



河南省高等学校计算机教育研究会统编教材

Visual Basic 程序设计

郭清溥 主编 崔清民 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

河南省高等学校计算机教育研究会统编教材

Visual Basic 程序设计

郭清溥 主 编

崔清民 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是一本面向非计算机专业学生学习程序设计的入门教材。本书重点对程序设计的基础知识、基本语法、编程方法和常用算法进行了较为系统、规范和详细地介绍,目的是让学生学会并掌握运用计算机解决实际问题的能力。本书通过控件名、变量名等的规范使用来强调程序设计风格的重要性,并希望学生通过对本书的学习,既可以掌握程序设计的基本方法,又可以养成良好的程序设计风格。Visual Basic 是可视化的程序设计语言,本书将程序设计和可视化界面设计有机地结合起来,既能提高学生学习的积极性,又能提高编程效率和教学效果,真正达到学以致用目的。

本书配套的习题参考答案和实验指导将在《Visual Basic 程序设计习题解答与上机指导》一书中给出。本书可作为本科院校、高职高专院校为非计算机专业学生开设的第一门程序设计语言课程的教材。

图书在版编目(CIP)数据

Visual Basic 程序设计/郭清溥主编. —北京:中国铁道出版社, 2007. 1

河南省高等学校计算机教育研究会统编教材

ISBN 978-7-113-07787-7

I. V… II. 郭… III. BASIC 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 019617 号

书 名: Visual Basic 程序设计

作 者: 郭清溥 崔清民

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 李晶璞 郑 楠

封面设计: 薛 为

封面制作: 白 雪

责任校对: 郑 双

印 刷: 北京市兴顺印刷厂

开 本: 787×1092 1/16 印张: 19.25 字数: 443 千

版 本: 2007 年 3 月第 1 版 2007 年 3 月第 1 次印刷

书 号: ISBN 978-7-113-07787-7/TP·2152

定 价: 25.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

河南省高等学校计算机教育研究会

主任：段银田

副主任：甘 勇 普杰信 王贺明

秘书：李学相

委员：段银田 甘 勇 普杰信 王贺明 李学相

翁 梅 曲宏山 郭清溥 申石磊 周清雷

刘克成 陆桂明 程万里 马占欣 陈 涛

张东升 朱国华 李 敏 黄贻彬 商信华

连卫民 杨立峰 商其坤

自上个世纪 80 年代初到本世纪初的 20 年间, 由于计算机奇迹般地展示出它惊人的运算速度、海量的存储能力和神奇的创造性, 使人类社会深深地感受到了计算机的存在和它的不可或缺性。在这种背景下, 全国各类高等学校已陆续开展了计算机基础教育, 普及了计算机文化基础知识和技术基础知识。相应地, 这两种类型的教材也大量涌现, 为计算机教育和应用的普及提供了丰富的智力资源。

然而, 进入 21 世纪以来, 高等学校的计算机基础教育面临着新的挑战。首先, 一个时期以来, 信息技术自身愈来愈向技术多元化的方向发展。网络、数据库、多媒体等技术已从科学的殿堂里走了出来, 并日益得到应用和普及, 各种信息技术在工程中的综合应用程度越来越高, 这一切促使全社会计算机的应用水平提升到了一个新高度。应用的普及也推动了需求的进一步多样化, 社会也因此更加迫切需要实用型信息技术人才。在这种背景下, 大学现行计算机基础教育教材已远远不能适应技术发展和应用的要求。其次, 由于近年来中小学信息技术教育的普遍开展, 使得原本在大学要完成的信息技术学习任务的一部分已经提前完成, 因此也需要调整当前高等学校计算机应用基础教学的内容, 可见更新现行教材已成为当前一项十分紧迫的任务。作为高等学校计算机基础教育教材改革创作的尝试, 河南省高等学校计算机教育研究会与中国铁道出版社共同策划了这套系列教材。

本套教材的创作是以社会对信息技术的应用需求为目标, 学习的方向应瞄准应用, 学习的目的是能够做事的观念。要知道, 仅能在操作层面上使用计算机并不是真正意义上的应用, 开发才是真正的应用, 也就是常说的开发应用, 这也就是大学生学习信息技术的方向和应采取的行动。这些观念应逐步成为教材创作的指导思想。

突出信息技术教育的目的性是本系列丛书内容的最大特色。信息技术何其多! 究竟学什么、写什么? 要改变那种无的放矢的、包罗万象的教材创作模式; 要有目的的去写过程, 摒弃那种遍历知识过程就是一切、没有目的、文字堆砌式的创作观念和方法。应当明白, 学习信息技术是为了做事情, 而不是为了其他。此外, 计算机基础教育的教材要提倡精简。要树立信息量观点, 能够释疑解惑的文字构成信息量, 可以写入教材, 不能起到释疑解惑作用的文字或冗余文字只能形成垃圾信息, 应当从教材中剔除出去。

例如, 对于操作技能类的教材来说, 完全可以按“展示一种目的, 精讲一个案例, 完成一个练习, 创造一个作品”这四句话的要求来进行教材创作。对于程序设计类教材来说, 教材应逐步体现并满足从程序设计向软件设计延伸的社会需求。

在教材创作中，应努力完成相关知识的整合，这不仅是本套教材所提倡的创作特色之一，也是信息技术教育改革的出路所在。对于计算机基础教育来说，知识整合主要体现在两个方面。其一，用公用事件整合适用的信息技术。把面向社会大众所发生的信息技术应用事件用其所必须的信息技术，而不是某个领域的全部信息技术加以整合应用。把从目的到技术的逆向思维作为新一代信息技术教材创作的思维方法和行为方法。整个创作过程应按照“目的决定过程，过程决定事件，事件决定对象，对象决定技术”的思路进行。其二，信息技术与其他特定学科的相互整合。这种整合开辟了信息技术与专业相结合进行教材创作的途径。更加有利于实现从目的到技术进行教材创作的思想，使特定学科的内容和信息技术实现“我中有你，你中有我”，达到更高层次的融合。这种融合有利于双方共同提高教学效率，拓宽知识领域，增加知识深度，激发创造思维。总之，本丛书的创作特色主要体现在用目的、事件、对象去整合适用的信息技术。帮助读者为了达到目的而学会利用信息技术做一些实实在在的事情。

最后，本人深知新一代计算机基础教育教材的创作远不是一蹴而就的事情，目标的实现尚需时日。序言的目的仅在于简要阐明本套教材在策划过程中提出的一些基本思想和对创作的原则要求，正确与否还须经过实践的检验。望作者和读者在创作与实践中不断斧正。

河南省高等学校计算机教育研究会理事长

段银田

前言

FOREWORD

“程序设计”是目前高等院校非计算机专业的必修课程。随着计算机技术的进步，程序设计语言得到了飞速地发展。Visual Basic 是近年来发展最为迅速的面向对象的程序设计语言，由于它具有简单易学、方便快捷等特点，使得越来越多的高等院校将 Visual Basic 作为学生学习的第—种程序设计语言。

本书根据高级语言程序设计的基本体系，结合可视化程序设计的方法，以 Visual Basic 为背景介绍了程序设计的基本思想和基本方法、结构化程序设计的基本控制结构、可视化程序设计的基本方法和数据库的基础知识，并通过大量的例题介绍了 Visual Basic 的基本功能。本书是高级语言程序设计的基本教材，因此，它不涵盖 Visual Basic 的全部功能。有兴趣的读者，可以在初步掌握程序设计的基础知识后，根据不同的需要查阅相关资料。

本书在内容编排、叙述表达等方面都充分考虑到了初学者和非计算机专业学生在学习计算机知识时的基础性和实用性两大特点，尽量做到由易到难、循序渐进。并且在对 Visual Basic 的基础知识、基本语法进行介绍的同时，通过简单常用算法，加强对学生编程能力的培养，为学生打好程序设计的基本功提供了有力的帮助。

众所周知，在“程序设计”课程的教学—中必须以切合将来软件工程开发的实际需要为首要目标，简单地—说就是应该把“完成任务”作为教学的首要目标。为了能够很好地实现这一目标，就应该采用“任务驱动式”的教学方法。所谓“任务驱动式”教学就是以“任务”为目的，以语言理论教学为过程，最终可以用某种语言来设计“任务”，实现任务的要求。在“任务驱动式”教学中，首先应该让学生了解什么是软件工程思想，其次在语言理论教学过程中，让学生掌握面向过程、面向对象的程序设计的风格，最后引导他们运用软件工程的思想来设计任务。本书在对例题的讲解过程中尽量严格按照“程序设计”的步骤进行，采用“任务驱动”的教学方式，以培养学生分析问题的完备性及统筹全局、协调关系的基本素质，使学生树立面向工程进行程序设计的思想，为其将来从事计算机程序设计打下良好的基础。

具有良好的程序设计风格应该是程序员所具备的基本素质。另外，程序设计风格的好坏在很大程度上会影响到程序质量的好坏。好的程序设计风格不仅有助于提高程序的可靠性、可理解性、可测试性、可维护性和可重用性，而且还能够促进技术的交流、改善软件的质量。本书在例题的讲解中还特别通过控件名、变量名、书写格式等的规范使用，以强调代码风格和程序设计风格的重要性，目的是让学生在学会并掌握运用 Visual Basic 解决实际问题的同时，又能养成良好的程序设计风格。

“程序设计”是一门实践性很强的课程，要想得到程序设计方法和应用软件开发方面的基础训练，就必须亲自动手编写程序。本书针对初学者“听懂容易下手难”的特点，除了在每一章的最后都配有不同类型的思考与练习以外，还编写了配套的《Visual Basic 程序设计习题解答与上机指导》。该书不仅给出了《Visual Basic 程序设计》中的习题参考答案，同时还按照教材中的知识点给出了实验要求、实验内容和操作方法。

同时为了方便教师备课，我们还制作了与本教材配套的电子教案。如果读者需要可以同中国铁道出版社联系。

本书由郭清溥担任主编，崔清民担任副主编。其中范素凤编写了第1章；夏跃伟编写了第2章；邵国金编写了第3章；郭清溥编写了第4章、第5章；张琳编写了第6章；崔清民编写了第7章；张慧编写了第8章；王崇科编写了第9章；魏勇编写了第10章。

中国铁道出版社为本套教材的策划和出版做了许多工作；河南省高等学校计算机教育研究会的专家和教授在本套教材的编写过程中提出了很多有价值的建议。在此，对他们表示衷心地感谢！

由于时间紧迫，书中难免存在疏漏及不妥之处，在此，恳请广大读者批评指正（E-mail地址：gqp@hnufe.edu.cn）。

编者

2006年9月

第 1 章 Visual Basic 概述	1
1.1 程序设计的基本概念	1
1.1.1 程 序	1
1.1.2 程序设计	2
1.1.3 程序设计语言	4
1.2 对象的概念	5
1.2.1 对象和类	5
1.2.2 对象的属性、事件和方法	7
1.3 初识 Visual Basic 6.0	10
1.3.1 Visual Basic 6.0 的发展	10
1.3.2 Visual Basic 6.0 的特点	12
1.3.3 Visual Basic 6.0 的安装环境与安装过程	13
1.3.4 Visual Basic 6.0 的启动与退出	17
1.3.5 第一个 Visual Basic 程序	18
1.3.6 Visual Basic 应用程序的开发模式	24
1.3.7 Visual Basic 的工程概述	25
习题 1	25
第 2 章 Visual Basic 的集成开发环境	27
2.1 Visual Basic 集成开发环境概述	27
2.1.1 Visual Basic 集成开发环境的主窗口	27
2.1.2 对象窗口	28
2.1.3 代码窗口	29
2.1.4 工程资源管理窗口	29
2.1.5 属性窗口	30
2.1.6 工具箱窗口	30
2.2 窗体和基本控件	31
2.2.1 对象的基本属性	31
2.2.2 窗 体	32
2.2.3 命令按钮控件	34
2.2.4 标 签	36
2.2.5 文本框	38
2.3 Visual Basic 的帮助系统	45
习题 2	47
第 3 章 Visual Basic 语言规则	50
3.1 数据类型	50
3.1.1 字符型数据	51

3.1.2	数值型数据	52
3.1.3	布尔型数据	53
3.1.4	日期型数据	53
3.1.5	对象型数据	54
3.1.6	变体型数据	54
3.1.7	自定义型数据	55
3.2	常量、变量和表达式	56
3.2.1	常 量	56
3.2.2	变 量	58
3.2.3	运算符和表达式	62
3.3	常用内部函数	67
3.3.1	格式化输出函数	68
3.3.2	数学函数	70
3.3.3	字符串函数	71
3.3.4	日期函数	71
3.3.5	转换函数	72
3.3.6	测试函数	72
3.4	程序设计风格	73
3.4.1	编码规则	73
3.4.2	书写要求	75
习题 3		79
第 4 章	基本程序结构	82
4.1	顺序结构	82
4.1.1	赋值语句	82
4.1.2	Print 方法	85
4.1.3	人机交互	86
4.1.4	With 语句	94
4.2	选择结构	94
4.2.1	简单条件选择结构的程序实现	95
4.2.2	选择结构的嵌套	98
4.2.3	多条件选择结构的程序实现	99
4.2.4	条件函数	102
4.2.5	Go To 语句的程序实现	103
4.3	循环结构	105
4.3.1	条件循环的程序实现	106
4.3.2	固定循环次数的程序实现	111
4.3.3	不固定循环次数的程序实现	115
4.3.4	循环结构的嵌套	118
4.4	常用算法举例	120
4.4.1	穷举法	120

4.4.2 递推法	123
4.4.3 几种求方程近似根的方法	125
习题 4	127
第 5 章 数 组	129
5.1 数组的概念	129
5.1.1 数组和数组元素	129
5.1.2 数组说明语句	130
5.1.3 动态数组	130
5.2 控件数组	132
5.2.1 控件数组的概念	132
5.2.2 控件数组的建立	132
5.2.3 控件数组的使用	133
5.3 常用算法举例	134
习题 5	148
第 6 章 过 程	151
6.1 子过程	151
6.1.1 事件过程和通用过程	151
6.1.2 建立子过程	152
6.1.3 调用子过程	154
6.2 函数过程	155
6.2.1 建立函数过程	155
6.2.2 调用函数过程	157
6.3 过程中参数的传递	159
6.3.1 形式参数与实际参数	159
6.3.2 传值参数与传址参数	160
6.3.3 数组参数的传递	162
6.4 过程和变量的作用域	163
6.4.1 过程的作用域	164
6.4.2 变量的作用域	165
6.4.3 静态变量	166
6.5 过程的嵌套和递归	167
6.5.1 过程的嵌套	168
6.5.2 过程的递归	169
6.6 常用算法举例	171
习题 6	176
第 7 章 常用标准控件	178
7.1 单选按钮和复选框	178
7.1.1 单选按钮的属性和事件	178
7.1.2 复选框的属性和事件	178
7.2 框 架	181

7.3	列表框和组合框.....	182
7.3.1	列表框.....	182
7.3.2	组合框.....	185
7.4	滚动条.....	187
7.4.1	滚动条属性.....	187
7.4.2	滚动条事件.....	187
7.5	计时器.....	189
7.5.1	计时器控件的属性.....	189
7.5.2	计时器控件的事件.....	189
7.6	图形控件.....	191
7.6.1	图片框和图像框.....	191
7.6.2	直线和形状控件.....	192
习题 7		194
第 8 章	文件系统.....	197
8.1	文件系统控件.....	197
8.1.1	驱动器列表框.....	197
8.1.2	目录列表框.....	198
8.1.3	文件列表框.....	199
8.1.4	文件系统控件应用.....	201
8.2	数据文件的访问.....	202
8.2.1	文件的结构和分类.....	202
8.2.2	顺序文件.....	203
8.2.3	随机文件.....	208
8.2.4	二进制文件.....	211
8.3	常用的文件操作语句和函数.....	212
习题 8		220
第 9 章	界面设计.....	221
9.1	通用对话框.....	221
9.1.1	“文件”对话框.....	224
9.1.2	“打印”对话框.....	228
9.1.3	“帮助”对话框.....	229
9.2	菜单设计.....	230
9.2.1	菜单编辑器的使用.....	230
9.2.2	动态菜单.....	236
9.2.3	弹出菜单.....	239
9.3	多重窗体和多文档界面.....	241
9.3.1	多重窗体操作.....	242
9.3.2	多文档操作.....	245
习题 9		249

第 10 章 数据库技术	252
10.1 数据库概述	252
10.1.1 什么是数据库	252
10.1.2 关系数据库	252
10.1.3 数据库访问的常见术语	253
10.1.4 Visual Basic 中常用的数据库	254
10.1.5 Visual Basic 中的数据库访问方法	254
10.2 可视化数据管理器	255
10.2.1 启动可视化数据管理器	255
10.2.2 打开数据库	256
10.2.3 建立数据库	257
10.2.4 建立数据库中的表	257
10.2.5 输入和编辑数据	260
10.3 Data 控件	261
10.3.1 Data 控件简介	261
10.3.2 Data 控件的重要属性	261
10.3.3 数据绑定控件	262
10.3.4 数据控件的常用方法	265
10.3.5 数据库记录的增加、删除及修改操作	265
10.4 ADO 数据控件	269
10.4.1 ADO 对象模型简介	269
10.4.2 ADO 编程方法	270
10.4.3 使用 ADO 数据控件	273
10.4.4 ADO 控件上新增绑定控件的使用	276
10.4.5 使用窗体向导	279
10.5 使用 SQL 查询数据库	281
10.5.1 结构化查询语言	281
10.5.2 使用 SELECT 语句查询	282
10.5.3 添加和删除记录	285
习题 10	290
参考文献	291

第 1 章 \ Visual Basic 概述

Visual Basic 6.0 是面向对象的可视化程序设计语言，是一种功能强大的 Windows 应用程序的快速开发工具。本章主要介绍面向对象编程的基础知识，包括程序设计、类、对象、属性、事件与方法等基本概念，以及 Visual Basic 6.0 的安装、Visual Basic 6.0 集成开发环境和 Visual Basic 6.0 应用程序的基本开发步骤。

1.1 程序设计的基本概念

1.1.1 程 序

在日常生活中，我们经常会听到“程序”这个词，如一位老师在教育学生时说“做什么事都要有一定的程序”，法制类的电视节目上也常常会提到“××不符合法律程序”等，你若要问一个小学高年级的学生如何理解“程序”这个词，他也会说“程序就是先做什么事，然后做什么事，最后再做什么事”。事实上，程序就是完成或解决某一问题的方法和步骤。它包括两方面的内容：一是要做什么事；二是怎么做。计算机程序就是为使计算机完成一个预定的任务而设计的一系列的语句或指令，这组指令也称为软件。要让计算机实现一组操作，必须先编写程序，然后使计算机执行此程序，如做报表、完成计算、画图甚至玩游戏，如果不装入适当的程序，则无法完成操作。

如果没有程序，计算机就像一堆金属和塑料放在桌子上一样毫无用处，或像一个不会用大脑思考的植物人。像 DOS 和 Windows 操作系统被称为系统软件，它是由程序员事先设计好程序并装入计算机中的，它可以使其他被称为应用程序的程序在计算机中运行。应用软件可以让计算机做实际的很有用的事情。商业应用软件，如 Office 2000，Microsoft Visual Studio 和游戏等，也都是由大型软件公司的程序员所编写的，如著名的 Microsoft，Borland 公司等。

下面来看一个简单的 Visual Basic 6.0 程序的示例：

```
a=8  
b=9  
s=a*b  
Print s
```

该程序片断的作用是：先使变量 a 的值为 8，变量 b 的值为 9，然后将变量 a 和变量 b 相乘，将其乘积赋给变量 s，最后输出变量 s 的值。

程序实质上就是由这样一行行的代码所组成。程序中的每一行代码都可以称做一行语句，也可理解为让计算机完成一定功能的步骤或指令。程序就是由这样一些指令所组成。每一段相对独立的程序都具有一定的功能。

程序的长度或规模可大可小。上例即是一个很小的程序，最短的程序只有一行语句，程序的规模和内容完全取决于所要解决的问题。

传统的程序是由过程或函数组成，过程或函数由一系列的字符组成。谈不上图形用户界面，所以代码量相对较少，但阅读起来较为晦涩难懂，程序出错后的查错、修改工作也较为困难，“牵一发而动全身”。现在的程序使用复杂的“用户友好界面”，它涉及到多窗口、菜单、对话框等，当然，程序编制起来较以前复杂，代码量也较大，因为有 80% 以上的代码花费在了设计友好界面的工作上。为了简化程序员或程序设计爱好者的编程工作，越来越多优秀的、可视化的编程工具应运而生，Visual Basic 就是其中之一。它提供了良好的编程界面，使得程序员或程序设计爱好者借助该编程工具在最短的时间内、花费较少的代码就能编写出优秀的应用程序。这样的应用程序就是面向对象的应用程序，维护起来非常容易，查错、纠错只限于某一块儿，如某个窗体或某个事件过程，一般不影响应用程序的其他内容。

1.1.2 程序设计

什么叫“程序设计”？设计、编写及检查调试程序的过程称为程序设计。程序设计不仅指在纸上编写程序的过程，它应当还包括：分析问题确定解题方法、写出具体操作步骤、用高级语言编写程序、上机调试程序、反复修改程序、正式运行、整理程序运行结果和书写程序文档等全过程。

如果要处理一个数值计算问题，如求斐波那契数列，可按如下步骤进行：

(1) 首先分析问题，包括分析问题的要求、最后要得出什么结果、用计算机有无解决的可能。

(2) 建立数学模型（如列出一组方程式，这里可列出方程 $X_n = X_{n-1} + X_{n-2}$ ），在建立数学模型时可能要对问题做一些近似简化处理。

(3) 决定用什么方法去解该项数学模型，可以从时间和空间的利用率上来考虑从多个方法中选取一种较好的方法来解。

(4) 根据所选用的方法写出操作步骤，也就是计算机中通常所说的“算法”。

(5) 选择一种熟悉的且适用的计算机语言，并根据前一步骤所确定的算法编写出程序，即通常所说的“编码”。

(6) 在所选择的计算机语言的运行环境中上机编译、运行程序。编好程序还不能说已完成了程序设计任务，因为还未验证程序是否正确。必须上机运行程序，并且用各种不同的数据测试在不同情况下能否得出正确结果。

(7) 根据编译、运行情况分析是否存在语法错误、逻辑错误或运行时错误，以及结果是否有错。如果有错，则可能是解题方法有错、算法有错或编程序时出错。找出错误原因，返回前面修改。

(8) 修改后再调试、运行，直到得到正确的结果为止。

(9) 最后分析整理结果，写出程序文档。

有人说，现在有些软件用起来很方便，并不需要自己编写程序，那么为什么还要学习程序设计呢？在此应当说明以下几点：

(1) 要让计算机进行一系列的操作，必须用计算机能懂的语言编写程序。任何软件的主要部分都是程序，只不过那些商品软件是由别人编好了程序让用户直接使用而已。当前软件的代码越来越趋于复杂化、大型化，当然功能也越来越强，特别是在人机交互能力方面更趋于人性化，也使得有些人感到计算机高深莫测，不是人所能相比的。其实，计算机就是根据人的意志进行工作的，而人正是通过程序指挥计算机工作的。

(2) 学习程序设计有助于用户了解计算机的工作原理。例如,知道数据是怎样存储和输入/输出的,知道如何解决含有逻辑判断和循环的复杂问题,知道图形是用什么方法画出来的等。这样在使用计算机时就不但知其然而且知其所以然,能够更好地理解计算机的工作流程和程序的运行状况,为以后修改应用程序以适应新的需要打下了良好的基础。

(3) 程序设计是计算机应用人员的基本功。一个有一定经验和水平的计算机应用人员不应当和其他计算机的用户一样,只满足于能使用某些现成的软件,利用菜单和鼠标去选择现成的功能,它应当具有个人开发应用程序的能力。现成的软件不可能满足一切领域的多方面的需求,即使是满足当前需求的软件产品,随着时间的推移和条件的变化也会变得不适应。因此,计算机应用人员应当具备能够根据本领域的需要进行必要的程序开发工作的能力。

程序的设计方法经历了一个漫长而又复杂的过程,最早的程序设计是通过放置在计算机面板上的开关电路来完成的。由于当时条件的限制,显然,这种程序设计方法只适用于最简单的程序,不可能编写出复杂的应用程序。随着汇编语言的发明,才使编写较长的程序成为可能。直到 20 世纪 50 年代第一种高级程序设计语言 (Fortran) 的诞生,才使程序设计向前迈进了一大步。

最开始用高级语言进行程序设计,可以写出数千行长的程序,没有模块化、结构化之说。但随着时间的推移,这种早期的程序设计显现出其特有的局限性,一旦编制大型程序,像记流水账一样写出的源代码光是读起来都很困难,更不用说控制了。直到 20 世纪 60 年代出现了结构化程序设计语言,这种程序设计方法才被淘汰。结构化程序设计使用容易定义的控制模块、代码块,而不使用 (或很少使用) 任意转向语句 GO TO,支持递归并有局部变量的独立子程序。结构化程序设计的精髓在于将一个程序分解成若干个相互关联又彼此独立的模块。

尽管结构化程序设计方法在编写中等复杂度程序时表现出卓越的性能,可一旦程序长度达到某一限度,这种方法在某些方面还是不尽如人意。为了编写更为复杂的程序,又有一种新的程序设计方法,即面向对象程序设计 (Object Oriented Programming, OOP) 应运而生。

面向对象程序设计汲取了结构化程序设计中最精华的部分,并把它们与那些最新概念有机地融为一体。程序设计人员进行面向对象程序设计时,不再是单纯地从代码的第一行编到最后一行,而是通过考虑如何创建对象,利用对象来简化程序设计,提供代码的可重用性。面向对象程序设计技术鼓励程序设计人员把一个问题分解成几个相互关联的子问题。每个子问题都是一个自包含对象,其中的代码与数据都与这个对象密切相关。这样一来,问题的复杂度就大大降低了,程序设计人员因而能够管理更大的程序。

现在的面向对象程序设计工作因为有了可视化的开发环境而较以前有了极大地改变,并且随着不断增强的应用程序开发环境,编程工作也变得越来越轻松、有趣味。编程人员在编写应用程序时,既不需要编写大量的代码来描述组成用户界面的各种对象的外观和位置,也不需要反复地调试运行来调整它们,而只要把由开发工具提供的各种对象拖放到屏幕上并直接调整位置、大小、颜色、字体等即可。程序运行时所见到的结果就是设计时调整的结果,也就是常说的“所见即所得 (What You See Is What You Get, WYSIWYG)”。犹如搭积木式的可视化编程技术,大大提高了应用程序的开发效率,降低了程序员的劳动强度。Visual Basic 是典型的一个可视化的编程开发工具,利用该编程工具,单个程序开发

人员在很短的时间里就可以创建一个简单的应用程序，而一个开发小组则可以创建一个复杂、分布式的应用程序。

1.1.3 程序设计语言

如果可以用标准的中文句子或英语句子等自然语言来编写程序，那么程序员会节省很多的精力。而任何计算机只能真正理解一种语言（即机器语言），而这种语言与中文或英语等有很大的区别。计算机的“大脑”是一块称为中央处理单元（简称 CPU）的集成电路，CPU 只执行以机器语言形式存储的指令。机器语言是一系列由 0 和 1 两种数字或字符组成的二进制数码，这些二进制数码就代表给 CPU 的程序指令。由不同个数、按不同顺序组成的二进制数码都有其特有的含义和用法，只有记住了这些组成和含义，才能用机器语言编制程序。然而，要想记住这些谈何容易，中级水平的英语也就只需掌握 5000 左右的单词量，但光是由 8 个二进制位组成的排列个数就是 $8!$ ，超过了 4 万。假如一种排列就是机器语言中的一个有效单词的话，单词量是非常大的，靠人脑来记忆是远远不行的。鉴于机器语言学习起来困难并且不能由人们有效地用来编写程序，所以为使人们可以给 CPU 写指令而开发了程序设计语言。其实，程序设计语言就是用来书写计算机程序的语言。

但是，我们用程序语言输入的程序指令如果不经转换，也就是不被翻译为 CPU 可执行的机器语言指令，那么就不能被计算机识别并执行。那么，如何将高级语言程序翻译成计算机可识别的机器语言程序呢？换句话说，这个翻译处理又是如何发生的呢？

答案是由另一个程序来完成编程语言到机器语言的翻译，这种翻译程序是由专家级的程序员用机器语言编写的，也叫汇编程序或编译程序。当然，若想正确地翻译，要求高级语言必须制定有相应的语法规则，高级语言程序要按照像 Visual Basic 这样的编程语言语法规则严格来书写，否则，翻译程序将搞混以至于不能继续进行下去，翻译工作将停止并产生错误信息。当然，千万不要以为如果程序出错就会损坏计算机，只是严重的时候会导致计算机的“大脑”（即 CPU）思路错乱罢了。

用编程语言写程序的过程称为编码，这些程序叫做源代码。再次强调，编码必须遵守所使用编程语言的规则，否则很难翻译。如有 3 个人，一个是中国人，一个是法国人，而另一个是能将标准的普通话中文翻译成法文的翻译，假设那个中国人说的普通话不是很标准，那么这个翻译将会花很多时间去努力纠正直至最终能听懂并翻译给那个法国人，但假设那个中国人说的是翻译根本无法听懂的上海话，那么翻译也束手无策。

一旦程序翻译为机器语言，计算机就能执行该指令了。大家购买的或在网上下载的很多软件都是由编程语言编写的，这些程序一般都已经翻译为了机器语言，以文件形式存在，这些程序的扩展名一般为 .exe，这样的文件称为可执行文件。计算机可以脱离编程环境运行可执行文件，计算机开始执行程序指令系列时称为计算机正在运行程序。

真正编码前一般有个算法设计的过程，算法是解决问题的方法和步骤，一般可用流程图、自然语言或类高级语言来完成。而程序是用来实现算法的，是用某一种高级语言编写的。用流程图或自然语言表示的算法计算机是无法识别和执行的。必须将其转换为可以接受的计算机语言程序，计算机才能运行它并得出结果。

目前计算机程序设计语言种类繁多，而且是层出不穷，如 C，C++，Visual C，Visual Basic，Delphi，Java，Asp 和 SQL 等，许多当前使用的商业软件就是使用这些语言开发的。不同的计算机语言具有不同的特点、时代特征和应用领域，如商业和管理领域多选