7.1.3 确定需要建立的表.....	125
5.1.1 自由表和数据库表.....	94	7.1.4 确定表中所需要的字段.....	125
5.1.2 表间关系的种类.....	94	7.2 创建数据库.....	126
5.2 创建表.....	95	7.2.1 创建和打开数据库.....	126
5.2.1 定义表的结构.....	95	7.2.2 数据库设计器.....	127
5.2.2 打开或关闭表文件.....	96	7.2.3 在数据库中添加表.....	127
5.2.3 建立数据库表.....	97	7.2.4 从数据库中删除表.....	128
5.2.4 设置字段属性.....	98	7.2.5 关闭数据库.....	128
5.2.5 设置索引.....	99	7.2.6 更新表和数据库的链接.....	128
5.3 修改表的结构.....	100	7.3 表的字段级的属性.....	128
5.3.1 在表中增加字段.....	101	7.3.1 重命名字段.....	128
5.3.2 从表中删除字段.....	101	7.3.2 给字段添加注释.....	129
5.3.3 修改表中现有字段.....	101	7.3.3 给字段添加标题.....	129
5.4 编辑数据表.....	101	7.3.4 建立输入掩码.....	129
5.4.1 显示表中的记录.....	101	7.3.5 设置字段默认值.....	129
5.4.2 向表中追加或删除记录.....	103	7.3.6 使用字段级的有效性检验.....	129
5.4.3 直接修改表中的记录.....	106	7.4 表在数据库中的属性.....	129
5.4.4 批量修改表中的记录.....	108	7.4.1 设定长表名.....	130
5.4.5 表中记录的定位.....	108	7.4.2 使用记录级有效性检验.....	130
5.4.6 表的复制操作.....	111	7.4.3 使用触发器.....	130
5.4.7 数据的统计.....	112	7.4.4 选择工作区.....	131
5.5 小结.....	112	7.4.5 表之间的关联性.....	131
5.6 习题.....	112	7.5 小结.....	131
第6章 表的索引	116	7.6 习题.....	132
6.1 索引及索引文件的种类.....	116	第8章 查询和视图	133
6.1.1 什么是索引.....	116	8.1 查询.....	133
6.1.2 索引的种类.....	116	8.1.1 查询的含义及查询设计的步骤.....	133
6.2 创建结构复合索引.....	117	8.1.2 选择查询字段.....	134
6.2.1 创建索引.....	118	8.1.3 设置排序记录.....	135

8.1.4 建立多表联接查询.....	136	10.2.3 设置表单的行为.....	173
8.1.5 设定查询条件.....	137	10.3 使用表单.....	174
8.1.6 建立分组查询.....	138	10.3.1 表单的基本操作.....	174
8.1.7 运行查询.....	139	10.3.2 在运行时刻设置属性.....	175
8.1.8 使用 SELECT 命令实现查询.....	139	10.3.3 表单间的调用.....	175
8.2 视图的应用.....	140	10.4 表单中的控件.....	178
8.2.1 视图的特征.....	140	10.4.1 表单中控件简介.....	178
8.2.2 创建视图.....	140	10.4.2 与控件相关的操作.....	179
8.2.3 使用视图更新数据.....	141	10.5 向表单中添加控件.....	179
8.3 小结.....	142	10.5.1 使用微调控件.....	180
8.4 习题.....	143	10.5.2 使用命令按钮或命令按钮组 控件.....	180
第9章 面向对象程序设计.....	145	10.5.3 使用标签控件显示信息.....	182
9.1 面向对象的基本知识.....	145	10.5.4 使用计时器控件.....	182
9.1.1 对象、属性、方法和事件.....	145	10.5.5 使用图像控件在表单上显示 图像.....	183
9.1.2 类、子类和类的特征.....	146	10.5.6 使用表格控件显示多行数据.....	184
9.1.3 类的层次关系和类的类型.....	149	10.5.7 页框控件.....	185
9.1.4 处理对象.....	150	10.6 小结.....	186
9.2 事件驱动程序设计技术.....	152	10.7 习题.....	186
9.2.1 事件驱动程序设计.....	152	第11章 设计菜单与工具栏.....	188
9.2.2 Visual FoxPro 6.0 中常用的事件.....	153	11.1 创建菜单系统.....	188
9.2.3 容器层次和类层次.....	154	11.1.1 规划菜单系统.....	188
9.2.4 为事件编写代码.....	154	11.1.2 菜单设计器.....	189
9.3 用类设计器创建类.....	155	11.1.3 创建菜单系统.....	191
9.3.1 决定要创建的类的类型.....	155	11.1.4 创建快捷菜单.....	194
9.3.2 创建新的类.....	156	11.1.5 为菜单系统指定任务.....	195
9.4 类的基本操作.....	159	11.1.6 测试与调试菜单系统.....	195
9.4.1 修改用户定义类和删除类库中 的类.....	159	11.1.7 定制菜单系统.....	196
9.4.2 使用自定义的类.....	160	11.2 创建自定义工具栏.....	196
9.4.3 删除用户注册的类.....	162	11.3 小结.....	198
9.5 小结.....	162	11.4 习题.....	198
9.6 习题.....	163	第12章 设计报表.....	200
第10章 屏幕界面的设计.....	164	12.1 设计报表.....	200
10.1 表单对象.....	164	12.1.1 报表的有关概念.....	200
10.1.1 表单设计器.....	164	12.1.2 报表布局.....	201
10.1.2 数据环境.....	165	12.2 创建报表布局.....	202
10.1.3 对象的属性.....	167	12.2.1 用报表向导来创建报表格式 文件.....	202
10.2 创建表单.....	169	12.2.2 创建快速报表.....	204
10.2.1 创建表单的方法.....	169		
10.2.2 使用表单集.....	172		

12.3 修改报表.....	206	13.1.1 软件工程的产生.....	218
12.3.1 设置报表数据源.....	206	13.1.2 软件工程概念及目标.....	219
12.3.2 为报表增加新的控件.....	207	13.1.3 软件工程研究的内容.....	219
12.3.3 数据报表的分组.....	211	13.2 软件开发模型.....	225
12.4 打印和预览报表.....	212	13.2.1 软件生命周期.....	225
12.4.1 报表预览.....	213	13.2.2 软件开发模型.....	227
12.4.2 打印报表.....	213	13.3 软件工程过程.....	231
12.5 设计标签.....	214	13.3.1 基本过程.....	231
12.5.1 使用“标签向导”创建标签.....	214	13.3.2 支持过程.....	232
12.5.2 使用“标签设计器”创建标签.....	216	13.3.3 组织过程.....	234
12.6 小结.....	216	13.3.4 ISO 9000-3.....	236
12.7 习题.....	217	13.4 小结.....	237
第 13 章 软件工程基础.....	218	13.5 习题.....	237
13.1 软件工程概述.....	218	附录 习题解答.....	238

第1章 概 论

计算机系统的组成包括硬件和软件两部分。硬件是组成计算机的物理设备，软件是控制和管理计算机以及方便用户使用计算机的程序和数据，软件包括了系统软件和应用软件。本章首先介绍计算机系统的发展和计算机软件、计算机语言和语言处理程序，然后介绍计算机软件技术的基本理论和软件开发环境。

本章重点：

- 计算机系统的组成
- 计算机程序设计语言与软件的基本概念
- 数据库的基本概念
- 软件开发环境的基本概念

1.1 计算机系统

由计算机的硬件和软件组成统一的整体称为计算机系统。计算机系统的硬件是构成计算机系统的全部硬设备，如中央处理机（CPU）、内存储器（RAM、ROM）、外存储器（软盘、硬盘和磁带等）以及各种输入设备（键盘、鼠标、显示器和打印机）等。为了充分发挥计算机硬件的使用效率还需要配备计算机的软件，它是构成计算机系统的全部软件设备，如系统软件和应用软件。

1.1.1 计算机系统的发展

计算机系统从 1950 年到现在约有近 50 年的历史，经历了四个时期的发展过程。图 1-1 描述了以计算机为基础的计算机系统的发展过程。

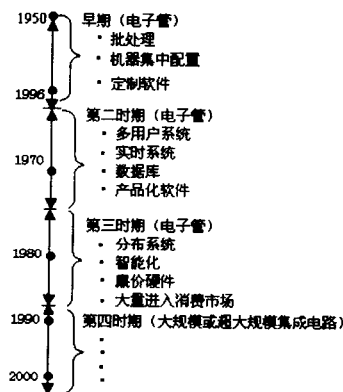


图 1-1 计算机系统的发展过程

1. 早期

在计算机系统发展的早期,硬件经历了不断地变化,而软件只是被多数人作为一种事后工作来看待。计算机程序设计几乎没有什么系统的方法可以遵循,软件的开发进度缓慢,或成本一增再增,并软件仅是为每一种用途分别设计,通用性很差,没有形成软件产品。在此期间,多数系统采用批处理工作方式。

2. 第二个时期

在这个时期是以产品化软件的使用和“软件车间”的出现为特点,随着以计算机为基础的系统日益增多,计算机软件库开始膨胀,软件维护的开销开始令人恐慌,尤其是许多软件带有个人色彩,使得它们实际上不可能维护,“软件危机”开始了。

3. 第三个时期

从 70 年代初期开始,一直延续到 80 年代初期是计算机系统发展的第三个时期,在这个时期分布式系统(多个计算机、各机器并行执行和相互通信)极大地增加了计算机为基础的系统的复杂性。由于微处理机和有关部件的功能越来越强且价格越来越低,因此,在计算机应用领域中,具有“嵌入智能”的产品取代了较大的计算机。硬件的迅速发展已经超过所提供的支持软件的能力,“软件危机”日益严重,为了维护软件,消耗掉了数据处理预算的 50% 以上,而软件的开发又跟不上新系统对软件需求的步伐。为了对付不断增长的软件危机,软件工程才得到了认真对待。

4. 第四个时期

具有更大主存的 16 位和 32 位微处理机,为以计算机为基础的系统开辟了全新的应用领域。

1.1.2 软件与软件的分类

软件是所有程序、数据和文档的总称,计算机软件通常可以分为系统软件和应用软件。

1. 系统软件

系统软件是由计算机系统的设计者和生产者提供的程序和使用及维护手册的总称,这些程序的作用在于扩充计算机功能,控制计算机高效率运行,为用户提供更多地方便。系统软件主要包括:操作系统,程序设计的各种语言的处理程序(汇编器、解释器和编译器),数据库管理系统,计算机网络软件,诊断和故障处理程序,编辑程序和调试程序等。由于配制了各种系统软件,从而大大地改善了用户使用计算机环境。

2. 应用软件

应用软件则是用户利用计算机系统提供的功能为解决特定问题而编制的程序及其使用说明书和维护手册的总称,这类程序种类繁多,不同的应用领域、不同的部门需要编制不同的应用程序。为了减少重复劳动,提高程序的重用率,应用软件也在逐步商品化,形成各种软件包。例如,各种科学计算机软件包、事务管理软件包、辅助教学软件包等,用户可根据自身的需要选购,十分方便。

有时,系统软件和应用软件不能截然分开,如各种标准库,既可以看成是应用软件,也可以看成是计算机厂家提供的系统软件。对于一个使用计算机的工程技术人员来说,熟

悉各种系统软件的目的是为了更好地发挥计算机的功能，更有效地从更高水平上开发应用软件，编制应用程序。

1.2 计算机语言和语言处理程序

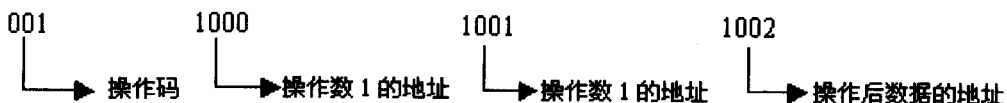
1.2.1 计算机语言概述

从1946年以来的半个世纪中，计算机硬件经历了四个时期，如电子管、晶体管、集成电路和大规模集成电路到超大规模集成电路的换代。现代计算机正向着巨型、微型、并行、分布、网络化和智能化几个方面发展，其相应的软件逐步丰富和完善。随着计算机硬件的发展，计算机语言经历了机器语言、汇编语言、高级程序设计语言和面向程序设计语言等过程。

1. 机器语言

在计算机出现的初期，人们直接使用机器语言编写程序，这种程序叫机器语言程序。它要求程序设计人员相当熟悉计算机的所有细节，例如计算机的指令系统、存储容量、寄存器的类型与个数等。这样做工作量大，非常容易出错、不易修改，并每台计算机的指令各不相同，其程序依赖于特定的机器类型，局限性很大，移植性更差。

所谓机器语言就是指机器的指令系统，它包含了每台计算机能够执行的基本操作，这些操作是由指令形式（一个字节或多个字节的二进制代码形式）来书写。它是裸机惟一直接能识别和执行的指令（或语言）。例如：



表示将地址1000，1001中的内容相加的和存放在1002地址之中。

2. 汇编语言

为了摆脱机器语言编程的困难，使计算机成为广大工程技术人员都能使用的工具，同时也为了减轻程序设计人员编程和调试机器语言程序的繁重的劳动，提高程序设计的效率，出现了以符号指令来替代机器指令的编程办法。用符号语言编写的程序称为符号程序。它是使用以英文名称的缩写的指令助记符，例如取数用ID，加法用ADD等，它比单纯用“0”和“1”的代码串容易记忆。在此基础上进一步扩充就成为汇编语言，用汇编语言编制程序比用机器指令代码方便得多，不仅易于检查和修改错误，而且指令、原始数据和结果的存放单元可由机器根据定位为指令自动分配。但是符号指令是不能被计算机直接识别和执行，必须将它翻译成机器语言后才能执行，这个翻译工作是由汇编程序来完成的。

程序设计人员在使用汇编语言编程时，仍必须熟悉机器的硬件结构，因此程序设计仍然很繁琐和低效。但是正因为依赖于硬件，熟练的程序员可结合硬件的特点设计出高质量的程序，所以直至今日，汇编语言仍起着重要的作用，常用于对时间和空间效率要求较高的计算机系统核心程序和实时控程序等。

3. 高级语言

高级语言（面向过程的程序设计语言）是由表达各种意义的“词”和“数学公式”，按照一定语法规则组成的面向过程的语言。用高级语言进行程序设计比较接近人的习惯，编出的程序与具体的机器指令无关，可以独立于机器，通用性较强如同汇编语言一样。用高级语言编写的源程序机器是不能直接识别和执行，必须翻译成机器指令后才能在计算机上执行。

目前，世界上有许多多程序设计语言，较流行的有好几十种，例如用于科学与工程计算的 FORTRAN 语言、会话式的 BASIC 语言、教学和系统设计的 PASCAL 语言、数据处理的 COBOL 语言、人工智能中使用的表处理 LISP 语言、逻辑程序设计的 PROLOG 语言、大型通用的 PL / I 语言，一种结构化、模块化、可编译的通用 C 程序设计语言及支持数据抽象类型概念的 Ada 语言等。

4. 面向对象的程序设计语言

与过去面向过程程序设计比较，面向对象程序设计语言的最大特点在于面向的是对象而不是面向过程。所谓对象是现实世界中的实体，例如桌子、电视机、硬件和规律等。把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一个类的对象，例如人是一个类，而每一个具体的人就是人类中的一个对象。面向对象的程序设计是程序设计中的一种新思想，该思想认为程序是相互联系的离散对象的集合，面向对象程序设计语言是支持这种新思想的程序设计语言，它主要具有封装性、继承性和多态性等特点。

用面向对象的程序设计语言编的程序也不能直接为机器所识别和执行，必须翻译成机器指令后才能在机器上运行。面向对象程序设计语言分为两类，一类是在面向过程程序设计语言的基础上增加了对象的成分构成混合性面向对象的程序设计语言，例如 C++、VC++、Borland C++、Delphi 等；另一类是按照面向对象设计思想而构成的面向对象程序设计语言称为纯的面向对象设计语言，如 Smalltalk、Eiffel 等。

1.2.2 计算机语言处理程序概述

用计算机语言编写的程序除机器语言程序外，全部程序都不能为机器直接识别和执行，必须经过各种语言的处理程序（或翻译器）处理转换成机器语言程序才能为机器所识别和执行。由翻译所得的结果称为目标程序，目标程序可以用机器语言表示也可以用汇编语言或其他中间语言表示，语言处理程序提供这种翻译功能。

1. 汇编程序

汇编程序的主要功能是把汇编语言源程序翻译成机器能识别的目标程序，其汇编过程如图 1-2 所示。

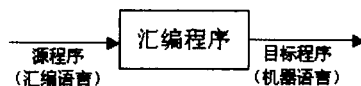


图 1-2 汇编过程

源程序通常是由 ASCII 码表示的符号化的指令串组成。当汇编程序加工源程序时，总是从头到尾一个符号接着一个符号地阅读，称为扫描源程序。从头到尾扫描

一次源程序为扫描一遍。具体地说，汇编程序必须完成几个方面任务：

- 处理语句中的操作符，生成与相应的机器码，产生机器指令。

- 处理语句中的标号和名字，并代之以具体的单元地址。
- 处理为指令。

为了完成汇编程序的翻译工作，需要以下表格：

- 机器码操作表（MOT）：用以确定指令的长度和助记符转换成机器码；
- 伪指令操作表（POT）：用以查找伪指令对应的操作；
- 地址计数器（比）：用以追踪和确定指令地址；
- 符号表（SYNT）：用以记录各标号、名字。

2. 编译程序

用高级语言编写的源程序可以通过编译的途径将其翻译成语义上等价的目标程序。目标程序可以用机器语言、汇编语言或某种中间语言表示，如果目标程序是用汇编语言表示的，则它必须再经汇编程序汇编成机器语言程序；如果目标程序是用某种中间语言表示的，则该目标程序可以进行由下面讨论的解释器解释执行或再经编译后执行。例如，PASCAL P_编译器所产生的中间代码叫 P—代码，这种 P—代码程序可以解释执行也可以再编译后执行。

编译途径一般可分为三个阶段，即编译阶段、连接装配阶段和运行阶段。在编译阶段，由编译程序扫描源程序并将其翻译成目标程序。当编译程序的输出是待装配的目标程序模块时，需经过连接装配阶段，即由连接装配程序把目标程序（模块）以及必须的运行子程序连接起来形成可执行的机器代码，如图 1-3 所示。

一般将编译过程分成词法分析、语法分析、语义分析和中间代码生成、代码优化及目标代码生成五个阶段，如图 1-4 所示。

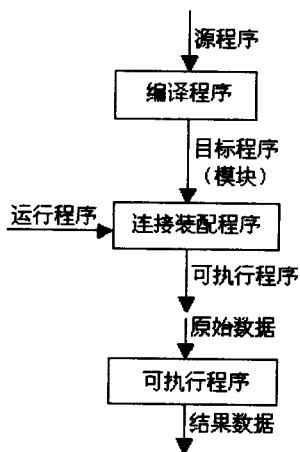


图 1-3 编译途径

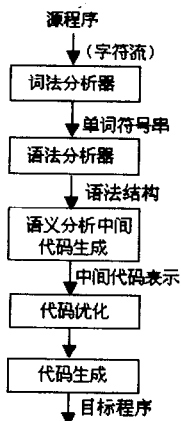


图 1-4 一般编译过程

此外，编译过程中还有建表和查表工作、进行出错处理两部分工作，图 1-5 是编译程序的一种典型的逻辑结构。

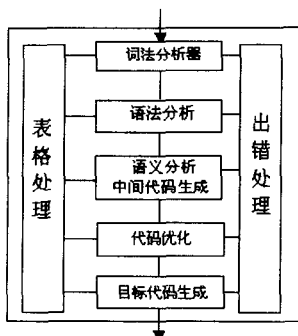


图 1-5 编译程序逻辑结构

- 词法分析：词法分析程序又称扫描器，它对源程序进行自左向右扫描，识别出一个一个单词（保留字、标识符、常数、运算符和界限符等），并将它们转换成相应的机内表示，从而形成相应的单词符号串（也报告词法错误）。
- 语法分析：语法分析程序又称分析器，它能根据程序设计语言的语法规则，将单词符号序列组合成（识别出）各种语法成分（如表达式、说明语句、分程序和程序等），在组合成语法成分的同时，进行语法检查，即检查语法成分在语法结构上的正确性，这就是语法分析。
- 语义分析和中间代码生成：在语法分析基础上，再作语义分析。计算机语言的语义是要求机器完成的指定操作。为了表达这种语义，并方便代码优化和生成目标程序，通常使用一种中间语言（它介于源程序语言和机器语言之间）来表示。语义分析的任务：一是进行语义正确性检查，二是进行语义处理并生成相应的中间代码。语义处理的任务根据语法成分不同而异。例如，对说明语句，则将其中定义的名字和有关信息写进符号表，并分配存储单元；对赋值语句，则要根据其语义生成相应的中间代码。常用的中间代码有逆波兰式、三元式和四元式等。由中间代码可以方便地生成目标指令。
- 代码优化：优化是对中间代码进行加工变换，以便获得高质量的目标程序。
- 目标程序生成：根据中间代码和表格中的信息，选择适当的机器指令，进行寄存器分配，确定各数据的存储位置，最后形成目标程序。

编译过程中需要很多表格，以登记源程序所提供或编译中产生的信息，如符号表、常数表、标号表和过程表。表格的建立、更新和查询由表格管理程序完成。

源程序中出現错误是难免的，源程序错误分三类：词法错误、语法错误和语义错误。检查和报告错误以及改正错误是编译程序的主要功能之一，这部分工作由出错处理程序完成。

3. 解释程序

解释途径是把翻译和运行合并在一起，称为解释执行阶段，如图 1-6 所示。这种方式并不形成整个目标程序而是按照源程序中语句的动态顺序逐句地进行解释并立即执行。完成此项工作的语言处理程序称为解释程序。它的输入是源程序和原始数据，输出是源程序

的计算结果而不是目标程序。

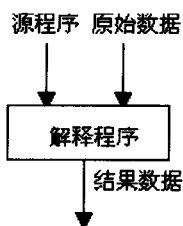


图 1-6 解释途径

解释程序的实际工作是复杂的。通常，解释程序对源程序进行词法和语法检查；把源程序经“换码”转变成另一种较便于处理的内部形式；还要建立一些表格，如标号表、标识符表等。解释程序的下一阶段任务是运用前一阶段提供的表格，对内部形式的源程序进行解释并执行。

在编译和解释两种方式中，解释器比编译器灵活。解释器执行一个源程序时，它有机器的控制权，它能在发现错误时停止源程序的执行并指示出错行；而执行一个由编译器产生的目标程序，发生错误时，程序只能简单退出或打印一些运行错误信息。调试时，解释器可以根据调试要求执行。目前一些高级的程序开发环境使编译器具有解释器的一些特性，可以编译一个程序并在“环境”的控制下执行，如果发生错误，你会得到提示并根据控制来改正错误，然后可以选择重新执行还是再次编译后执行。

就执行速度而言，编译器是绝对优先的，其速度可以是解释器的 10 倍乃至 100 倍。

1.3 计算机软件技术概述

1.3.1 程序设计

计算机软件都是由程序模块构成的。无论计算机所完成的工作多么浩繁、复杂或精细，它必须是按人们预先编好的程序进行工作。那么，怎样设计计算机程序呢？

通常认为，一个完整的程序设计过程可分为以下三个步骤：

- 分析问题，构造模型。
- 算法设计，过程描述。
- 程序编写，测试运行。

简单地可归结为“模型—算法—编码”三个步骤。

1. 模型（分析问题和构造模型）

每一个程序都应有明确的功能要求，即程序应解决的问题的内容、性质及规模。在程序设计开始时，必须分析那些问题，对理论上可遵循的公式、方法进行适当的修正，考虑量化和积累效应，使公式化为适应于计算机的计算模式，而对于非数值类的数据处理，更无现成的公式可循，完全要靠分析问题来构造模型了。实际上，许多应用领域的问题不能直接在计算机上解决，需要经过简化以后建立起相应的数学模型，才能上机运算。各种分

析问题、建立模型的方法已成为应用领域中引人注目的研究课题，有很多专著及文献可供参考。

2. 算法（算法设计和过程描述）

模型一旦建立起来，也就是确定了程序应该“做什么”，那么，在编写出程序之前首先要为它寻找一个“如何做”的算法，然后按算法编出程序。可以给算法下这样的定义：一个算法表示一个计算过程，它应该是明确的、确定的和有限的。它本身以能被执行的基本指令的形式给出。一般说来，运算结果的精确度及运算速度均与算法有着密切的关系。算法设计是一件较困难的工作，这方面的内容在“计算机算法”、“算法分析”等课程中专门的讲述。其中有一种行之有效的方法是逐步求精程序设计技术。

算法设计的同时，还要进行数据设计，包括选择数据类型，考虑数据的组织和存放形式，即数据结构。数据结构直接影响数据存取的速度，也影响处理的效率。随着计算机应用领域的日益广泛，计算机算法中愈来愈多地涉及到非数值型程序设计，其中最基本的技术是查找和排序。算法给出的是计算机“如何做”的过程中所包含的明确步骤，这种处理过程必须准确地描述出来。描述的方法很多，例如书面语言形式、流程图形式和类程序设计语言的形式。有了处理过程的描述就可以较方便地编写程序了。

3. 编码（程序编写和测试运行）

编写程序前先选定一种程序设计语言。一个好的又适于具体问题的语言可以使程序的结构清晰、简洁，可以正确地记述待解决的问题，同时还可以准确地表示过程，以便数据的抽象和模块化。

仅编制一个可运行的、结果正确的程序是不够的，应该努力提高程序设计的技术和改善程序的质量。一个高质量的程序，应具备以下条件：

- 建立正确的数学模型和确立有效的计算方法。
- 运行结果必须是正确的，且在精度及各方面均满足要求。
- 程序本身具有良好的结构，逻辑清楚，易读易懂。
- 程序运行时间尽可能短，同时尽可能合理地使用内存。
- 便于检查、修正、移植和维护。

程序要进行检查和调试，目的是查找和改正程序中存在的错误，使程序能顺序地运行。程序的调试是一项复杂的工作，是一项值得研究的内容，这也是软件工程中一个重要的环节。

1.3.2 数据结构

数据结构主要研究数据的组织方法以及相应的操作算法。它不仅是一般程序设计的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础，因此，它在计算机科学中占有重要的地位。对于相当多的程序设计问题来说，理清数据的内在关系可以获得对问题的正确观察角度，看清问题的结构，甚至解法的结构。

数据元素是数据的基本单位，是不可分割的。计算机所处理的数据一般是一组相同性质的数据元素，可以用集合表示。例如，一组包括整数 1~5 的数据元素可以表示为集合 $N=\{1, 2, 3, 4, 5\}$ 。