

高等学校计算机程序设计课程系列教材

Visual Basic程序设计基础

申石磊 季 超 主编



高等教育出版社
Higher Education Press

高等学校计算机程序设计课程系列教材

Visual Basic 程序设计基础

Visual Basic Chengxu Sheji Jichu

申石磊 季 超 主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容简介

本书针对非计算机专业学生的特点,从 Visual Basic 程序设计基础开始讲解,内容安排由浅入深、循序渐进,面授教学和实验教学融为一体,写作风格力求文字准确、概念清晰、简明易懂。每章除包含大量典型例题外,章尾均附有实验练习和课外习题。

全书共分 12 章:第 1 章是 Visual Basic 概述,第 2 章是简单程序设计,第 3 章是数据运算基础,第 4 章是控制结构,第 5 章是数组应用,第 6 章是过程设计,第 7 章是控件设计,第 8 章是绘制图形,第 9 章是键盘鼠标事件,第 10 章是数据文件,第 11 章是复杂界面设计,第 12 章是课程设计指导。

本书根据教育部高等学校计算机基础课程教学指导委员会的教学基本要求(2008 版),参照全国计算机等级考试大纲编写,可以作为高等院校非计算机专业第一门程序设计课程的教材,还可以供全国计算机等级考试使用。

图书在版编目(CIP)数据

Visual Basic 程序设计基础 / 申石磊, 季超主编.

北京: 高等教育出版社, 2010.3

ISBN 978 - 7 - 04 - 018686 - 4

I. ①V… II. ①申… ②季… 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 003687 号

策划编辑	郑 涛	责任编辑	焦建虹	封面设计	于文燕
版式设计	张 岚	责任校对	刘 莉	责任印制	陈伟光

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 涿州市星河印刷有限公司

开 本 787 × 1092 1/16
印 张 15.75
字 数 380 000

购书热线 010-58581118
咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2010 年 3 月第 1 版
印 次 2010 年 3 月第 1 次印刷
定 价 17.40 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 18686-00

前言

Visual Basic 采用可视化编程技术、面向对象的事件驱动编程机制,可以快速开发数据库、多媒体和网络应用程序。Visual Basic 简单、易学,非常适合作为高等院校非计算机专业的程序设计课程。

本书根据教育部高等学校计算机基础课程教学指导委员会编制的《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》,参照全国计算机等级考试大纲,精选教学内容,可以作为理工、管理、经贸、农医、文史类大学生的 Visual Basic 程序设计教材。

全书共分 12 章:第 1 章是 Visual Basic 概述,第 2 章是简单程序设计,第 3 章是数据运算基础,第 4 章是控制结构,第 5 章是数组应用,第 6 章是过程设计,第 7 章是控件设计,第 8 章是绘制图形,第 9 章是键盘鼠标事件,第 10 章是数据文件,第 11 章是复杂界面设计,第 12 章是课程设计指导。本书编写特色如下:

(1) 精心设计篇章结构

围绕程序界面设计和程序代码设计两条主线,按照三个层次编排教学内容。第 1 章和第 2 章作为第一个层次,介绍编程环境和简单界面设计,为后续内容打下基础。第 3~6 章构成第二个层次,重点讲解如何编写程序代码。第 7~11 章为第三个层次,讲解程序界面设计和程序代码的灵活运用。第 12 章是全书内容的扩展,旨在提高学生的综合应用能力,实验内容可由教师自行设计。

(2) 融会实验教学环节

本书将面授教学和实验教学紧密结合在一起,有些章尾附有实验目的、实验指导和实验内容。程序调试分散在各章的实验指导中,面授教学可以结合例题介绍程序调试方法,使学生逐步掌握程序调试技巧。课外作业安排在实验教学环节之后,每章教学过程按照“面授教学—实验练习—课外作业”的顺序设计。考虑到非计算机专业学生的特点,实验内容由易到难、循序渐进。

(3) 精选全书教学内容

本书作为 Visual Basic 程序设计基础,未涉及数据库编程、网络编程和多媒体编程。全书的基本概念、例题、实验内容和课外作业经过反复推敲,力求文字简练、概念准确。课外作业的选择习题设计为多项选择,包含了书中讲解的大部分概念。各章都选取了具有典型性、示范性和启发性的例题,保证在有限学时内达到预期的教学目标。

本书由申石磊和季超任主编,王红涛、楚艳萍和侯松鹂任副主编,申石磊和季超负责制订编写大纲,并提出编写要求,王红涛、楚艳萍和侯松鹂负责实验设计和全书校对工作。申石磊编写第 1 章、第 8 章、第 9 章和第 12 章,季超编写第 2 章和第 3

II 前言

章,楚艳萍编写第4章和第7章,王红涛编写第5章和第6章,侯松鹂编写第10章和第11章。

本书编写人员都是在高等学校从事计算机教学的教师,具有一定的教学经验,但由于时间仓促,书中错误在所难免,恳请使用本书的教师和学生给予批评指正。

编 者

2009年9月

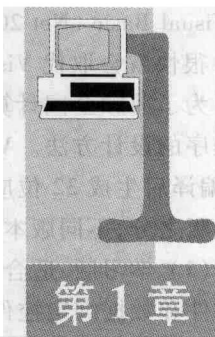
目 录

第 1 章 Visual Basic 概述	1	2.2 窗体设计	20
1.1 Visual Basic 系统简介	1	2.2.1 窗体属性	21
1.1.1 Visual Basic 演变过程	1	2.2.2 窗体方法	24
1.1.2 Visual Basic 功能特点	2	2.2.3 窗体事件	26
1.2 Visual Basic 工作环境	3	2.3 简单控件	27
1.2.1 设计窗口	3	2.3.1 标签	27
1.2.2 窗体窗口	5	2.3.2 命令按钮	28
1.2.3 工具窗口	6	2.3.3 计时器	29
1.2.4 属性窗口	6	2.4 文本框	30
1.2.5 代码窗口	7	2.4.1 文本框的属性	31
1.2.6 工程窗口	7	2.4.2 文本框的事件	33
1.3 控件操作	8	2.4.3 文本框的方法	34
1.3.1 绘制控件	8	2.5 工程管理	35
1.3.2 编辑控件	8	2.5.1 工程结构	35
1.3.3 控件布局	9	2.5.2 工程组	36
1.4 使用帮助系统	9	2.5.3 保存工程	36
1.4.1 使用帮助窗口	9	2.5.4 打开工程	36
1.4.2 获取相关帮助	10	2.5.5 运行工程	37
1.4.3 使用在线帮助	11	2.5.6 设置启动方式	37
1.4.4 运行示例程序	11	实验二	38
1.5 工作环境设置	11	习题二	41
1.5.1 “编辑器格式”选项卡	11	第 3 章 数据运算基础	44
1.5.2 “编辑器”选项卡	12	3.1 标准数据类型	44
实验一	13	3.1.1 数值数据	44
习题一	14	3.1.2 字符数据	45
第 2 章 简单程序设计	16	3.1.3 逻辑数据	46
2.1 面向对象的概念	16	3.1.4 日期数据	46
2.1.1 对象	16	3.1.5 变体类型	46
2.1.2 属性	16	3.2 常量和变量	46
2.1.3 方法	17	3.2.1 常量	46
2.1.4 事件	17	3.2.2 变量	48
2.1.5 程序设计举例	18	3.3 运算符和表达式	50

3.3.1 数值表达式	50	5.1.3 变体数组	90
3.3.2 字符表达式	51	5.1.4 For Each...Next 语句	91
3.3.3 日期表达式	51	5.2 动态数组	92
3.3.4 关系表达式	52	5.2.1 定义动态数组	92
3.3.5 逻辑表达式	52	5.2.2 删除动态数组	93
3.4 常用内部函数	53	5.3 控件数组	95
3.4.1 数值运算函数	54	5.3.1 设计阶段创建控件数组	95
3.4.2 字符处理函数	55	5.3.2 运行阶段创建控件数组	97
3.4.3 日期时间函数	56	5.4 记录数组	97
3.4.4 类型转换函数	57	5.4.1 记录类型	98
3.4.5 格式转换函数	58	5.4.2 记录数组	98
3.4.6 颜色设置函数	58	实验五	100
3.4.7 程序调用函数	59	习题五	103
实验三	60	第6章 过程设计	106
习题三	62	6.1 Sub 过程	106
第4章 控制结构	65	6.1.1 Sub 过程的定义	106
4.1 顺序结构	65	6.1.2 Sub 过程的调用	107
4.1.1 简单语句	65	6.1.3 Sub Main 过程	109
4.1.2 数据输入	66	6.2 Function 过程	109
4.1.3 数据输出	67	6.2.1 Function 过程的定义	109
4.2 选择结构	69	6.2.2 Function 过程的调用	110
4.2.1 双分支语句	69	6.3 过程的参数传递	111
4.2.2 多分支语句 (If 结构)	71	6.3.1 参数传递方式	112
4.2.3 多分支语句 (Select Case 结构) ...	73	6.3.2 数组参数	114
4.2.4 条件函数	74	6.3.3 对象参数	115
4.3 循环结构	75	6.3.4 可选参数	115
4.3.1 For...Next 循环语句	75	6.3.5 可变参数	116
4.3.2 Do...Loop 循环语句	77	6.4 过程和变量的作用域	117
4.3.3 循环嵌套	79	6.4.1 过程的作用域	117
4.3.4 退出循环	80	6.4.2 变量的作用域	117
4.3.5 GoTo 语句	81	6.4.3 变量的生存期	118
4.3.6 DoEvents 语句	81	6.5 多重过程调用	120
实验四	82	6.5.1 嵌套调用	120
习题四	84	6.5.2 递归调用	122
第5章 数组应用	87	实验六	124
5.1 固定数组	87	习题六	128
5.1.1 定义数组	87	第7章 控件设计	134
5.1.2 使用数组	88	7.1 图形控件	134

7.1.1 直线和形状	134	9.2.1 指针形状	175
7.1.2 图像框	136	9.2.2 鼠标事件	176
7.1.3 图文框	137	9.3 拖放操作	178
7.2 量值控件	139	9.3.1 拖放属性	179
7.2.1 滚动条	139	9.3.2 拖放方法	179
7.2.2 进度条	141	9.3.3 拖放事件	179
7.3 选值控件	143	实验九	182
7.3.1 单选按钮和复选框	143	习题九	183
7.3.2 框架	144	第10章 数据文件	186
7.3.3 列表框	145	10.1 数据文件概述	186
7.3.4 组合框	149	10.1.1 数据文件结构	186
7.4 文件系统控件	150	10.1.2 数据文件类型	187
7.4.1 驱动器列表框	150	10.1.3 数据文件操作	188
7.4.2 目录列表框	151	10.2 文件操作函数	188
7.4.3 文件列表框	151	10.2.1 文件指针	188
实验七	154	10.2.2 常用函数	189
习题七	155	10.3 顺序文件操作	189
第8章 绘制图形	158	10.3.1 打开顺序文件	190
8.1 绘图坐标	158	10.3.2 向文件中写入数据	190
8.1.1 坐标度量单位	158	10.3.3 从文件中读取数据	191
8.1.2 定义容器坐标	159	10.4 随机文件操作	193
8.2 绘图属性	160	10.4.1 打开随机文件	194
8.2.1 当前坐标	160	10.4.2 读写随机文件	194
8.2.2 图形样式	160	10.5 二进制文件操作	196
8.3 绘图方法	162	10.5.1 操作语句	196
8.3.1 PSet 方法	162	10.5.2 应用举例	197
8.3.2 Line 方法	163	实验十	198
8.3.3 Circle 方法	165	习题十	200
8.4 图像操作	166	第11章 复杂界面设计	203
8.4.1 读取像素	166	11.1 通用对话框	203
8.4.2 变换图像	167	11.1.1 通用对话框控件	203
实验八	168	11.1.2 文件对话框	204
习题八	169	11.1.3 颜色对话框	207
第9章 键盘鼠标事件	171	11.1.4 字体对话框	208
9.1 键盘事件	171	11.2 菜单设计	209
9.1.1 KeyPress 事件	171	11.2.1 菜单结构	210
9.1.2 KeyDown 和 KeyUp 事件	172	11.2.2 菜单属性	210
9.2 鼠标事件	175	11.2.3 下拉菜单设计	211

11.2.4 弹出菜单设计·····	215	12.2 系统内部对象·····	233
11.3 工具栏设计·····	219	12.2.1 App 对象·····	233
11.3.1 ImageList 控件·····	219	12.2.2 Screen 对象·····	234
11.3.2 ToolBar 控件·····	220	12.2.3 ClipBoard 对象·····	235
11.3.3 编写程序代码·····	222	12.3 应用程序向导·····	236
实验十一·····	223	12.3.1 启动应用程序向导·····	236
习题十一·····	227	12.3.2 创建应用程序界面·····	237
第 12 章 课程设计指导·····	230	12.4 发布应用程序·····	239
12.1 错误处理程序·····	230	12.4.1 启动打包和展开向导·····	239
12.1.1 运行错误处理·····	230	12.4.2 创建安装程序·····	240
12.1.2 出错信息分析·····	231	12.4.3 展开安装程序·····	241



Visual Basic 采用面向对象的可视化编程技术和事件驱动编程机制,其功能强大、使用方便,非常适合作为非计算机专业的第一门程序设计课程。在 Visual Basic 集成开发环境下,可以开发各种应用程序,如网络应用程序、多媒体应用程序、数据库应用程序等。

本章是全书的基础,在学习本章内容时,务必完成本章实验内容,熟练掌握 Visual Basic 的集成开发环境,为后续内容的学习打下基础。

1.1 Visual Basic 系统简介

Visual Basic 是目前广泛使用的程序设计语言之一,在开始学习程序设计之前,先简要了解 Visual Basic 程序设计语言的演变过程和功能特点。

1.1.1 Visual Basic 演变过程

Visual Basic 从 BASIC 程序设计语言演变而来,沿用了 BASIC 语言的一些语法,继承了 BASIC 语言简单、易学的特点。BASIC 语言诞生于 1964 年,适用于字符界面的 DOS 操作系统,曾经对计算机的推广和应用发挥了重要作用。随着 Windows 图形界面操作系统的出现,可视化编程工具应运而生,其中 Visual Basic 就是一个杰出的代表。

Visual Basic 各个版本的基本特点如下:

1991 年,Visual Basic 1.0,16 位应用程序开发工具,只有英文版。

1992 年,Visual Basic 2.0,16 位应用程序开发工具,只有英文版。

1993 年,Visual Basic 3.0,16 位应用程序开发工具,只有英文版。

1995 年,Visual Basic 4.0,16 位应用程序开发工具,只有英文版。

1997 年,Visual Basic 5.0,32 位应用程序开发工具,既有英文版又有中文版。

1998 年,Visual Basic 6.0,32 位应用程序开发工具,既有英文版又有中文版。

2001 年,Visual Basic .Net,基于 Visual Basic 6.0,一种真正面向对象的程序设计语言,功能更加强大,界面设计更加简单,程序代码更加稳定。目前,两个非常流行的版本

为 Visual Basic .Net 2005 和 Visual Basic .Net 2008。随着计算机应用的快速发展, Visual Basic .Net 很快就会取代 Visual Basic 6.0。

为了兼顾全国计算机等级考试,本书将以 Visual Basic 6.0 简体中文专业版为背景,介绍应用程序的设计方法。Visual Basic 6.0 可在 Windows 98、Windows 2000 和 Windows XP 等环境下运行,编译后生成 32 位应用程序,运行速度快,且安全可靠。Visual Basic 6.0 有学习版、专业版和企业版之分,不同版本适用于不同层次的用户,其主要特点如下:

(1) 学习版:适合初学者开发应用程序,包括 Visual Basic 6.0 的基本功能,如标准控件、网格控件和数据绑定控件等,同时还提供了联机帮助文档。

(2) 专业版:适合专业编程人员开发应用程序,在学习版的基础上,又增加了 ActiveX 控件、Internet 控件和报表控件。

(3) 企业版:在专业版的基础上,又增加了套装 BackOffice 工具,适合专业编程人员开发功能强大的分布式应用程序。

1.1.2 Visual Basic 功能特点

Visual Basic 6.0 具有许多功能特点,如网络编程、Internet 组件下载、创建 ActiveX 控件、数据库访问、动态数据交换(DDE)、对象的链接与嵌入(OLE)、动态链接库(DLL)等,其主要功能特点如下。

1. 可视化程序设计

Visual Basic 的第一个特点就是可视化,这里的 Visual 是指图形程序界面的设计方法。在 Visual Basic 环境下,程序设计人员不需要编写程序代码,仅将工具箱中的控件添加到窗体,适当设置属性,即可创建应用程序界面。

2. 面向对象的程序设计

面向对象的程序设计方法把现实世界中的客观实体看做是不同类型的对象,对象由属性和方法组成。属性是一组数据,用来描述对象所具有的特征;方法是一段程序代码,用来描述对象所具有的行为。

Visual Basic 6.0 支持面向对象的程序设计方法,采用可视化编程技术,所有对象以图形方式显示,只要把对象添加在窗体上,就会自动生成描述对象的程序代码。实际上,Visual Basic 只是应用了面向对象的程序设计方法,因此使用 Visual Basic 编写程序比较简单。

3. 事件驱动编程机制

事件是对象能够识别的动作,不同的对象识别不同的事件,一个对象可以识别多个事件。每个事件都可以对应一段程序代码,当事件发生时,对象识别并响应事件,执行事件对应的程序代码,完成指定操作。这样的编程机制称为事件驱动编程机制。

Visual Basic 采用事件驱动编程机制,编程人员只需编写响应事件的程序代码。各个事件之间没有联系,每段程序代码功能单一、相对独立,相当于把一个大型应用程序分解为若干小程序段,既降低了编程难度,又便于程序维护。

事件驱动编程机制不是按照事先预定的顺序执行程序代码,而是响应不同的事件,执行不同的程序代码,事件发生的先后顺序决定了程序代码的执行顺序。每次运行程序时,事件发生的先

后顺序不同,程序代码的执行顺序也就不同。

4. 程序代码结构化

Visual Basic 继承了 BASIC 语言的结构化程序设计特点,具有高级程序设计语言的语句结构,程序代码由顺序结构、选择结构和循环结构组成。语句简单、易懂,接近人类自然语言,适合初学者学习程序设计。

1.2 Visual Basic 工作环境

Visual Basic 集成开发环境包括:设计窗口、窗体窗口、工具窗口、属性窗口、工程窗口、代码窗口、立即窗口、窗体布局窗口、本地窗口、监视窗口。根据工作环境需要,用户可以移动、缩小、放大或关闭这些窗口。本节仅介绍几个常用窗口,其他窗口将在后续的实验指导中介绍。

1.2.1 设计窗口

1. 启动系统

在 Windows XP 环境下,双击桌面上的图标,或单击“开始”按钮,从“所有程序”菜单中选择“Microsoft Visual Basic 6.0 中文版”命令,开始启动 Visual Basic 6.0 系统。在启动过程中,可能出现如图 1-1 所示的“新建工程”对话框(选中“不再显示这个对话框”复选框,以后再启动 Visual Basic 系统,就不再显示“新建工程”对话框)。“新建工程”对话框中的选项卡的作用如下:

- “新建”:列出可以建立的所有类型的工程,如“标准 EXE”、“ActiveX 控件”等。
- “现存”:查找并打开已经建立的某个工程文件。
- “最新”:选择并打开最近使用过的某个工程文件。



图 1-1 “新建工程”对话框

本书只介绍“标准 EXE”工程类型。在“新建”选项卡中,双击“标准 EXE”图标,或先选中“标准 EXE”图标,再单击“打开”按钮,出现如图 1-2 所示的集成开发环境。

2. 关闭系统

工作完毕,可以关闭 Visual Basic 系统,有以下几种方法:

- (1) 单击图 1-2 右上角的关闭按钮 .
- (2) 选择“文件”→“退出”命令。
- (3) 按组合键 Alt + Q。

在关闭 Visual Basic 时,系统自动保存当前 Visual Basic 工作环境的设置,并作为下次启动时的默认设置,如窗口大小、窗口位置、工具栏状态、代码文本字体等。因此,每次启动 Visual Basic 系统后,工作环境可能不同。

3. 窗口组成

设计窗口又称主窗口,如图 1-2 所示,设计窗口由标题栏、菜单栏和工具栏组成。

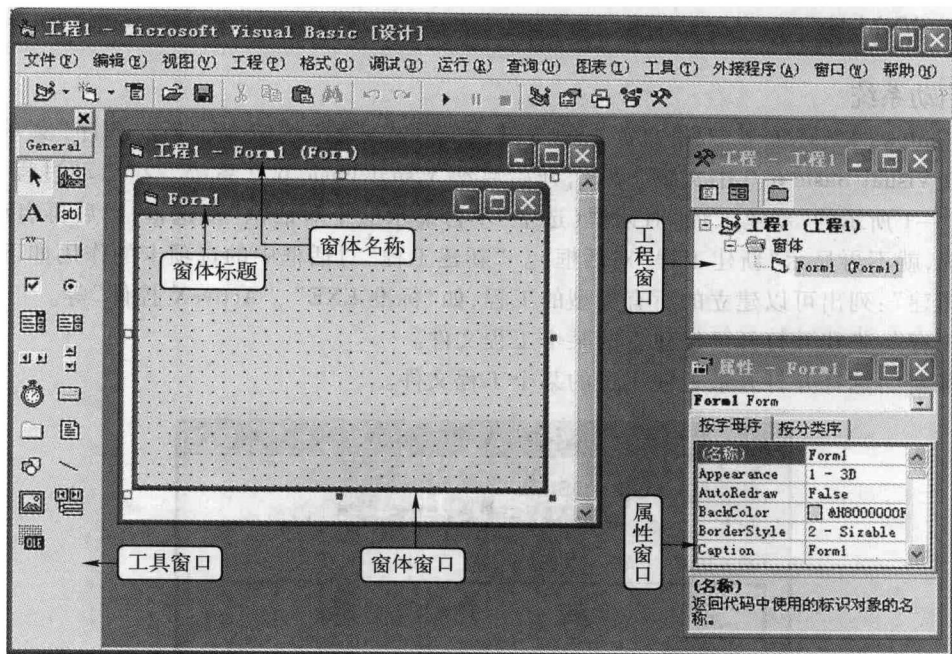


图 1-2 设计窗口(主窗口)

(1) 标题栏:标题栏中显示“工程名称 - 程序名称[工作模式]”。图 1-2 中的“工程 1”为工程名称,“Microsoft Visual Basic”为程序名称,“设计”为工作模式。Visual Basic 6.0 的 3 种工作模式包括:设计、运行和中断(Break),工作模式随工作状态的改变而不同。

(2) 菜单栏:菜单栏中包含程序设计使用的全部命令。菜单栏共有 13 个菜单项,每个菜单项包含一组相关的操作命令。例如,“文件”菜单主要包含工程文件的操作命令,“编辑”菜单用来编辑程序代码,“视图”菜单用来打开各种窗口,使用“运行”菜单中的命令可以运行程序。

(3) 工具栏:工具栏中的按钮对应菜单中的常用命令,用户不需要打开菜单,只要用鼠标单

击工具栏中的图标,就能执行一些常用菜单命令,从而提高操作效率。Visual Basic 6.0 提供了“标准”、“编辑”、“调试”和“窗体编辑器”4 种工具栏,通常只显示“标准”工具栏。从“视图”→“工具栏”菜单中选择命令,或右击“标准”工具栏,从快捷菜单中选择命令,均可打开或关闭工具栏。

每种工具栏都有固定和浮动两种形式。图 1-2 中的“标准”工具栏为固定形式,图 1-3 中的“标准”工具栏为浮动形式。在固定工具栏中,将鼠标指针移到左端的双竖线,按住鼠标左键向下拖动或双击鼠标左键,固定工具栏就变为浮动工具栏。双击浮动工具栏的标题栏,或向上拖动浮动工具栏,浮动工具栏又变为固定工具栏。

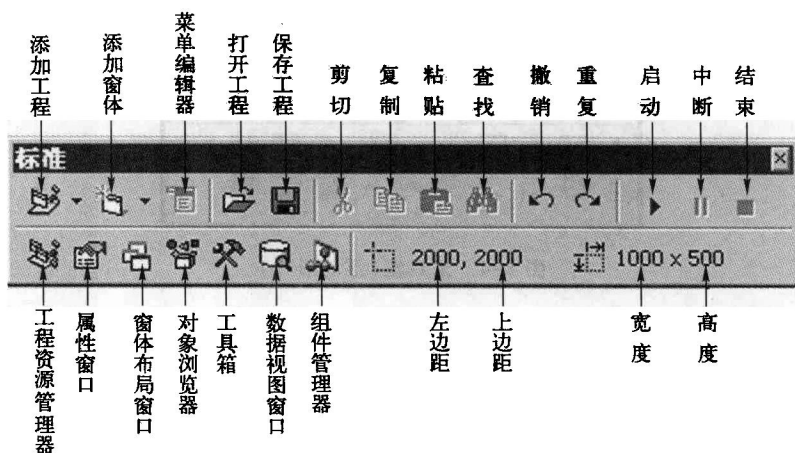


图 1-3 “标准”工具栏(浮动形式)

1.2.2 窗体窗口

窗体窗口的全称为窗体设计器窗口,用来设计应用程序界面,内部包含窗体(Form)。通常情况下,一个应用程序至少有一个窗体,最多可有 255 个窗体,每个窗体最多放置 254 个控件。

在设计应用程序时,窗体如同一块画布,可在窗体上添加各种各样的控件,创建应用程序界面。当程序运行时,一个窗体对应于一个窗口或一个对话框,可在窗体上绘制图形、显示文本或输出数据。

如图 1-2 所示,窗体窗口的标题栏显示“工程名称 - 窗体名称(Form)”,Form 表示窗体。启动 Visual Basic 系统或新建一个工程后,默认工程名称为“工程 N”,默认窗体名称和窗体标题为“FormN”(N=1,2,3,...)。在设计应用程序时,用户还可以改变工程名称、窗体名称和窗体标题。

当窗体窗口关闭时,执行“视图”→“对象窗口”命令,可以再次显示窗体窗口。拖动窗体窗口的边框或四角,可以改变窗体窗口的大小。窗体周围有 8 个小方块(图 1-2),拖动 3 个实心蓝色小方块,可以改变窗体的大小。默认设计状态下,窗体上布满了网格小点,这些网格用来对齐窗体上的控件,运行程序时网格小点自然消失。

1.2.3 工具窗口

工具窗口(又称工具箱窗口)包括 21 个工具图标,其中 20 个图标(不包括图标 )用来在窗体上绘制控件。将鼠标指针指向工具箱中的图标,显示控件的英文名称。单击工具箱中的图标,按住鼠标左键在窗体上拖动,可在窗体上绘制控件。

在设计状态,工具箱总是显示;在运行状态,工具箱自动隐藏。打开工具窗口的操作有:①选择“视图”→“工具箱”命令;②单击“标准”工具栏中的“工具箱”按钮 。

默认情况下,工具箱位于设计窗口左侧。拖动标题栏或双击标题栏,可使工具箱处于浮动状态,如图 1-4 所示;再次双击标题栏,又恢复到原来状态。

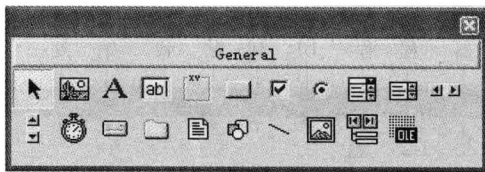
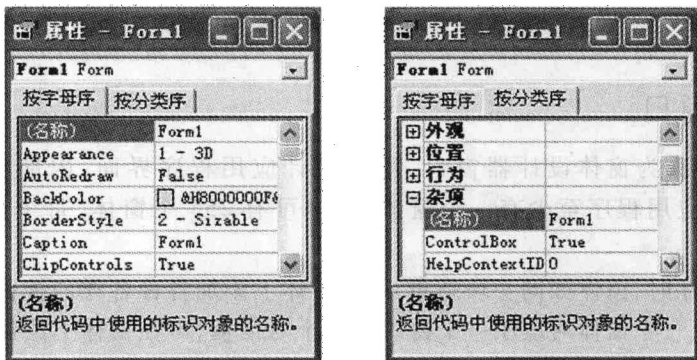


图 1-4 工具箱(浮动形式)

1.2.4 属性窗口

属性窗口如图 1-5 所示,用来设置窗体对象的属性,各个组成部分如下:



(a) “按字母序”选项卡

(b) “按分类序”选项卡

图 1-5 属性窗口

(1) 标题栏:位于属性窗口顶端,显示当前所选择的对象名称。

(2) 对象列表:位于标题栏下面,用下拉列表框显示当前窗体中的所有对象,单击下拉列表框,显示对象名称和对对象类型,可从中选择一个对象。

(3) 属性列表:位于对象列表下面,用列表框显示当前对象在设计阶段可以使用的属性名称和属性取值,左列为属性名称,右列为属性取值。

(4) 显示方式:选择“按字母序”选项卡,按字母顺序显示属性名称;选择“按分类序”选项

卡,按类别显示属性名称。

(5) 属性说明:在属性列表选择一个属性,显示简短信息,说明属性作用。

如果属性窗口是关闭的,打开属性窗口的操作有:

- ① 选择“视图”→“属性窗口”命令。
- ② 单击“标准”工具栏中的“属性窗口”按钮。
- ③ 按 F4 键。
- ④ 右击窗体,选择“属性窗口”命令。

1.2.5 代码窗口

代码窗口用来编写程序代码。如果一个工程有多个窗体,先单击窗体窗口或双击工程窗口中的窗体名称,选择一个窗体,再打开如图 1-6 所示的代码窗口。打开代码窗口的方法如下:

- (1) 双击窗体或双击窗体上的某个控件。
- (2) 执行“视图”→“代码窗口”命令。
- (3) 按 F7 键。
- (4) 单击工程窗口中的“查看代码”按钮。
- (5) 右击窗体或控件,选择“查看代码”命令。

如图 1-6 所示,代码窗口由标题栏、对象下拉列表框、事件下拉列表框(又称过程框)和代码编辑区组成。标题栏中显示“工程名称 - 窗体名称 (Code)”,Code 表示代码窗口。对象下拉列表框包括当前窗体中的所有对象,事件下拉列表框包括所选择对象的全部事件。对象下拉列表框中的“通用”选项用来声明窗体变量、全局变量或编写通用过程。

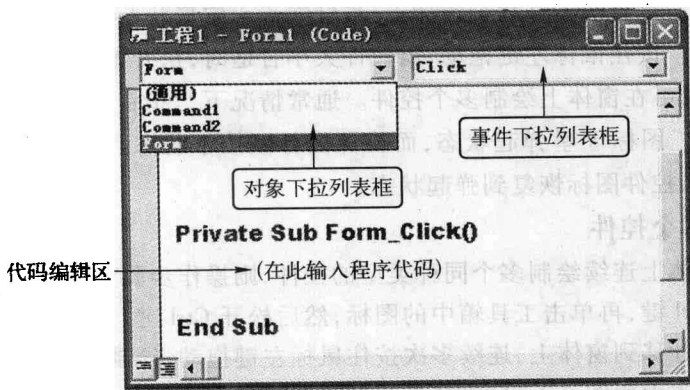


图 1-6 代码窗口

在编写程序代码时,先单击对象下拉列表框,选择一个对象,再单击事件下拉列表框,选择一个事件,代码编辑区出现一个事件过程模板,在模板内部输入程序代码。

1.2.6 工程窗口

工程窗口的全称为工程资源管理器窗口,用来管理应用程序中的所有文件,如工程文件、窗体文件、标准模块文件等。在工程窗口中,以层次结构显示工程文件中的各类文件,括号外是工

程名称和窗体名称,括号内是工程文件名称和窗体文件名称。单击加号“+”按钮,展开下层文件,加号变为减号;单击减号“-”按钮,折叠下层文件,减号变为加号。

如果工程窗口是关闭的,打开工程窗口的操作有:①选择“视图”→“工程资源管理器”命令;②单击“标准”工具栏中的“工程资源管理器”按钮;③按组合键 Ctrl + R。

工程窗口标题栏下面有3个按钮,其作用如下:

- (1) 查看代码:单击此按钮,打开代码窗口,查看或编写程序代码。
- (2) 查看对象:窗体窗口关闭时,选择窗体名称,单击此按钮,显示窗体窗口。双击工程窗口中的窗体名称,同样可以显示窗体窗口。
- (3) 切换文件夹:单击此按钮,显示或不显示文件层次结构中的文件夹。

1.3 控件操作

控件用来构成应用程序界面,控件操作包括绘制控件、编辑控件和调整控件布局。

1.3.1 绘制控件

在设计程序界面时,需要在窗体上绘制各种各样的控件,绘制控件的方法如下。

1. 一次绘制一个控件

如果只在窗体上绘制一个某种类型的控件,则操作步骤如下:

- (1) 选择图标。用鼠标单击工具箱中的图标,选择要绘制的控件。
- (2) 确定位置。将鼠标指针移到窗体上,指针变为十字形状“+”。
- (3) 绘制控件。按住鼠标左键拖动,当控件大小合适时,松开鼠标左键。

重复上述操作,可在窗体上绘制多个控件。通常情况下,“指针”图标呈按下状态;单击某个控件图标,“指针”图标呈弹起状态,而控件图标呈按下状态。控件绘制完毕,“指针”图标恢复到按下状态,控件图标恢复到弹起状态。

2. 连续绘制多个控件

如果需要在窗体上连续绘制多个同种类型的控件,则操作步骤如下:

- (1) 先按住 Ctrl 键,再单击工具箱中的图标,然后松开 Ctrl 键。
- (2) 将鼠标指针移到窗体上,连续多次按住鼠标左键拖动,绘制多个同类控件。
- (3) 单击工具箱中的图标,结束绘制控件操作;单击其他图标,绘制其他控件。

3. 在窗体中心绘制控件

双击工具箱中的某个控件图标,在窗体中心绘制一个默认大小的控件。

1.3.2 编辑控件

1. 选择控件

在操作控件前,必须先选择控件。在窗体上选择控件的方法如下:

- (1) 单击窗体上的某个控件,选择一个控件。