



普通高等教育“十一五”规划教材

Visual Basic程序设计教程

An Introduction to Visual Basic Programming

主 编 倪飞舟

副主编 陈浩强 丁亚涛 郭玉堂



中国水利水电出版社

www.waterpub.com.cn

普通高等教育“十一五”规划教材

Visual Basic 程序设计教程

主 编 倪飞舟

副主编 陈浩强 丁亚涛 郭玉堂



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书在总结编者多年教学实践、考试培训、软件开发经验的基础上,从教学和实用角度出发,采用“案例”驱动的编写方式,系统地介绍了 Visual Basic 的基础知识和程序设计方法。全书共 11 章,前 10 章分别对程序语言设计、VB 集成开发环境、VB 语言设计基础、VB 控制结构、数组和自定义类型、过程、用户界面设计、文件和数据库应用基础等方面进行介绍;第 11 章根据 Visual Basic 的发展趋势,简述了 Visual Basic 语言的新一代产品: Visual Basic.NET。

本书注重基础,强调实践,在内容讲解上采用循序渐进、逐步深入的方法;重点突出,案例取舍得当,注释语句丰富,方便教学和自学。尤其是在讲解语法和编程思路时,注重界面设计与算法设计的结合,突出了 Visual Basic 语言的特点和优势,是学习 Visual Basic 的适用教材。

本书适合作为高等院校本专科学生的教材,也可用作广大软件开发人员以及工程技术人员的参考用书。同时,本书也非常适合作为全国计算机等级考试的备考书。

本书配有免费电子教案,读者可以从中国水利水电出版社网站以及万水书苑下载,网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

Visual Basic 程序设计教程 / 倪飞舟主编. —北京:中国水利水电出版社, 2010.2

普通高等教育“十一五”规划教材

ISBN 978-7-5084-7204-1

I. ①V… II. ①倪… III. ①BASIC 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 019778 号

策划编辑: 雷顺加/徐海洋 责任编辑: 张玉玲 加工编辑: 徐 雯 封面设计: 李 佳

书 名	普通高等教育“十一五”规划教材 Visual Basic 程序设计教程
作 者	主 编 倪飞舟 副主编 陈浩强 丁亚涛 郭玉堂
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16 开本 17.75 印张 442 千字
版 次	2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷
印 数	0001—4000 册
定 价	30.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

根据教育部非计算机专业计算机基础课程教学指导分委会对当代高等院校学生的要求,高等院校计算机基础教育是面向全体高等院校学生进行的信息技术教育,目标是把学生培养成为既懂专业知识又能够掌握计算机应用技能的复合型人才,为学生的专业学习和科学研究提供帮助。

程序设计是计算机基础教育中继计算机与信息技术基础教育之后的又一重要知识领域,而 VB 程序设计就是其中最重要的课程之一。

Visual Basic 6.0 是 Microsoft 公司推出的基于 Windows 环境的计算机程序设计语言,它继承了 Basic 语言简单易学的优点,又增加了许多新的功能。Visual Basic 6.0 采用面向对象与事件驱动的程序设计思想,使编程变得更加方便、快捷。

使用 Visual Basic 既可以开发个人或小组使用的小型工具,又可以开发多媒体软件、数据库应用程序、网络应用程序等大型软件,是国内外最流行的程序设计语言之一,也是学习开发 Windows 应用程序首选的程序设计语言。它非常适合于初学者理清复杂的软件系统结构,是学习面向对象程序设计的敲门砖。目前国内大多院校都将 Visual Basic 语言及程序设计作为非计算机专业的计算机程序设计公共课。

本书针对当前大学生的学习特点组织编写,简明、易学、实用,并借助作者多年教学和项目开发的经验,指导读者全面了解 Visual Basic 程序设计的方法,培养其开展应用项目设计的基本技能。本书语言表达严谨流畅,逻辑性强,图文并茂,示例丰富。书中例题均在 Visual Basic 6.0 语言的程序开发环境下调试通过,关键性程序语句都标有详细的注释。

全套教材分为理论教学主教材和配套上机实验指导教材两部分。

理论教学主教材主要是从程序设计语言和可视化用户界面设计两个方面进行介绍,重点介绍程序结构和程序设计方法等基础问题,对程序设计的基本知识、基本语法、编程方法和典型算法等进行了较为系统、详细的介绍,让学生学会分析问题、掌握基本的编程方法。

配套上机实验指导教材则主要围绕 Visual Basic 6.0 语言的程序实际开发环境,对可视化用户界面设计的知识和技能展开实践性教学指导,紧密围绕理论教学的各知识点来安排实验项目,进一步巩固和提高学生对理论教学知识点的理解和掌握,增强其在实际程序开发环境下的实践能力。

理论教学主教材共 11 章,其中前 10 章分别对程序语言设计、VB 集成开发环境、VB 语言设计基础、VB 控制结构、数组和自定义类型、过程、用户界面设计、文件和数据库应用基础等方面进行了论述;第 11 章根据 Visual Basic 的发展趋势,简述了 Visual Basic 语言的新一代产品——Visual Basic.NET。

配套上机实验指导教材共 8 章,根据理论教学主教材各章知识点相应分成了 8 个实验,每个实验配有若干实验题。其中,第 4 章收录了程序语言设计中的多个经典算法与典型例题;第 5 章和第 6 章分别给出了理论教学主教材全部习题解答和配套实验教材全部习题操作详解;第 7 章考试指导部分对全国计算机等级考试和全国高校计算机水平考试进行了详细介绍,并

提供了最新考试大纲和考试样题；第 8 章为方便学生应试和自我检测给出了两套考试模拟题并附有答案。

本书由倪飞舟任主编，陈浩强、丁亚涛、郭玉堂任副主编。主要编写人员分工如下：第 1、8、11 章由倪飞舟、吴茜编写，第 2、3 章由陈浩强编写，第 4、7 章由杨雪洁编写，第 5、6 章由丁亚涛编写，第 9、10 章由周晓编写。全书由倪飞舟、郭玉堂负责统稿及定稿。

此外，感谢胡学钢、王浩、孟浩、张久彪、尹荣章、杜春敏、郭彪、徐启明、朱炜、许兆华、黄晓梅、董光锋、孙业桓、张锋、鲍丙刚、王家金、刘晓芳、杨载民、李苑青等同志在本书编写过程中提供的大力帮助。

本书可作为高等院校非计算机专业 Visual Basic 程序设计课程的教材，也可供从事计算机应用和开发行业的人员学习参考。另外，本书的编写也兼顾了教育部考试中心制定的《全国计算机等级考试二级考试大纲（Visual Basic 程序设计）》的要求，可作为全国计算机等级考试和全国高校计算机水平考试的培训教材。

为适应教学和读者自学的需要，我们还制作了与教材配套的电子教案，读者可以从中国水利水电出版社网站或万水书苑下载，网址为：<http://www.waterpub.com.cn/softdown/> 和 <http://www.wsbookshow.com>。

在本书的编写过程中作者参阅了大量国内外文献资料，在此向这些文献资料的作者们表示深深的谢意。由于作者水平所限，加之时间仓促，书中难免有不足之处，敬请各位专家和广大热心读者不吝指教。

编者

2010 年 1 月

目

录

前言

第1章 概述	1
1.1 程序设计语言概论	1
1.1.1 计算机组成和基本工作原理	1
1.1.2 计算机程序设计语言	2
1.1.3 计算机程序设计基本方法	5
1.2 VB语言的特点	9
1.2.1 VB语言的发展概况	9
1.2.2 VB语言的特点	9
1.3 VB程序的开发环境	10
1.3.1 VB 6.0的启动	11
1.3.2 VB集成开发环境(IDE)	12
1.4 VB程序设计步骤	16
1.4.1 一个简单的VB小程序	16
1.4.2 VB程序设计的步骤与书写规则	19
1.4.3 VB的程序调试	19
1.4.4 程序举例	20
习题	23
第2章 VB语言设计基础	24
2.1 面向对象编程基础	24
2.1.1 对象和控制	25
2.1.2 对象的属性和方法	26
2.1.3 对象的事件和事件过程	30
2.2 窗体	37
2.2.1 窗体的组成、创建和移除	37
2.2.2 窗体的属性	39
2.2.3 窗体的事件	41
2.2.4 窗体常用的方法	43
2.3 数据类型	44
2.3.1 基本数据类型	44
2.3.2 数据类型说明符	49
2.3.3 自定义数据类型	49
2.4 常量与变量	50
2.4.1 常量	50

2.4.2 变量	52
2.5 运算符与表达式	54
2.5.1 算术运算符与算术表达式	55
2.5.2 字符串运算符和字符串表达式	57
2.5.3 日期运算符和日期型表达式	58
2.5.4 关系运算符和关系表达式	58
2.5.5 逻辑运算符与逻辑运算表达式	59
2.5.6 混合表达式	59
2.6 常用的内部函数	60
2.6.1 数学函数	61
2.6.2 字符串函数	62
2.6.3 日期函数	63
2.6.4 数据类型转换函数	64
2.6.5 格式输出函数	64
2.6.6 Shell函数	66
2.7 VB中的语句	67
2.7.1 语句编写规则	67
2.7.2 赋值语句	68
2.7.3 流程控制语句	69
习题	71
第3章 顺序结构	72
3.1 数据输入	73
3.1.1 通过InputBox函数输入数据	73
3.1.2 通过文本框输入数据	74
3.2 数据输出	75
3.2.1 用标签输出数据	75
3.2.2 利用文本框输出数据	76
3.2.3 用MsgBox函数输出数据	77
3.2.4 利用Print方法输出数据	79
3.3 标签	82
3.4 文本框	83
3.5 命令按钮	85
3.6 程序举例	87

习题	91	6.4 控件数组	141
第4章 选择结构	92	6.5 自定义类型	143
4.1 If语句	92	6.6 字符串的处理	145
4.1.1 If-Else结构	92	6.7 列表框	150
4.1.2 If结构	94	6.8 组合框	151
4.1.3 ElseIf结构	94	6.9 程序举例	154
4.1.4 If语句的嵌套	96	习题	160
4.1.5 IIf函数	98	第7章 过程	162
4.2 Select Case语句	99	7.1 概述	162
4.3 框架	100	7.2 子过程	163
4.4 单选按钮	101	7.2.1 子过程的定义	163
4.5 复选框	103	7.2.2 子过程的调用	164
4.6 程序举例	105	7.3 函数过程	165
习题	109	7.3.1 函数过程的定义	165
第5章 循环结构	113	7.3.2 函数过程的调用	166
5.1 While-Wend语句	113	7.4 事件过程	168
5.2 Do-Loop语句	115	7.5 参数传递的方式	170
5.3 For-Next语句	117	7.5.1 传值	170
5.4 流程转向语句	119	7.5.2 传引用	172
5.4.1 Exit语句	119	7.5.3 传数组	174
5.4.2 Goto语句	120	7.6 嵌套调用与递归调用	176
5.5 循环嵌套	121	7.6.1 嵌套调用	176
5.6 循环算法	122	7.6.2 递归调用	177
5.6.1 穷举法	122	7.7 作用域与生存期	179
5.6.2 迭代法	124	7.7.1 作用域	179
5.7 图片框	125	7.7.2 生存期	181
5.8 图像框	127	7.8 滚动条	182
5.9 计时器	127	7.9 直线和形状	185
5.10 程序举例	129	7.9.1 直线	185
习题	131	7.9.2 形状	186
第6章 数组	133	7.10 程序举例	188
6.1 一维数组	133	习题	192
6.1.1 一维数组的定义	133	第8章 界面设计	193
6.1.2 数组元素的引用	134	8.1 对话框	193
6.1.3 数组的应用	134	8.1.1 通用对话框	193
6.2 二维数组	136	8.1.2 自定义对话框	199
6.2.1 二维数组的定义	136	8.2 菜单	200
6.2.2 二维数组的应用	137	8.2.1 下拉式菜单	200
6.3 动态数组	138	8.2.2 弹出式菜单	203

8.3 多重窗体.....	203	10.2.1 创建 Access 格式数据库.....	236
8.3.1 窗体添加和启动.....	204	10.2.2 数据窗体设计器.....	240
8.3.2 窗体操作.....	205	10.3 ADO 数据控件.....	241
8.4 ActiveX 控件.....	207	10.3.1 ADO 的对象与集合.....	242
8.4.1 进度条.....	207	10.3.2 添加 ADO.....	243
8.4.2 选项卡.....	209	10.3.3 ADO 应用.....	243
8.4.3 列表视图.....	210	10.4 数据及数据绑定控件.....	246
8.4.4 树形视图.....	213	10.4.1 Data 控件.....	246
8.4.5 Animation 控件.....	215	10.4.2 通用数据绑定控件.....	247
习题.....	217	10.4.3 专用数据绑定控件.....	249
第 9 章 文件.....	218	10.5 SQL 简介.....	251
9.1 概述.....	218	10.5.1 SQL 语言的特点.....	251
9.2 文件打开与关闭.....	218	10.5.2 SQL 语言对数据库的操作.....	252
9.2.1 文件打开.....	219	10.6 报表制作.....	254
9.2.2 文件关闭.....	219	10.7 程序举例.....	256
9.3 文件读写.....	220	习题.....	259
9.3.1 顺序文件.....	220	第 11 章 VB.NET 简述.....	260
9.3.2 随机文件.....	224	11.1 概述.....	260
9.4 文件操作.....	227	11.2 Visual Basic.NET 的集成开发环境.....	263
9.4.1 文件操作语句.....	227	11.3 Visual Basic.NET 应用程序的 开发步骤.....	265
9.4.2 文件操作函数.....	228	11.4 Visual Basic.NET 的数据类型.....	265
9.5 文件系统控件.....	228	11.5 Visual Basic.NET 的控件.....	266
9.5.1 驱动器列表框.....	228	11.6 Visual Basic.NET 的基本特性.....	267
9.5.2 目录列表框.....	229	习题.....	270
9.5.3 文件列表框.....	229	附录 1 常用字符与 ASCII 码对照表.....	271
习题.....	231	附录 2 常用的内部函数.....	272
第 10 章 Visual Basic 与数据库.....	233	附录 3 Visual Basic 6.0 与 Visual Basic.NET 中 的菜单等效项.....	274
10.1 概述.....	233	参考文献.....	276
10.1.1 数据库.....	233		
10.1.2 数据访问对象模型.....	235		
10.2 数据管理器.....	236		

第1章 概述



本章导读

自第一台计算机于 1946 年诞生至今, 已有了半个多世纪。计算机堪称是 20 世纪人类最伟大的一项技术发明, 它是人类大脑的延伸, 使得人类的智慧和创造力能够得到充分的施展。作为先进生产力的典型代表, 计算机技术及其应用已经渗透到社会生活的各个领域, 遍及世界的各个角落, 已经成为人们学习、工作和生活的得力助手, 有力地推动了整个信息化社会的发展。在 21 世纪, 掌握以计算机为核心的信息技术知识与技能, 是现代大学生必备的基本素质。

通过对计算机基础的学习我们已知道, 计算机是通过执行程序来完成各种各样的工作的, 由于计算机目前还不能够理解人类的自然语言, 因此编写程序只能借助于某种程序设计语言。

本章主要介绍程序设计语言的概念和 Visual Basic 语言的特点, 以及 Visual Basic 程序的开发环境等内容, 使读者对该语言有一个初步的感性认识。



本章要点

- 计算机程序设计语言概述
- 计算机程序设计基本方法
- VB 集成开发环境
- VB 程序设计的步骤与书写规则

1.1 程序设计语言概论

1.1.1 计算机组成和基本工作原理

计算机是在程序控制下对数字化信息进行存储和加工的电子设备, 它具有高速、精确和自动等特点。

一个完整的计算机系统由硬件系统和软件系统两部分构成。硬件系统是组成计算机系统的各种物理设备的总称, 是计算机系统的物质基础, 如 CPU、存储器、输入设备和输出设备等。软件系统是为运行、管理和维护计算机而编写的各种程序、数据和文档的总称, 可以说它是整个计算机系统的灵魂。实际上, 用户所面对的是经过若干层软件“包装”的计算机, 一台计算机的功能不仅仅取决于硬件系统, 而更大程度上是由所安装的软件系统所决定的。

在计算机系统中,对于软件和硬件的功能划分并没有一个明确的分界线。软件所实现的功能可以用硬件来实现,称为硬化或固化,例如在 ROM 芯片中就固化了系统的引导程序;同样,硬件所实现的功能也可以用软件来实现,称为硬件软化,例如多媒体计算机中的一些媒体播放软件,就实现了视频卡对于视频信息的处理。

计算机的工作过程实际上就是快速地执行指令的过程。指令是能被计算机识别并执行的二进制代码,它规定了计算机能够完成的某一种操作。一条指令通常是由操作码和操作数两个部分组成。当计算机在工作时,有两种信息在执行指令的过程中流动:数据流和控制流。数据流是指原始数据、中间结果、结果数据、源程序等;控制流是由控制器对指令进行分析、解释后向各部件发出的控制命令,指挥各部件协调地工作。

计算机在运行时,CPU 从内存读出一条指令执行,指令执行完毕,再从内存中读出下一条指令执行。CPU 不断地读取指令、分析指令、执行指令,这就是程序的执行过程。程序是一组有序指令的集合,由某种程序设计语言编写而成,程序设计语言是人与计算机之间进行交流的工具。计算机的工作就是执行程序,即自动连续地执行一系列指令,从而指挥计算机“做什么”和“如何做”。一条指令的功能虽然有限,但是由一系列指令组成的程序所可完成的任务是无限的。

1.1.2 计算机程序设计语言

自然语言是人们进行交流的工具,对同一件事情的描述可采用不同的语言(如中文、英语等)进行交流。而计算机程序设计语言则是人与计算机间进行交流的工具,是人们用于书写计算机程序的工具。程序开发人员的工作就是根据任务需要和特点选用不同的计算机程序设计语言去编制相应的程序。

计算机程序设计语言(Programming Language)是人们为描述计算过程而设计的一种具有语法语义描述的记号。对于计算机操作人员而言,程序设计语言是除计算机本身之外的所有工具中最重要工具,是其他工具的基础。没有程序设计语言的支持,计算机无异于一堆废料。从计算机问世至今,人们一直在为研发更新更好的程序设计语言而努力。目前程序设计语言的数量还在不断激增,新的程序设计语言和设计方法仍不断出现。已问世的程序设计语言有成千上万种,但只有其中的极少数得到了人们的广泛认可。

程序设计语言种类繁多,发展迅速。从其发展历史和应用特点来看,大致可以分为以下几个阶段。

1. 面向机器的程序设计语言

1944~1946 年间宾夕法尼亚大学的 Mauchly 与 Eckert 研制出了 ENIAC 电子数字积分计算机。1949 年,普林斯顿大学的 Von Neumann 研制出了 EDVAC 电子离散变量计算机,这是世界上第一台能把可执行的程序放在磁芯中的计算机,从此现代意义上的计算机便真正诞生了。这台计算机一问世便奠定了所谓的 Von Neumann 式体系结构:程序与数据不加区别地存储在(磁芯)存储器中。它所能处理的最基本的信息单位就是二进制数字,即计算机只能识别理解由 0 与 1 构成的二进制序列。那时人们要用二进制机器代码编成代码序列(即程序)来控制计算机执行规定的操作,这种代码就是第一代程序设计语言,也称机器语言。

早期的计算机程序都是直接用机器语言编写的。机器语言是计算机能够直接执行的二进

制指令代码，每条指令都用 0 和 1 组成的序列串表示，这些指令的集合就是指令系统。机器语言的特点是它能直接反映计算机的硬件结构，用其编写的程序不需作任何处理即可直接输入计算机执行，因而运行速度很快。但编写机器语言程序是一种非常枯燥而繁琐的工作，要记住每一条指令的编码与含义极端困难，编写出的程序既不易阅读也不易于修改。而且由于机器语言是特定于机器的，不同的机器有不同的指令系统，人们无法把为一种机器编写的程序直接搬到另一种机器上运行。一个问题如果要在多种机器上求解，那么就要对同一问题重复编写多个应用程序。由于机器语言程序直观性差，与人们习惯使用的数学表达式及自然语言差距太大，故而难学、难记，程序难以编写、调试、修改、移植和维护，限制了计算机的推广。

由于这一问题，人们设想能不能使用更接近于自然语言与数学语言的语言。经过努力，在 20 世纪 50 年代，第二代语言——汇编语言问世了。汇编语言也是一种面向机器的程序设计语言，它用助记符号来表示机器指令的操作符与操作数（亦称运算符与运算对象），如用“ADD”与“MOVE”分别表示机器语言中的加法与代码移动操作。汇编指令与机器指令之间是一对一的关系，汇编语言程序要经过特定的翻译程序（即汇编程序）将其中的各个指令逐个翻译成相应的机器指令（二进制指令代码）后才能执行，因而汇编语言程序运行效率较机器语言略低。但汇编语言的问世毕竟使人们在编写程序时不必再花较多的精力去记忆、查询机器代码与地址，因而编程工作较使用机器语言变得容易多了。

尽管与机器语言相比，汇编语言的抽象程度要高得多，但由于它们之间是一对一的关系，哪怕用汇编语言编写一个很简单的程序，也要使用数百条指令。为了解决这个问题，人们又研制出了宏汇编语言，一条宏汇编指令可以翻译成多条机器指令，这使得人们的程序设计工作量得以减轻。为了解决由多人编写的大程序的拼装问题，人们又研制出了连接程序，它用于把多个独立编写的程序块连接组装成一个完整的程序。虽然汇编语言比机器语言好学、好记、好用，也容易进行维护，但由于汇编语言一般都是针对特定的计算机或计算机系统设计的，因此它对机器的依赖性仍然很强，它的低级性也使得人们不易于用它编写较大的程序，而且在机器语言中存在的其他许多问题在汇编语言中也没有得到很好的解决。

机器语言和汇编语言都是面向机器的程序设计语言，它们与计算机的硬件紧密相关。不同类型的计算机往往有着不同的指令系统和汇编语言，用面向机器的语言编写的程序，一般是为特定的计算机硬件系统专门设计的。这样的程序可读性和可移植性很差，不仅如此，还要求程序员具有足够的计算机知识，熟练掌握所编程机器的指令系统。

2. 面向过程的程序设计语言

世界上第一个高级程序设计语言是在 20 世纪 50 年代中期由 John Backus 领导的一个小组研制的 FORTRAN 语言。这种语言与人类的自然语言和习惯使用的数学公式都比较接近，编写出的程序有严格的书写格式，结构严谨。FORTRAN 语言和随后出现的 BASIC 语言、PASCAL 语言、COBOL 语言以及 C 语言等，都被称为高级语言。程序员在使用高级语言编写程序时，不需要熟悉计算机的指令系统，可以将精力集中于解题思路和方法上。计算机显然不能直接执行高级语言程序，必须先翻译成为机器语言程序之后才能执行。

高级语言的一条语句相当于多条汇编指令或机器语言指令，表达能力强而且容易理解和书写。高级语言在程序设计时注重问题域中过程的描述和实现，因此又称为面向过程的程序设计语言。用这种语言编写的程序不依赖于具体的机器，可以很方便地在不同类型的计算机

中移植。高级语言采用结构化程序设计思想,将任务自顶向下逐步细化,划分为一些易于理解的功能模块,并确定模块之间的调用关系。在实现这些模块时,将控制结构限制为顺序结构、选择结构和循环结构。程序由这三种基本结构组合而成,每一种基本结构只有一个入口和一个出口,如图 1-1 所示。综上所述,面向过程的程序设计语言显著地降低了编程的难度和强度,改善了程序的可靠性和可维护性,提高了程序开发的效率。用面向过程的程序设计语言编写的程序,逻辑结构清晰,层次分明,易于实现。

虽然用高级语言编写的程序易学、易读、易修改,通用性好,不依赖于机器,但计算机却不能直接运行其编制的程序,必须经过语言处理程序的翻译或解释后才能被机器接受。因此,高级语言程序的执行速度通常比不上机器语言。

3. 面向问题的程序设计语言

面向问题的程序设计语言又称为非过程化的程序设计语言或第四代程序设计语言(Fourth Generation Language,以下简称 4GL)。4GL 这个词最早是在 20 世纪 80 年代初出现在软件厂商的广告和产品介绍中。因此,这些厂商的 4GL 产品不论是在形式上还是功能上,差别都很大。但是,由于这一类语言具有“面向问题”、“非过程化程度高”等特点,可以成数量级地提高软件生产率,缩短软件开发周期,因此赢得了很多用户。

4GL 是与第三代语言及其他新型设计语言同时发展的,它与前三代语言的主要区别在于侧重于描述程序“做什么”而不是“如何做”,函数式语言与逻辑式语言均有此特征,而高级语言与低级语言相比,也显现出一些这种特征。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次软件系统的开发环境,如报表生成、多窗口表格设计、菜单生成系统、图形图像处理系统和决策支持系统,为用户提供了一个良好的应用环境。它提供了功能强大的非过程问题手段,用户只需要告知系统做什么,而无须说明怎么做,因此可以大大提高软件生产率。

20 世纪 90 年代以来,随着计算机硬件技术的发展和应用水平的提高,以及诸如数据库系统、电子表格软件、统计软件包及其他专用软件包越来越多的使用,大量基于数据库管理系统的 4GL 商品化软件已在计算机应用领域获得了广泛应用,成为面向数据库应用开发的主流工具,如 Oracle、Informix24GL、SQL Windows、Power Builder 等。它们为缩短软件开发周期,提高软件质量发挥了巨大的作用,为软件开发注入了新的生机和活力。

例如最著名的数据库查询语言 SQL,它是 IBM 公司开发的一种关系数据库查询语言。程序生成程序或应用生成程序一般都基于某种特定的规格说明方法并能按一定的规格说明产生相应的输出(如高级语言程序)。目前已有大量第四代语言问世,如 ADF、IDEAL、NATURAL、NOMAD、MANTIS、MAPPER、RAMIS 等。

4. 面向对象的程序设计语言

结构化程序设计思想虽然有着诸多优点,但是它与人们在现实生活中自然形成的思维方式和习惯存在一定的差距。结构化程序设计方法在设计程序时,过于突出过程的重要性,而把数据放在相对从属的地位。由于操作数据的过程与数据分离为相互独立的实体,大大降低了程序的可重用性和可维护性;而且随着软件规模的急剧膨胀,使得这些问题变得更加严重。

自 20 世纪 70 年代以来,软件设计思想又产生了一次革命,面向对象的程序设计思想和方法受到了越来越广泛的重视和应用。在此之前的高级语言几乎都是面向过程的,程序的执行是流水线式的,在一个模块被执行完成前,人们不能干别的事,也无法动态地改变程序的

执行方向。这和人们日常处理事物的方式是不一致的,人们希望发生一件事就处理一件事,也就是说,不能面向过程,而应是面向具体的应用功能,也就是对象(Object)。其方法就是软件的集成化,如同硬件的集成电路一样,生产一些通用的、封装紧密的功能模块,称之为软件集成块,它与具体应用无关,但能相互组合,完成具体的应用功能,同时又能重复使用。对使用者来说,只关心它的接口(输入量、输出量)及能实现的功能,至于如何实现的,那是它内部的事,使用者可完全不用关心。

面向对象的程序设计方法力求符合人们自然的思维习惯,运用类和对象的观点描述问题域,有效地降低了问题的难度和复杂性,然后用程序设计语言对问题进行描述和实现。面向对象的程序设计思想认为,现实世界由一些形形色色的对象组成,对象有自己的属性和方法,对象之间通过消息相互通信。将某些对象的共性进行抽象并加以描述,就形成了类。在继承原有类特性的基础上,还可以派生出新类。不同类的对象能够对同一个消息产生不同的响应,这就是多态性。

20世纪80年代中期之后,相继出现了许多面向对象的程序设计语言。这些语言大致可以分为两类:一类是纯面向对象语言,例如 Eiffel 语言和 Java 语言;另一类是混合型面向对象语言,它们往往是由面向过程的语言发展而来的,例如 C++语言。Visual Basic 语言虽具有面向对象程序设计思想的一些要素,但由于其前身是 BASIC 语言,因此从这个角度出发,Visual Basic 语言也可以算作是一种混合型面向对象语言。

5. 面向应用的程序设计语言

高级语言未来的下一个发展目标是面向应用的程序设计语言。也就是说,只需要告诉程序你要干什么,程序就能自动生成算法,自动进行处理,这就是非过程化的程序语言。

总之,目前各种风格、各种类型的程序设计语言都在发展,其中以面向对象语言的发展最为迅速。但是,面向对象语言不能解决所有问题,它们也应吸收其他风格的语言的长处,成为以面向对象为主、兼顾其他风格的新型语言。随着软件技术的发展,还会有新风格语言问世,一些具有特定风格的语言也会不断扩充,以增强其自身的生命力。

1.1.3 计算机程序设计基本方法

程序设计的过程要消耗大量的人力,为了提高程序开发的效率,便于对程序的维护,许多年来,人们一直在研究程序设计的方法。目前最常用的是结构化程序设计方法和面向对象的程序设计方法。

1. 结构化程序设计方法

在计算机刚出现的早期,它的价格昂贵、内存很小且速度不高。程序员为了在小的可怜的内存下解决大量的科学计算问题,并为了节省昂贵的 CPU 机时费,不得不使用巧妙的手段和技术,手工编写各种高效的程序。其中显著的特点是在程序中大量使用 GOTO 语句,使得程序结构混乱、可读性差、可维护性差,通用性就更差了。

结构化程序设计的概念最早由荷兰科学家 E.W.Dijkstra 提出,1966 年他就指出:可以从高级语言中取消 GOTO 语句;程序的质量与程序中所包含的 GOTO 语句的数量成反比;任何程序都要基于顺序、选择和循环三种基本的控制结构;程序具有模块化特征,每个程序模块具有唯一的入口和出口,如图 1-1 所示。这为结构化程序设计的技术奠定了理论基础。

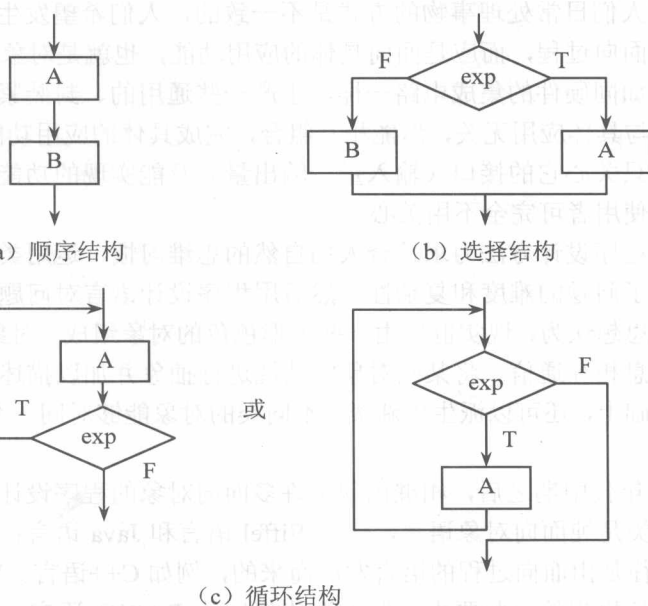


图 1-1 三种基本控制结构

结构化程序设计方法主要包括两个方面：

- (1) 在软件设计和实现过程中，提倡采用自顶向下、逐步细化的模块化程序设计原则。
- (2) 在代码编写时，强调采用单入口单出口的三种基本控制结构（顺序、选择、循环结构），避免使用 GOTO 语句。

结构化程序的结构简单清晰，可读性强，模块性强，描述方式符合人们解决复杂问题的普遍规律，在软件重用性、软件维护等方面有所进步，可以显著提高软件开发的效率。因此，在应用软件的开发中发挥了重要的作用。

2. 面向对象程序设计方法

面向对象的程序设计（Object Oriented Programming, OOP）起源于 Smalltalk 语言。用面向对象的方法解决问题，不是将问题分解为过程，而是将问题分解为对象。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。

当然，面向对象的程序设计并不是要抛弃结构化程序设计方法，而是站在比结构化程序设计更高、更抽象的层次上去解决问题。当所要解决的问题被分解为低级代码模块时，仍需要结构化编程的方法和技巧，但是，它在将一个大问题分解为若干小问题时所采取的思路却与结构化方法是不同的：

(1) 结构化的分解突出过程——如何做（How to do）？

它强调代码的功能是如何得以完成。

(2) 面向对象的分解突出真实世界和抽象的对象——做什么（What to do）？

它将大量的工作由相应的对象来完成，程序员在应用程序中只需说明要求对象完成的任务。

面向对象的程序设计给软件的发展带来了以下进步：

- 符合人们习惯的思维方法，便于分析复杂而多变化的问题。
- 易于进行软件的维护和功能的增减。
- 可重用性好，能用继承的方式减短程序开发所花的时间。
- 与可视化技术相结合，改善了工作界面。

目前在 Windows 环境下常用的面向对象的程序设计语言有 C++、Java、Visual Basic 等。虽然风格各异，但都具有共同的思维和编程模式。

初次接触面向对象的程序设计，经常会听到以下这些术语，因此了解这些术语将会对后续课程的学习十分有帮助。

(1) 对象 (Object)。我们可以将对象想像成日常生活中的某个实在的物体，如一辆汽车、一张桌子、一台计算机等都是对象；一个人、一篇文档、一份账单也是对象。总之，对象是人們要进行研究的任何事物，从最简单的整数到复杂的飞机等均可看作对象，它不仅能表示具体的事物，还能表示抽象的规则、计划或事件。任何对象都具有各自的特征、行为。比如人具有身高、体重、年龄、性别、视力等特征；也具有说话、行走、起立、做各项体育运动等行为。

在面向对象的程序设计中，对象的概念就是对程序所要求处理的现实世界中对象的模型化，它是数据和代码的组合，同样具有自己的特征和行为。对象的特征用数据来表示，称为“属性”；对象的行为用对象中的代码来实现，称为对象的“方法”。总之，任何对象都由属性和方法组成。在计算机中，对象可以是屏幕、打印机、窗体、数据库、命令按钮等。

将反映对象的属性和行为封装在一起，是面向对象编程的基本元素，也是面向对象程序设计的核心。

在 VB 中，常用的对象就是窗体和各种控件。除此之外 VB 还提供了其他一些对象，包括打印机 (Printer)、调试窗口 (Debug)、剪贴板 (Clipboard)、屏幕 (Screen) 等一些可访问的实体对象。

(2) 类 (Class)。类是创建对象实例的模板，是具有相同或相似性质的对象的集合与抽象，它包含所创建对象的属性描述和行为特征的定义。因此，对象的抽象是类，类的具体化就是对象，也可以说类的实例是对象。例如，人类是人的抽象，一个个不同的人是人类实例。各个人具有不同的身高、体重等属性值和不同的行为能力。

比如在 VB 中，工具箱上的可视图标是 VB 系统设计好的标准控件类，有命令按钮类、文本框类等多种。通过将控件类实例化，可以得到真正的控件对象，也就是当在窗体上画一个控件时，就将类转换为对象，即创建了一个控件对象，简称为控件，如图 1-2 所示。在 VB 系统中，类分为容器类（如窗体）和控件类（如命令按钮）两种。

(3) 属性 (Property)。属性是用来表示对象的特征。每个对象都有一组特定的属性，每个属性一般都有一组默认值，用户可根据需要改变属性值。不同的对象具有不同的属性。比如人都属于人类，但作为其对象的每个人所包含的诸如姓名、身高、体重、年龄等属性虽相同，但却因各自有着不同的属性值从而得以区分。至于不同类的对象，属性的差别就会更大了。

比如在 VB 中，窗体、文本框、命令按钮等都是对象，系统分别用一组特定的属性来描述它们。这些对象除了共同具有一些常规属性外（比如名称等）外，还各自拥有一些只有自己所特有的一些专用属性。

(4) 消息 (Message) 和方法 (Method)。对象之间进行通信的结构叫做消息。在对象的操作中, 当一个消息发送给某个对象时, 消息包含接收对象去执行某种操作的信息。发送一条消息至少应包括说明接受消息的对象名、发送给该对象的消息名 (即对象名、方法名)。一般还要对参数加以说明, 参数可以是认识该消息的对象所知道的变量名, 或者是所有对象都知道的全局变量名。

类中操作的实现过程叫做方法, 是对对象的属性的各种操作。系统将一些通用的过程或函数预先编写好并封装起来, 作为方法直接供用户调用。一个方法包括方法名、参数和方法体。

例如 Visual Basic 中的 Print 就是一种方法, 是用来输出信息的专用过程。

(5) 事件 (Event)、事件过程和事件驱动。

1) 事件。事件是面向对象程序设计中对应于消息的术语。对象的事件是指由系统事先设定的、能被对象识别和响应的动作。同一事件, 作用于不同的对象, 就会引发不同的反应, 产生不同的结果。例如在学校, 上课铃声就是一个事件, 教师听到铃声就要准备开始上课, 向学生传授知识; 学生听到铃声, 就要准备听教师上课, 接受知识; 而行政人员则不受影响, 就可不做响应。

在 Visual Basic 中, 系统为每个对象预先定义好了一系列的事件。例如, 单击 (Click)、双击 (Dbclick)、改变 (Change)、键盘按下 (KeyPress)、获取焦点 (Setfocus) 等事件。

2) 事件过程。当在对象上发生了事件后, 应用程序就要开始去处理这个事件, 而处理的步骤就是事件过程。一个对象可识别多个事件, 它是针对某一对象的过程, 并与该对象的一个事件相联系。面向对象程序设计的主要工作, 就是为对象编写事件过程。

例如上述上课响铃事件, 对于教师对象就是要编写授课的事件过程, 如打开多媒体设备、打开电子课件、开始组织内容上课等; 对学生对象就要去编写听课的事件过程, 如打开教材、打开笔记本、边听课边记笔记等。

3) 事件驱动。在传统的面向过程的应用程序中, 应用程序自身控制了执行哪一部分代码和按何种顺序执行代码, 即代码的执行是从第一行开始, 随着程序流程执行代码的不同部分。程序的执行的先后次序由设计人员编写的代码决定, 用户无法改变程序的执行流程。

在面向对象的程序设计中, 程序是由若干个规模较小的事件过程组成, 程序的执行发生了根本的变化。程序执行后系统等待某个事件的发生, 然后去执行处理此事件的事件过程。待事件过程执行完毕后, 系统又处于等待某一事件发生的状态, 这就是事件驱动程序设计方式。用户对这些事件驱动的顺序决定了代码执行的顺序, 因此应用程序每次运行时所经过的代码路径可能都是不同的。

(6) 封装 (Encapsulation)。将数据 (属性) 和操作数据的过程 (方法) 衔接起来, 构成一个具有类类型的描述称为封装。封装是一种信息隐蔽技术, 目的在于将对象的使用者和对象的设计者分开。用户只看到对象封装界面上的信息, 不必知道实现的细节。例如, 当需要打印文档时, 用户只要单击打印按钮就可将文档打印, 而不必了解打印按钮是如何与硬件通讯来打印文档的。因此, 对用户来说, 该打印按钮的细节就被封装起来了。

封装性可降低开发过程的复杂性, 提高了效率和质量, 也保证了程序中数据的完整性和安全性。

(7) 继承 (Inheritance)。表示类之间的相似性的机制。也就是可以从一个类生成另一个

类, 派生类(也称子类)继承了父类和祖先类的数据和操作。例如我们把“车”抽象为一个类, 则“汽车”、“摩托车”、“电动车”、“自行车”等都继承了“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、降低编码和维护的工作量。

(8) 多态性(Polymorphism)。多态性是指同一命名方法提供了多态性结果。也就是当同样的消息被不同的对象接受时, 却导致完全不同的行为。例如, “启动”是“车”类都具有的操作, 但“汽车”的“启动”是“发动机点火——启动引擎”, 而“自行车”的“启动”是指踩踏板。

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。

综上所述, 我们可以认为“面向对象=对象+类+继承+通信”, 如果一个软件系统是使用这样四个概念来设计和实现的, 我们就可以认为这个软件系统是面向对象的。

1.2 VB 语言的特点

1.2.1 VB 语言的发展概况

Visual Basic 语言(简称 VB)是一门编程语言, 是微软公司所开发的基于 Windows 操作系统平台下内含协助开发环境、可视化的、面向对象的、采取事件驱动方式的应用软件开发工具, 被广泛用于开发 Windows 环境下的各种应用程序。其语法基础是 Basic 编程语言。Basic 是 Beginners All-purpose Symbolic Instruction Code 的缩写, 含义为初学者通用的符号指令代码。Basic 语言简单易学, 拥有广大而又稳定的用户群, 对计算机的普及应用也起到了重要的作用。VB 几乎全盘接收了 Basic 语言的语法, 因而较易掌握; 并在此基础上增加了面向对象程序设计思想的一些要素, 以及可视化的编程工具和方法, 使得其功能更为强大, 成为编写 Windows 应用程序的一种利器。

Visual 的含义是可视化的, 是指一种开发图形用户界面(GUI)的方法。传统的高级语言只适合开发字符界面的软件, 在 Windows 环境下开发图形界面的软件, 因为需要建立窗口、对话框、控件和菜单等界面元素, 就显得力不从心了。可视化的程序设计语言保留了高级语言常规的编程功能, 并提供一系列可视化的设计工具, 使得程序员可以较为容易地建立各种各样的界面元素, 大大降低了 Windows 应用软件编程的复杂性。

微软公司于 1991 年推出 VB 1.0 版, 历经数年的更新换代, 1998 年升级为 VB 6.0 版, 并有学习版、专业版和企业版 3 种版本。为方便中国用户的使用, 微软公司从 VB 5.0 版开始, 同步推出 VB 的中文版。目前 VB 已经发展到了 VB.NET, 成为微软公司 .NET 技术战略的一个重要组成部分。本书以 VB 6.0 版为背景进行 VB 语言的讲解。

1.2.2 VB 语言的特点

VB 语言作为一种广泛使用的可视化程序设计语言, 主要有如下几个特点: