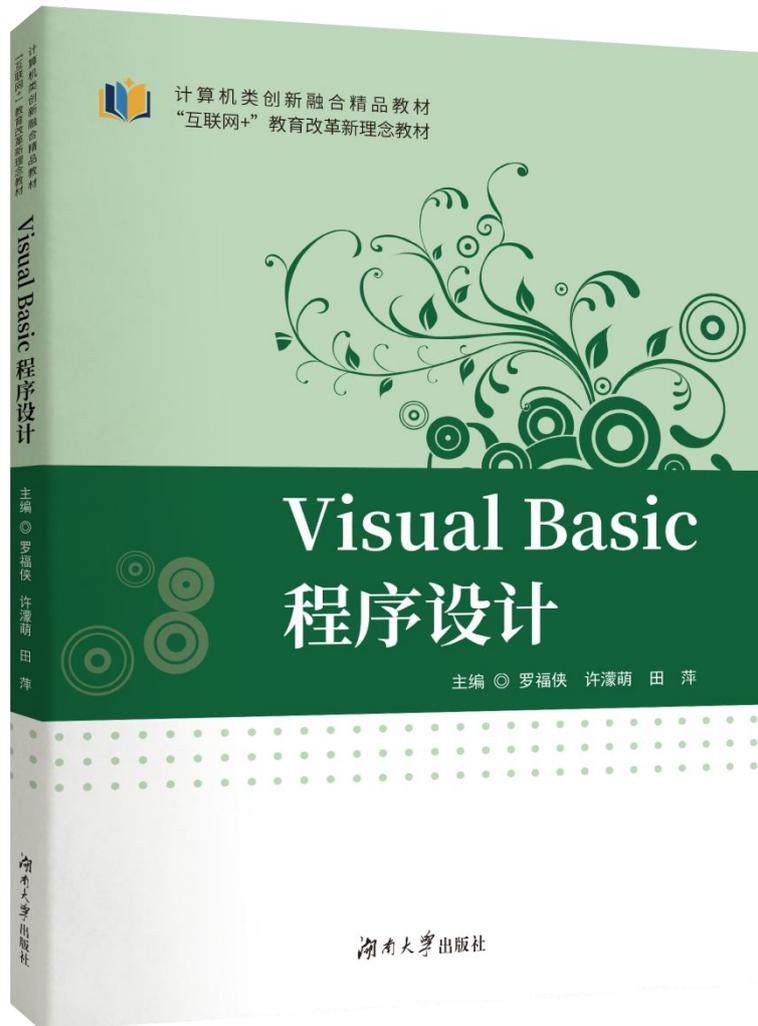


Visual Basic 程序设计



类目：计算机类

书名：Visual Basic 程序设计

主编：罗福侠 徐濛萌 田萍

出版社：湖南大学出版社

开本：大 16 开

书号：978-7-5667-4051-9

使用层次：通用

出版时间：2025 年 7 月

定价：49.80 元

印刷方式：双色

是否有资源：有



计算机类创新融合精品教材
“互联网+”教育改革新理念教材

计算机类创新融合精品教材
“互联网+”教育改革新理念教材

Visual Basic 程序设计



Visual Basic 程序设计

Visual Basic 程序设计

主编 © 罗福侠 许濛萌 田 萍

主编 © 罗福侠 许濛萌 田 萍

责任编辑: 周紫燕
封面设计: 唐语书装



湖南大学出版社

湖南大学出版社



计算机类创新融合精品教材
“互联网+”教育改革创新理念教材



Visual Basic 程序设计

主 编 ◎ 罗福侠 许濛萌 田 萍

主 审 ◎ 张一平

副主编 ◎ 王 贺 武文凯 王 倩

张 帅 宋 乔

湖南大学出版社

· 长 沙 ·

图书在版编目 (CIP) 数据

Visual Basic 程序设计 / 罗福侠, 许濛萌, 田萍主
编. -- 长沙: 湖南大学出版社, 2025.7. -- ISBN 978-
7-5667-4051-9
I . TP312.8
中国国家版本馆 CIP 数据核字第 2025A32G80 号

Visual Basic 程序设计

Visual Basic CHENGXU SHEJI

主 编: 罗福侠 许濛萌 田 萍

责任编辑: 周紫燕

印 装: 唐山唐文印刷有限公司

开 本: 889 mm×1194 mm 1/16 印 张: 13.5 字 数: 343 千字

版 次: 2025 年 7 月第 1 版 印 次: 2025 年 7 月第 1 次印刷

书 号: ISBN 978-7-5667-4051-9

定 价: 49.80 元

出 版 人: 李文邦

出版发行: 湖南大学出版社

社 址: 湖南·长沙·岳麓山 邮 编: 410082

电 话: 0731-88822559 (营销部) 88821174 (编辑室) 88821006 (出版部)

传 真: 0731-88822264 (总编室)

网 址: <http://press.hnu.edu.cn>

电子邮箱: xiaoshulianwenhua@163.com

版权所有, 盗版必究

图书凡有印装差错, 请与营销部联系



PREFACE



Windows 是当今主流操作系统，而 Visual Basic 作为一种支持可视化程序设计的语言，也是开发 Windows 应用程序最简单易学的语言之一。为了满足社会对 Visual Basic 程序设计人才的需求，以及适应学校相关专业教学的需要，我们编写了这本《Visual Basic 程序设计》教材。

本教材共分为 9 章，内容主要包括 Visual Basic 概述，对象、窗体与控件，Visual Basic 语言基础，基本程序控制结构，数组，过程，用户界面设计，文件，数据库程序设计。与此同时，本教材设计了“学习目标”“章节概述”“习题巩固”“操作演练”等模块，以此丰富教学内容。

本教材具有以下几个显著的特点：

首先，内容全面且系统。本教材从 Visual Basic 程序设计的基础知识讲起，包括定义、分类等，逐步深入到 Visual Basic 程序设计的具体操作知识。这种全面而系统的内容安排，使得读者可以快速建立起对 Visual Basic 程序设计的整体认识，并深入了解其应用细节。

其次，强调实践与应用。本教材不仅仅是一本理论书籍，还包含了大量的实践和应用内容，以帮助读者理解书中知识点，提高动手解决问题的能力。

再次，图文并茂，易于理解。为了让读者更好地理解 Visual Basic 程序设计，本教材运用图文结合、图表结合的方式，不仅增强了教材的可读性，也方便了读者对知识的理解和记忆。

本教材内容全面，知识由浅入深、注重实践，各章节均安排了适量的习题，适合作为高校计算机专业学生的课程教材，也可作为全国计算机等级考试以及各类程序设计培训班学员和 Visual Basic 语言自学者的参考用书。

编者

2025 年 3 月





CONTENTS



第1章 Visual Basic概述	1
1.1 初识 Visual Basic	2
1.2 Visual Basic 的特点	3
1.3 安装 Visual Basic 6.0	4
1.4 Visual Basic 的启动与退出	5
1.5 Visual Basic 6.0 集成开发环境	6
1.6 程序设计方法的发展历程	10
1.7 创建应用程序的过程	11
第2章 对象、窗体与控件	18
2.1 对象	19
2.2 窗体	22
2.3 控件	27
第3章 Visual Basic语言基础	36
3.1 数据类型	37
3.2 常量与变量	38
3.3 运算符与表达式	42
3.4 常用内部函数	46
第4章 基本程序控制结构	56
4.1 算法	57
4.2 顺序结构	59
4.3 选择结构	61
4.4 循环结构	67

第5章 数组	82
5.1 数组的概念	83
5.2 静态数组声明	84
5.3 动态数组	86
5.4 一维数组	87
5.5 二维数组	94
第6章 过程	101
6.1 过程的概念	102
6.2 Sub 过程	102
6.3 Function (函数) 过程	105
6.4 参数传递	109
6.5 变量的作用域与生存期	114
6.6 过程的嵌套与递归调用	115
6.7 键盘与鼠标事件	118
第7章 用户界面设计	124
7.1 菜单设计	125
7.2 工具栏设计	133
7.3 状态栏设计	140
7.4 对话框设计	143
7.5 图形设计	152
7.6 多重窗体程序设计	161
第8章 文件	169
8.1 文件概述	170
8.2 顺序文件	171
8.3 随机文件	178
8.4 二进制文件	181
8.5 文件系统控件	182
第9章 数据库程序设计	190
9.1 数据库基础知识	191
9.2 数据库访问	192
9.3 记录集对象	203
附录	207
参考文献	209

第1章

Visual Basic 概述

学习目标

知识目标

- ▶了解 Visual Basic 的基本概念及其发展历程。
- ▶掌握 Visual Basic 6.0 的安装方法和集成开发环境的主要组成部分。

能力目标

- ▶能够独立启动和退出 Visual Basic 6.0 应用程序。
- ▶熟练使用 Visual Basic 6.0 的集成开发环境进行基本的编程操作。

素质目标

- ▶培养对 Visual Basic 编程的兴趣和热情，形成持续学习的习惯。
- ▶提升逻辑思维能力和问题解决能力，能够在编程实践中不断探索和创新。

章节概述

Visual Basic (VB) 作为微软公司开发的一种事件驱动的编程语言，旨在简化 Windows 应用程序的开发过程。本章对 Visual Basic 编程语言进行概要介绍，并帮助读者熟悉相关开发环境。首先，简要介绍 Visual Basic 的基本概念及其特性，强调其在快速应用程序开发中的重要性。接着，介绍如

何安装 Visual Basic 6.0 以及如何启动和退出该软件。随后，回顾程序设计方法的发展历程，从早期的程序设计逐步引入结构化程序设计和面向对象程序设计的思想。最后，介绍创建应用程序的基本过程。

1.1 初识 Visual Basic

Visual Basic 是微软公司开发的一种通用的基于对象的程序设计语言，其集结构化、模块化、面向对象的设计理念于一身，并采用包含辅助开发环境的事件驱动机制，实现了可视化的程序设计。作为当今世界上使用较为广泛的编程语言之一，VB 被广泛认为是编程效率极高的工具。无论是开发功能强大、性能可靠的商务软件，还是编写用于解决实际问题的实用小程序，VB 都以其快速、简便的特点成为首选的开发工具。

Visual 是指采用可视化的开发图形用户界面（graphical user interface, GUI）的方法，一般不需要编写大量代码去描述界面元素的外观和位置，而只要把需要的控件拖放到屏幕上的相应位置即可；Basic 是指 Basic 语言，因为 VB 是在原有的 Basic 语言的基础上发展起来的，至今包含了数百条语句、函数及关键词，其中很多和 Windows GUI 有直接关系。对于专业人员而言，他们可以利用 Visual Basic 实现其他任何 Windows 编程语言所能实现的功能。而对于初学者来说，只需要掌握几个关键词，就能轻松地创建出实用的应用程序。

VB 提供了三种版本——学习版、专业版和企业版，以满足不同层次的开发需求。学习版让编程初学者也能轻松上手，开发 Windows 应用程序；专业版则为经验丰富的编程人员提供了全面且功能强大的开发工具；企业版则更进一步，支持专业开发团队以协作的方式，共同打造出强健的分布式应用程序。

1991 年，微软公司推出了 Visual Basic 1.0，在当时引起了很大的轰动。这个连接编程语言和用户界面的进步被称为 Tripod（有时叫作 Ruby）。许多专家把 VB 的出现当作软件开发史上的一个具有划时代意义的事件。在当时，它是第一个“可视”的编程软件。这使得程序员欣喜万分，都尝试在 VB 的平台上进行软件创作。微软也不失时机地在四年内接连推出 2.0、3.0、4.0 三个版本。从 VB 3.0 起，微软将 Access 的数据库驱动集成到了 VB 中，这一举措极大地提升了 VB 的数据库编程能力。而到了 VB 4.0，VB 又引入了面向对象的程序设计思想，进一步丰富了其功能。VB 不仅功能强大，而且易于学习。此外，VB 还创新性地引入了“控件”的概念，这意味着大量已经编写好的 VB 程序都可以被开发者直接拿来使用，极大地提高了开发效率。

自 2002 年开始，微软将 .NET Framework 与 Visual Basic 结合而成为 Visual Basic.NET (VB.NET)，新增了许多特性及语法，又将 VB 推向一个新的高度。最新版本 Visual Basic 2024 也带来许多令人期待的新功能。

1.2 Visual Basic 的特点

VB 的核心设计理念就是便于程序员使用，无论他们是初学者还是专家。VB 采用了一个直观的 GUI 系统，使得建立应用程序变得简单快捷，同时也不失开发复杂程序的能力。在 VB 中，程序是由基于窗体的可视化组件组合而成的，程序员可以通过添加代码来指定这些组件的属性和方法。由于组件已经默认定义了一部分属性和方法，因此程序员无须编写大量代码即可完成一个简单的程序。

窗体控件的增加和改变可以用拖放技术实现。工具箱里显示的是可用控件（例如文本框或者按钮）。每个控件都有自己的属性和事件。控件创建的时候属性会被赋予默认的值，程序员也可以修改属性值。大多数的属性值可以在程序运行的时候随着用户的动作和修改进行改动，这样就形成了一个动态的程序。例如，在窗体的大小改变事件中，我们需要加入能够自动调整控件位置的代码。当程序运行时，一旦用户更改了窗口的大小，控件的位置也会相应地自动调整。同样地，在文本框的文字改变事件中，我们也加入相应的代码，使得程序能够在用户输入文字时自动进行翻译，或者阻止某些特定字符的输入。总结起来，VB 具有以下特点：

（1）面向对象的思想

VB 采用了面向对象的设计思想，把复杂化为简单，最终实现某些算法功能。对象是指可操作实体，如窗体、窗体中的命令按钮、标签、文本框等，界面编程是指根据界面设计要求在界面上设计出窗口、菜单、按钮等类型对象，并为每个对象设置属性。

（2）事件驱动机制

在 Windows 环境下驱动事件，运行对象能响应多个不同的事件，每个事件由代码组成，代码决定了对象的算法功能。当触发事件（例如单击命令按钮）就会运行事件代码，实现相应的算法功能；不触发事件则就处于零状态，不能执行代码功能。整个应用程序是由彼此独立的事件过程构成的。

（3）集成式开发环境

VB 为编程提供了全面的集成开发环境，在这个环境中，开发者可以设计界面、编写代码、调试程序，并最终将应用编译成可在 Windows 操作系统中运行的可执行文件。此外，VB 还提供了生成应用程序安装包的集成开发环境，这为编程者发布和部署应用程序带来了极大的便利。

（4）结构化程序设计语言

VB 拥有丰富多样的数据类型，其采用的 Basic 语言完美契合了结构化设计思想，同时保持了简单易学的特性。

（5）强大数据库访问功能

VB 利用数据 Control 控件可以访问关系数据库，VB 6.0 提供了 ADOControl（数据库控制）控件，不但可以用最少代码实现数据库操作和控制，也可以取代 DataControl（数据控制）控件和 RDOControl（远程数据对象控制）控件。

(6) 支持对象链接和嵌入技术

VB 支持对象链接和嵌入 (OLE) 技术, 借助这一技术, 开发者能够将声音、图像、动画、数字处理、Web 等多种对象无缝集成到软件中, 进而打造出功能全面且强大的应用程序。

(7) 多种向导

VB 提供了多种向导, 如应用向导、安装向导、数据对象向导和数据窗体向导。

(8) 支持动态交换、动态链接技术

通过动态数据交换 (DDE) 编程技术, VB 开发应用能和其他 Windows 应用建立数据通信, 通过动态链接库技术在 VB 中可方便地调用 C 语言或汇编语言编写的程序。

(9) 联机帮助

在 VB 中, 用户可以随时通过帮助菜单或按下 F1 功能键来方便地获取所需的帮助信息。VB 的帮助窗口中展示了相关的示例, 用户可以通过复制和粘贴操作轻松获取大量的示例代码, 这为用户学习和使用代码提供了极大的便利。

1.3 安装 Visual Basic 6.0

Visual Basic 6.0 的安装文件可以在微软的官方网站上下载。下面以 Visual Basic 6.0 中文版为例, 简要介绍 Visual Basic 6.0 的安装过程。

(1) 准备阶段

首先, 确保计算机满足操作系统要求, 并已备份重要数据和设置。

其次, 解压下载的安装文件, 双击解压后的“SETUP.EXE”启动“Visual Basic 6.0 中文企业版”安装向导。

(2) 安装步骤

第一步, 在“最终用户许可协议”界面中选择“接受协议”。

第二步, 在“产品号和用户 ID”界面中输入产品 ID 号及用户信息。

第三步, 选择“安装 Visual Basic 6.0 中文企业版”。

第四步, 在“是否替换 VSS 6.0”窗口中选择“否”。

第五步, 选择安装类型及安装文件夹, 建议使用默认文件夹为安装路径, 选择“典型安装”。

第六步, 在“是否使用新的 VSS 数据库”窗口中选择“否”。

(3) 完成安装

安装程序将文件复制到硬盘中, 完成 Visual Basic 6.0 中文企业版的安装, 并要求用户重新启动计算机, 以更新系统配置使安装生效。

当重启后出现“安装 MSDN”窗口和“现在注册”窗口, 用户可以根据自己的需要进行选择和设置。

1.4 Visual Basic 的启动与退出

1.4.1 VB 6.0 的启动

VB 安装成功后，和其他 Windows 应用程序一样会在 Windows 程序组中创建快捷菜单。启动 VB 和启动其他 Windows 应用程序一样有以下 3 种方法：

(1) 从“开始”菜单启动

单击“开始”按钮，依次选择 | 程序 | Microsoft Visual Basic 6.0 | Microsoft Visual Basic6.0 启动 VB 菜单和命令，即可启动 VB。

(2) 双击桌面上的快捷方式启动

双击桌面上的 VB 6.0 快捷方式图标，即可启动 VB。

(3) 通过已建立好的 VB 工程文件启动

双击任意一个已经建立好的 VB 工程文件（.vbp），即可启动 VB 并打开该工程。

1.4.2 VB 6.0 的退出

和其他常用应用软件一样，VB 的退出有两种方法：

(1) 方法一

单击 VB 窗口标题栏右上角的关闭按钮。

(2) 方法二

通过 VB “菜单栏”中的“文件”→“退出”命令。

如果当前工程尚未保存，则弹出如图 1-1 所示对话框。



图 1-1 VB6.0 保存对话框

1.5 Visual Basic 6.0 集成开发环境

1.5.1 VB 集成开发环境

VB 集成开发环境（IDE）是一个综合性的软件工具集合，它集成了应用程序的设计、编辑、运行和调试等多种功能，为程序设计提供了极大的便利。

启动 VB，选择新建“标准 EXE”工程之后，打开 VB 集成开发环境，如图 1-2 所示。

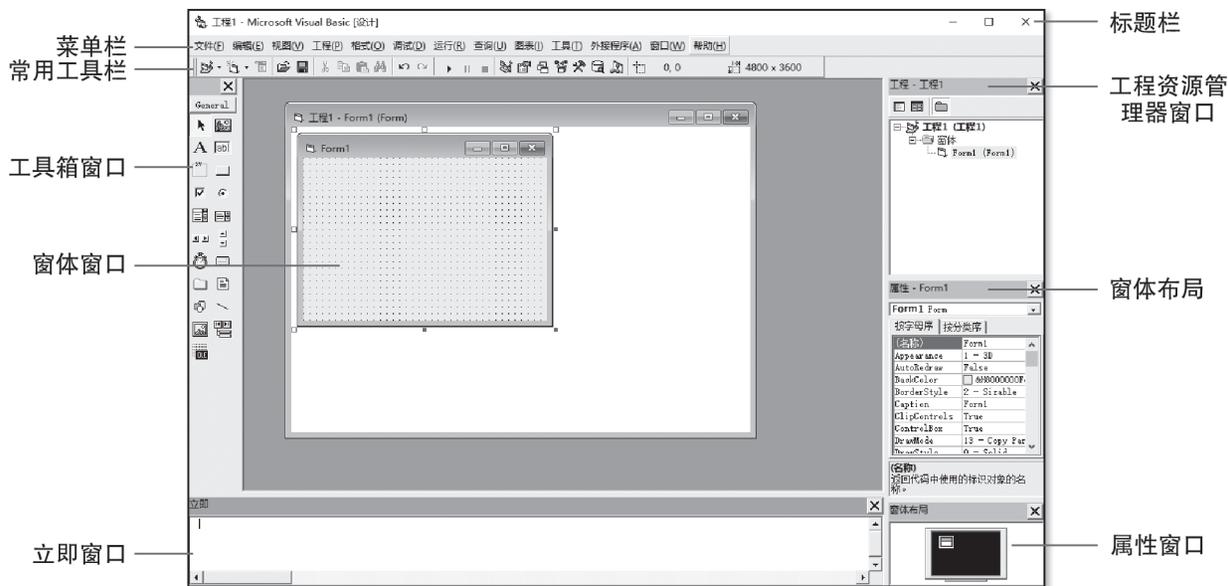


图 1-2 VB 集成开发环境

1.5.2 窗口功能简介

(1) 主窗口

主窗口即应用程序窗口，由标题栏、菜单栏和工具栏组成。

VB 包括设计模式、运行模式和中断模式三种工作模式。在标题栏上应用程序名称后面的“[]”内显示当前的工作模式，图 1-2 中的 VB 集成开发环境处于“设计”模式。

设计模式下，用户可以设计窗体、绘制控件、编写代码，并通过属性窗口方便地设置各种属性。

运行模式下，用户可以与应用程序进行实时交互，查看代码内容但无法进行编辑，也不能编辑界面。

中断模式（break）下，用户可以查看各个变量当前的值，以此来监控和判断程序是否按照预期正常执行。此时代码是可以被编辑的，但界面的编辑依然被禁止。中断模式会弹出一个“立即”窗口，允许用户输入并执行一些简短的命令。

(2) 窗体 (Form) 窗口

窗体是设计 VB 程序的界面，是应用程序最终面向用户的窗口，窗体是一个容器，上面可以放置各种控件。

(3) 代码 (Code) 窗口

代码窗口用来编辑窗体和标准模块中的代码，在窗体或控件上双击即可打开代码窗口。代码窗口包括“对象列表框”“过程列表框”“代码区”“过程查看”“全模块查看”按钮等。

(4) 属性 (Properties) 窗口

属性窗口用来显示或设置所选窗体或控件的属性。在设计模式下，属性窗口列出了当前选定窗体或控件的属性值，窗口中的属性可以按字母序或分类序两种方式排列，通过窗口的滚动条可找到各个属性，从而为其设置属性值。属性窗口由对象列表框、属性排列方式、属性列表框和属性含义说明四部分组成。

(5) 工程资源管理器 (Project Explorer) 窗口

工程是一个集合，它包含了创建应用程序所需的所有文件。在工程资源管理器窗口中，可以看到当前工程里所有的窗体和模块。这个窗口上有三个按钮，从左到右依次是“查看代码”按钮、“查看对象”按钮和“切换文件夹”按钮。通过点击“查看代码”和“查看对象”这两个按钮可以轻松实现在窗体窗口和代码窗口之间来回切换。

在工程资源管理器窗口中，含有建立一个应用程序所需要的文件的清单。工程资源管理器窗口中包含以下几类文件。

① 工程文件 (.vbp) 和工程组文件 (.vbproj)。

每个工程对应一个工程文件，该文件包含与该工程有关的全部文件和对象的清单。当一个程序包含两个或两个以上的工程时，这些工程构成一个工程组。

② 窗体文件 (.frm)。

每个窗体都对应一个独特的窗体文件，并且一个工程中可以包含多个这样的窗体文件。这些窗体文件包括了窗体及其控件的属性设置、窗体级别的变量声明、外部过程的声明、事件处理过程以及用户自定义的过程。当窗体或控件的数据中包含二进制属性，例如图片或图标时，在保存窗体文件的同时，系统会自动生成一个与之同名的 frx 文件，用于存储这些二进制数据。

③ 标准模块文件 (.bas)。

该文件是一个纯代码性质的文件，它不属于任何一个窗体，主要在大型应用程序中使用。标准模块文件主要用来声明全局变量和定义一些通用的过程，可以被不同窗体的程序调用。该文件是可选的。

④ 类模块文件 (.cls)。

VB 不仅提供了丰富多样的预定义类，还允许用户根据自己的特定需求来定义个性化的类。用户可以通过类模块自由地定义自己的类。

⑤ 资源文件 (.res)。

该文件存放的是各种“资源”，是一种可以同时存放文本、图片、声音等多种资源的文件。

(6) 工具箱 (Toolbox) 窗口

工具箱窗口默认位于窗体的左侧，它由一系列的工具图标组成。在设计时，用户可以利用这些图标在窗体上方便地放置各种控件。系统启动后，工具箱默认包含 20 个标准控件。当用户将鼠标指向某个控件时，系统会提示出该控件类的名称，例如 Label、TextBox、CommandButton 等，如图 1-3 所示。

小贴士

需要时，用户可以通过“工程”→“部件”菜单命令加载其他控件（ActiveX 控件）到工具箱中。

工具箱中的图标名称简介如下：

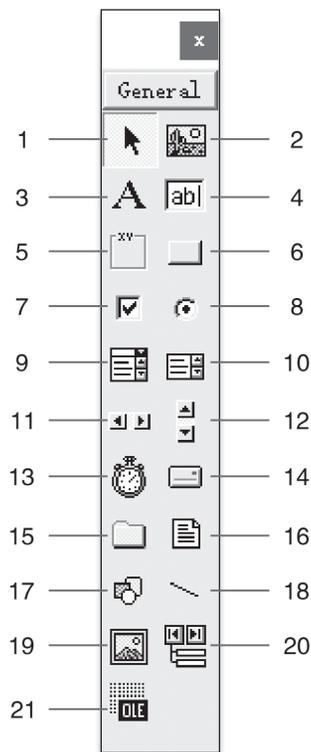


图 1-3 工具箱窗口

① 指针 (Pointer)。

指针指示窗体中的图形元素，是工具箱中唯一一个不是控件的图标。

② 图片框 (PictureBox)。

图片框控件主要用于展示图片，并且它也可以作为其他控件的容器来使用。该控件支持多种图片文件类型，包括但不限于“.bmp”“.ico”“.gif”“.jpg”等格式。

③ 标签 (Label)。

标签在运行时可显示文本信息，不能输入文本。

④ 文本框 (TextBox)。

文本框运行时既可以显示文本又可以输入文本。

⑤ 框架 (Frame)。

框架会作为容器显示其他控件。

⑥ 命令按钮 (CommandButton)。

命令按钮常用于执行指令，单击时可执行指定的操作。

⑦ 复选框 (CheckBox)。

复选框控件用于切换 ON/OFF 状态。可以使用复选框向用户提供选项，例如 True/False 或 Yes/No。

⑧ 单选按钮 (OptionButton)。

单选按钮控件用于从一组选项中选择一个选项。一组单选按钮通常放置在选项组控件中。

⑨ 组合框 (ComboBox)。

结合了文本框和列表框的功能，允许用户输入或选择数据。

⑩ 列表框 (ListBox)。

列表框控件显示一个项目列表，用户可以选择一个或多个项目。可以将列表框设置为单选或多选。

⑪ 水平滚动条 (HScrollBar)。

水平滚动条控件可以使用滚动条来改变字号大小等。

⑫ 垂直滚动条 (VScrollBar)。

垂直滚动条控件可以使用滚动条来改变图片的宽度或高度等。

⑬ 计时器 (Timer)。

在指定时间间隔内，自动且周期性地运行事件。

⑭ 驱动器列表框 (DriveListBox)。

驱动器列表框用于显示和选择计算机上的驱动器。

⑮ 目录列表框 (DirListBox)。

目录列表框用于显示和选择文件系统中的目录。

⑯ 文件列表框 (FileListBox)。

文件列表框用于显示和选择指定目录下的文件。

⑰ 形状 (Shape)。

形状控件用于在窗体上绘制基本形状，如矩形、圆形、椭圆形等，并可进行简单的形状操作。

⑱ 直线 (Line)。

直线控件用于在窗体上绘制直线，并允许用户自定义直线的颜色、宽度等属性。

⑲ 图像框 (Image)。

图像框用于在窗体中显示图片。

⑳ 数据 (Data)。

数据用于在窗体上实现数据的展示、输入、编辑和处理功能。

㉑ OLE 容器。

OLE 容器控件用于在窗体上嵌入和显示 OLE 对象, 如 Excel 表格、Word 文档等, 实现与其他应用程序的数据交互和集成。

(7) 其他窗口

① 立即 (Immediate) 窗口。

调试程序时, 可以直接在立即窗口中用 Print 方法或在程序中用 Debug.Print 方法显示表达式的值。

② 窗体布局 (Form Layout) 窗口。

窗体布局窗口主要用来指定应用程序运行时窗体的初始位置。

③ 对象浏览器 (Object Browser) 窗口。

在“视图”菜单中选择“对象浏览器”或通过按下 F2 键, 可以打开“对象浏览器”窗口, 用于查看工程中定义的模块或过程, 也可浏览对象库、类型库、类、方法、属性、事件以及过程中使用的常数。

1.6 程序设计方法的发展历程

自 1946 年第一台计算机诞生以来, 计算机技术迅猛发展, 程序设计语言历经了从机器语言、汇编语言到高级语言的多个阶段, 同时程序设计方法也在不断发展和提升。

1.6.1 早期的程序设计

人类最初的编程语言是由计算机能直接识别的二进制指令构成的机器语言。显然, 这种语言对计算机而言易于识别, 但对人类来说却显得极为晦涩难懂。在这一阶段, 人类的自然语言与计算机编程语言之间存在着巨大的鸿沟。这属于面向计算机的程序设计阶段。设计人员的核心关注点在于如何让程序尽可能地被计算机接受并按指令准确无误地执行, 而程序是否易于被人类理解则并不被视为重点。软件开发人员只能是少数软件工程师, 因此软件开发的难度大, 周期长, 而且开发出的软件功能很简单, 界面也不友好, 计算机的应用仅限于科学计算。

随后, 汇编语言应运而生, 它将机器指令映射为一些易于理解的助记符, 如 ADD、SUB 等。这一发展阶段使得汇编语言与人类的自然语言之间的鸿沟略有缩小, 但仍然与人类的思维方式相去甚远。程序员在使用汇编语言时, 仍需考虑大量的机器细节。因此, 这一阶段的程序设计仍然非常注重计算机的硬件系统, 它仍然属于面向计算机的程序设计范畴。

1.6.2 结构化程序设计

1965 年，荷兰的计算机科学家 E.W. Dijkstra 提出了结构化程序设计方法，他的主要观点是采用自顶向下、逐步求精及模块化的程序设计方法，即任何程序都可由顺序、选择、循环三种基本控制结构构造。结构化程序设计得益于模块分化与功能分解、自顶向下等方法的采用，能够将一个较为复杂的问题拆解成多个功能独立的子问题或模块。这些子问题可以分别由不同的人员负责解决，从而有效地解决了多人协作开发大型软件时，如何高效地完成高可靠性系统的问题。

尽管如此，结构化程序设计仍面临两个未得到妥善解决的问题。首先，模块化主要聚焦于控制流，仍存在与人的思维方式不协调之处，因此难以自然、准确地反映真实世界的情况。其次，该方法在实现过程中过于强调功能模块的操作方法，而被操作的数据则处于从属地位，导致程序模块和数据结构之间的耦合关系较为松散。这种设计使得程序在复杂度增加时更容易出错，且难以维护。

1.6.3 面向对象程序设计

面向对象的程序设计是在 20 世纪 80 年代初提出的，其起源于 Smalltalk 语言。这种方法为使软件容易在程序设计中模仿建立真实世界模型的方法，对系统的复杂性进行概括、抽象和分类，使软件的设计和实现形成一个由抽象到具体、由简单到复杂的循序渐进的过程，从而解决了大型软件开发过程中存在的效率低、调试复杂、维护困难等问题。

当然，面向对象的程序设计并非意在摒弃结构化程序设计方法，而是立足于一个比结构化程序设计更高、更抽象的层次去解决问题。在将其分解为低级代码模块时，仍然需要运用结构化编程的技巧。然而，在将一个重大问题分解为小问题这一过程中，面向对象程序设计所采取的思路却是别具一格的。

结构化的分解突出过程，即如何做，即代码的功能是如何得以完成的。面向对象的分解突出真实世界和抽象的对象，即做什么，它将大量的工作交由相应的对象来完成。

面向对象程序设计更符合人们的思维习惯，便于对复杂问题进行分析。在软件的维护和功能的增减方面，它也表现出了更大的易实现性。通过继承的方法，面向对象程序设计提高了软件开发的效率。同时，与可视化技术相结合，它实现了用户界面的图形化，使得用户界面更加美观、友好。

1.7 创建应用程序的过程

VB 拥有“所见即所得”的友好界面设计特性，但应用程序设计并不仅仅局限于界面设计，程序设计同样是非常重要的部分。因此，创建一个 VB 应用程序需要完成两方面的工作：一是 Visual 可视化界面设计，二是 Basic 程序设计。

1.7.1 VB 工程管理

在 VB 开发环境中，创建一个应用程序，被称为建立一个工程。一个 VB 工程是由若干个不同类型的文件组成的，工程就是这些文件的集合。一个 VB 工程通常包含一个工程文件（.vbp）和若干个窗体文件（.frm），有时根据需要也会包含其他类型的文件，如标准模块文件（.bas）、类模块文件（.cls）、资源文件（.res）、自定义控件文件（.ocx）与用户文档（.dob 或 .dox）等。为方便使用和管理，保存工程时，建议将工程中的相关文件都保存在一个独立的文件夹中。

（1）工程文件

在创建一个 VB 工程时，系统会建立一个扩展名为“vbp”的工程文件。工程文件的主要作用是记录在创建该工程时所建立的所有文件的相关信息。但值得注意的是，尽管工程文件包含了其他文件建立时的相关信息，但它并不包含这些文件的详细内容，因此它并不能代表工程的全部内容。因为工程文件记录了工程中所有文件的相关信息，当打开一个已建立的工程中的工程文件时，会同时打开该工程中所有其他相关的文件。对于一个新建立的工程，初次保存工程时，系统会逐个提示保存所有文件，而对于一个已建立的工程，如再次打开进行修改，只要执行保存工程命令，即可对所有文件的修改进行保存，而不必逐个保存。

（2）窗体文件

窗体文件也称为窗体模块文件。由于窗体是创建 VB 应用程序界面时必不可少的对象，因此窗体和窗体文件是 VB 中最重要的对象和文件，一个 VB 工程必须至少包含一个窗体，最多可包含 255 个窗体，每个窗体都有一个对应的窗体文件（.frm）。窗体文件不仅包含了用于处理窗体及其内部对象的事件过程，还包含了窗体及窗体中各个对象的属性设置和相关的说明信息。因此，窗体文件可以综合概括为：窗体文件 = 窗体界面 + 窗体程序代码。



小贴士

标准模块文件的主要作用是将应用程序中能够被多个模块共同使用的程序代码段（即通用过程）集中组织在一起。同时，对于需要在多个模块之间共享的全局变量，也通常被定义在标准模块中。与窗体模块不同，标准模块仅包含程序代码，而没有与之对应的用户界面。根据具体需求，一个 VB 工程可以包含多个标准模块文件，也可以不包含任何标准模块文件。

类模块文件主要用来创建新的类，并对类的属性和方法进行规定。和标准模块文件类似，在一个 VB 工程中，类模块文件也是可选的。

1.7.2 程序设计步骤

在使用 VB 实际创建一个应用程序之前，应先做好系统需求和功能分析，确定数据来源、数据处理方法等。在此基础上，就可以启动 VB 系统，进入程序的实际创建过程。启动 VB 后，系统总是将新建工程命名为“工程 1”（Project 1）。

以下是用 VB 系统创建应用程序的一般步骤：

（1）创建程序界面

程序界面作为程序与用户之间交互的桥梁，主要由窗口、窗口内的各种按钮、文本框、菜单栏、工具栏等元素构成。创建程序界面实质上是根据程序的功能需求、程序与用户间信息传递的形式和内容，以及程序的工作方式等因素，来确定窗口的大小和位置，明确窗口中包含哪些对象，进而利用窗体设计工具来绘制和布置所需的控件对象。

（2）设置对象的属性

在创建程序界面的过程中，应根据实际需求，同时为窗体及其上的对象设置相应的属性。这些属性的设置既可以在设计时通过属性窗口进行，也可以通过程序代码在程序运行时进行动态改变。

（3）编写程序代码

界面仅决定程序的外观。程序通过界面上的对象接收到必要的信息后如何动作，要做些什么操作，对用户通过界面输入的信息做出何种响应、进行哪些信息处理，还需要通过编写相应的程序代码来实现。编写程序代码可通过代码编辑器进行。

（4）保存工程

一个 VB 工程（程序）创建完成以后，可使用“文件”菜单中的“保存工程”命令或工具栏上的“保存工程”按钮进行保存。初次保存时，应根据系统提示依次对所有文件进行保存。一个工程中的所有文件最好都保存在同一个独立的文件夹中，这样有利于管理和使用。

（5）测试和调试应用程序

测试和调试程序是确保所开发的程序能够达到预期功能，并保证其运行正确、稳定可靠的必要环节。

（6）创建可执行程序

创建可执行程序的过程，实际上就是将整个工程编译成一个可执行文件（.exe 文件），这样，该程序就能脱离 VB 开发环境，直接在 Windows 操作系统下独立运行。

【例 1-1】设计一个应用程序，要求在屏幕上显示“建设美丽中国”，字号为三号，颜色为红色，单击窗体，窗体上会出现两个命令按钮，分别为“继续”按钮和“结束”按钮。单击“继续”按钮，窗体上会显示“绿水青山就是金山银山”。单击“结束”按钮，程序运行结束。

① 创建用户界面。

执行“文件”→“新建工程”命令，先在弹出的“新建工程”对话框中单击“标准 EXE”图标，然后单击“确定”按钮，创建一个默认名称为“工程 1”的工程，在此工程中有一个默认名称为“Form1”的窗体。在窗体“Form1”中添加控件。

单击工具箱中的命令按钮图标，光标会变成“+”，在窗体上移动光标到合适的位置，绘制命令按钮 Command1，用同样的方法绘制出命令按钮 Command2。单击工具箱中的标签图标，光标会变成“+”，在窗体上移动光标到合适的位置，绘制一个标签 Label1。

② 设置属性。

在创建用户界面后，可以通过“属性”窗口设置窗体及每个控件的属性。各个对象属性的设置如表 1-1 所示。

表 1-1 各个属性的设置

对象	属性	属性值
Form1	Caption	绿色发展观
Command1	Caption	继续
	Visible	False
Command2	Caption	结束
	Visible	False
Label1	Caption	建设美丽中国
	Visible	True
	Alignment	2-center

③ 编写代码。

双击窗体，出现代码窗口，在 Form_Click（）过程中编写如下事件过程代码：

```
Private Sub Form_Click ()
    Command1.Visible = True
    Command2.Visible = True
End sub
```

在 Command1_Click() 过程中编写如下事件过程代码：

```
Private Sub Command1_Click ()
    Label1.Caption = "建设美丽中国"
End sub
```

在 Command2_Click（）过程中编写如下事件过程代码：

```
Private Sub Command2_Click ()
    End
End sub
```

④ 运行程序。

执行“运行”→“启动”命令，即可运行程序。

单击窗体，显示“继续”按钮和“结束”按钮。

单击“结束”按钮，退出应用程序。

⑤ 保存。

首先，执行“文件”→“保存工程”命令，在打开的“文件另存为”对话框中，选择要保存窗体文件的目录及文件名，单击“保存”按钮，保存窗体文件。

其次，在弹出的“工程另存为”对话框中，选择要保存工程文件的目录及文件名，单击“保存”按钮，保存工程文件。

操作演练

创建一个 VB 应用程序，要求运行程序时，在窗体空白处双击鼠标，窗体上显示“VB 欢迎您”，用鼠标单击窗体上的“显示”按钮，窗体上显示“欢迎学习 VB 程序设计”，如图 1-4 所示，用鼠标单击窗体上的“退出”按钮，程序终止运行。

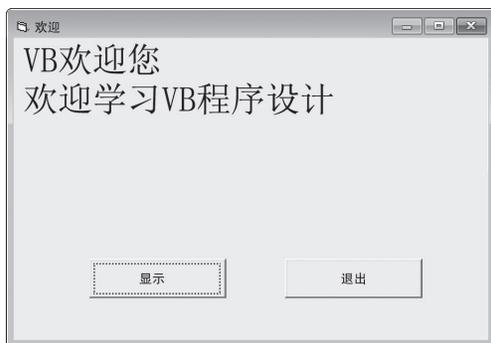


图 1-4 程序运行时用鼠标单击“显示”命令按钮后的运行效果

实现过程如下：

(1) 新建工程

启动 VB 6.0，在“新建工程”对话框中的“新建”选项卡中选择“标准 EXE”项，单击“打开”后进入编辑窗口。

(2) 添加控件

根据实际需求，在窗体设计器上精心设计应用程序的界面，并在窗体上添加两个按钮控件，如图 1-5 所示。

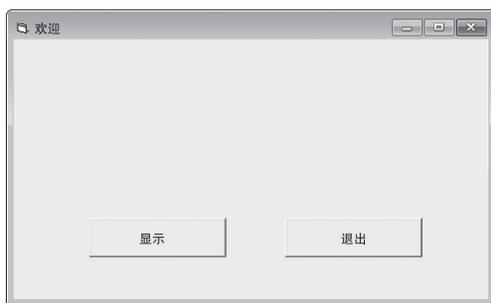


图 1-5 在窗体上添加控件

(3) 设置属性

按表 1-2 所示设置好各控件的属性。

表 1-2 对象属性设置

对象	属性	设置
Form1	Name	Form1
	Caption	欢迎
Command1	Name	Command1
	Caption	显示
Command2	Name	Command2
	Caption	退出

(4) 编写事件过程代码

在代码窗口中给相应对象添加事件过程，即编写程序代码。该程序的代码如下：

```
Private Sub Command1_Click ( ) ' “显示” 按钮的单击事件
    Print " 欢迎学习 VB 程序设计 "
End Sub

Private Sub Form __ DblClick ( ) ' 窗体的双击事件
    Print "VB 欢迎您 "
End Sub

Private Sub Command2_Click ( ) ' “退出” 按钮的单击事件
    End
End Sub
```

(5) 运行应用程序

按 F5 键或工具栏上的“启动”按钮或选择“运行”→“启动”命令进行程序的调试运行。运行效果如图 1-4 所示。

(6) 保存应用程序

当程序调试成功后，进行工程的保存操作。首先，通过“文件”→“保存工程”命令，保存窗体设计，此时生成的窗体文件扩展名为“.frm”，如图 1-6 所示。接着，再保存工程，此时生成的工程文件扩展名为“.vbp”，如图 1-7 所示。

(7) 生成 EXE 文件

程序设计完成并经过成功测试后，还可以将其编译成可直接执行的 EXE 文件。这种类型的文件具有独立性，可以脱离 VB 环境运行。

具体操作：选择“文件”菜单下的“生成 .EXE”命令，出现“生成工程”对话框，如图 1-8 所示，选择保存文件夹，输入文件名，单击“确定”按钮，即可完成 EXE 文件的生成。

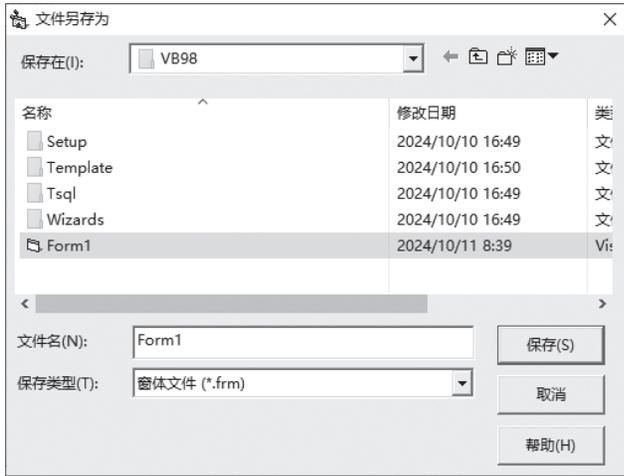


图 1-6 窗体另存对话框

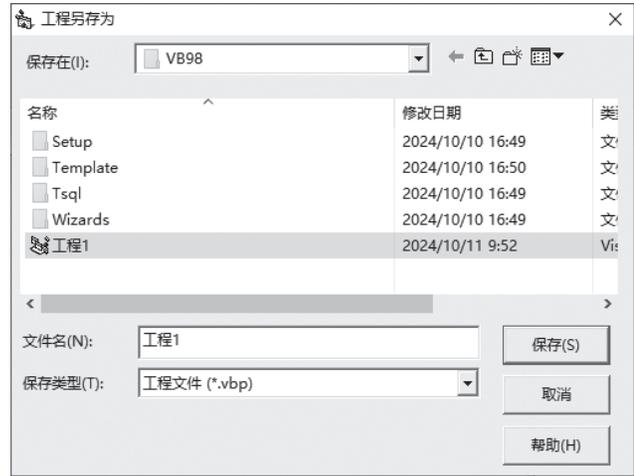


图 1-7 工程另存对话框

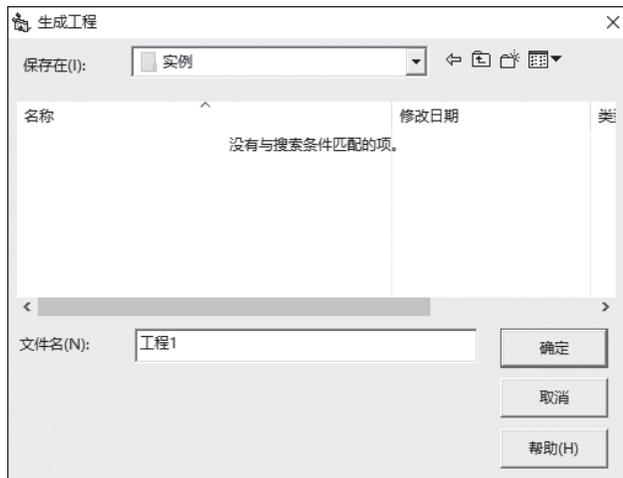


图 1-8 “生成工程”对话框

第 2 章 对象、窗体与控件

学习目标

知识目标

- ▶ 深入理解对象在 Visual Basic 中的核心概念及其重要性。
- ▶ 熟练掌握窗体和控件的基本属性、方法及事件处理机制。

能力目标

- ▶ 能够灵活运用 Visual Basic 中的窗体和控件进行复杂的界面布局与设计。
- ▶ 具备根据实际需求，自定义控件属性并实现特定功能的能力。

素质目标

- ▶ 培养对界面设计的审美意识和用户体验的关注，注重细节处理。
- ▶ 提升问题解决能力，能够在设计界面或使用控件时迅速找到问题的解决方案。

章节概述

在 Visual Basic 编程中，对象、窗体与控件是构建应用程序的核心要素，通过对象的组织和管理、窗体的设计以及控件的配置和操作，共同实现用户界面和功能逻辑的直观呈现与高效交互。本章将详细介绍对象、窗体与控件。首先，我们将探讨“对象”这一基本概念，讲解其在 Visual

Basic 中的作用和如何利用对象进行程序设计。接下来，我们将深入分析“窗体”，即应用程序的用户界面窗口，说明如何设计和管理窗体以创建直观的用户交互体验。最后，本章将介绍“控件”的概念，包括常用控件的类型、功能和使用方法，帮助读者理解如何通过控件实现不同的用户界面功能。

2.1 对象

面向对象程序设计的核心是对象，VB 程序设计的过程就是与 VB 提供的大量对象进行交互的过程。

2.1.1 类与对象

(1) 类

类是创建对象实例的模板，它是对同种对象的集合进行抽象的结果，包含了定义这些对象所需的所有属性描述和行为特征。在 VB 中，类可以由系统预先设计并提供，如工具箱上的各种图标所示，它们就是 VB 系统事先设计好的标准控件类。同时，程序员也可以根据实际需求，自行设计并创建类。

(2) 对象

对象是具有某些特征的具体事物的抽象。现实世界中的各种各样的实体都可以称为对象，例如一个人、一台机器、一个球等都是对象。每个对象都具有描述其特征的属性及附属于它的行为。在面向对象的程序设计中，对象是类的一个实例，继承了类的属性、方法，通过将类实例化，可以得到真正的控件对象。在 VB 中，当在窗体上绘制一个控件时，实际上是将类实例化，从而创建了一个具体的控件对象，简称控件。在 VB 中，对象主要包括两大类：一类是窗体，另一类是控件。

小贴士

窗体是一个特殊的对象，因为它既可以作为类来定义窗体的属性和行为，也可以作为对象来实际展示和操作。当向一个工程添加一个新窗体时，实质上是使用窗体类来创建一个新的窗体对象。

2.1.2 对象的属性、事件和方法

VB 的对象是具有特殊属性和行为方法的一个可视化实体，每一个对象均有自己的特殊属性、事件和方法，称为对象的三要素。

(1) 属性

属性即对象的性质，即用来描述和反映对象特征的参数。对象中的数据就保存在属性中，不同

的对象有各自不同的属性。例如，在描述一个人时，有

姓名 = 小红

性别 = 女

年龄 = 20

民族 = 汉

上述式中，左侧的姓名、性别、年龄和民族成为对象的属性名，等号右侧的数据称为属性值。

在 VB 中设置对象的属性有两种方法：

- ① 在设计阶段选中对象后利用属性窗口直接设置对象的属性。
- ② 在代码窗口中编写代码，格式为：

```
对象名.属性名 = 属性值
```

例如，Label1.Caption = " 欢迎使用 VB6.0 "，可以将标签 Label1 的 Caption 属性设置为“欢迎使用 VB6.0”。

(2) 事件、事件过程和事件驱动

① 事件。

事件是发生在对象上的一个动作，该动作能够被对象所识别。在用户与应用程序的交互过程中经常会导致此类事件的发生。例如，当用户点击 Command1 按钮时，就会触发 Command1 的单击 (Click) 事件。VB 系统为每个对象预先定义了一系列的事件，除了单击之外，还有双击 (Db1Click)、鼠标滑过 (MouseMove)、键盘按下 (KeyPress) 等。

② 事件过程。

当事件在对象上发生时，应用程序需要负责对该事件进行处理。处理事件的具体步骤构成事件过程。每一个事件过程都是专门为响应某个对象的某一特定事件而设计的。在 VB 程序设计中，主要工作就是为对象编写可能触发事件过程的程序代码。事件过程的一般形式为：

```
Private Sub 对象名_事件 ([参数列表])  
..... '事件过程代码  
End Sub
```

例如，单击命令按钮 Command1 时，将标签的内容设置为“欢迎使用 VB6.0”，则对应的事件过程为：

```
Private Sub Command1_Click ( )  
Label1.Caption=" 欢迎使用 VB6.0"  
End Sub
```

在具体编程中，只要在代码窗口的顶部正确选中了编程对象以及该对象的某一事件，系统就会自动生成对应的事件过程框架。此时，用户只需在这个事件过程框架中编写当该事件发生时具体要执行的操作所对应的程序代码即可。另外，当用户对一个对象执行一个动作时，有可能会在该对象

上同时触发多个事件，例如，当程序运行时在文本框 Text1 中输入 a 时，会触发 Change、KeyPress、KeyDown、KeyUp 事件。

小贴士

实际设计程序时，我们只对感兴趣的事件过程编码，没有编码的为空事件过程，系统不会处理该事件过程。

③ 事件驱动。

在面向过程的程序设计中，程序的执行顺序完全由设计人员编写的代码所决定。应用程序自身负责控制哪段代码将被执行，以及这些代码将遵循怎样的顺序来执行。用户无法对程序的执行流程进行更改或调整。

在面向对象的程序设计中，程序采用事件驱动的编程机制来执行。这意味着程序在执行时，会一直处于等待事件发生的状态。当某个事件发生后，程序会立即去执行与该事件对应的事件处理过程。执行完毕后，程序会继续等待其他事件的发生。即事件发生的顺序决定了代码执行的顺序。因此，程序多次运行时所经过的代码路径可能是不同的。若没有事件发生，则整个程序一直处于等待状态。

VB 程序的执行步骤如下：

- ① 启动应用程序，装载和显示窗体。
- ② 窗体或窗体上的对象等待事件的发生。
- ③ 事件发生时，执行相应的事件过程。
- ④ 重复执行步骤②和③。

如此反复，直到遇到“End”结束语句，结束程序运行或单击“结束”按钮强行停止程序执行。

(3) 方法

面向对象的程序设计中，对象除有属于自己的属性和事件外，还拥有自己的行为，即方法。在 VB 中，系统已经将一系列通用的过程和函数编写并封装好，形成了用户可以直接调用的方法。因此，当用户调用一个方法时，实质上就是在执行该对象内部的某个具体的函数或过程。因为方法是面向对象的，所以在调用方法时一定要指明对象名。对象的方法调用格式为：

```
[对象名.]方法名[参数列表]
```

其中，若省略了对象名，则表示调用当前对象的方法，一般指窗体。

例如：

```
Form1.Print " 欢迎使用 VB6.0 "
```

该语句的作用是调用窗体的 Print 方法，在窗体上显示“欢迎使用 VB6.0”。

2.2 窗体

窗体是设计图形用户界面的基础平台，它在应用程序运行时作为用户与程序交互的实际窗口。窗体充当着所有控件的容器，类似于一块“画布”，在设计用户界面时，开发者就像摆放积木一样，将工具箱中的控件“摆放”在这块画布上。

窗体的属性决定了其外观和操作方式。大部分属性既可以在设计状态下通过属性窗口进行设置，也可以在程序运行时通过编程语句进行动态设置。然而，有少量属性仅限于在设计时或运行时进行设置。

2.2.1 主要属性

下面（1）～（8）中所列属性为窗体和多数控件的通用属性。

（1）Name 属性

Name 属性表示对象的名称，它通常作为标识符在程序中被引用，但不会显示在窗体或控件的界面上。该属性只能在设计时进行修改，是所有对象都具有的一个基本属性。

（2）Caption 属性

Caption 属性是标题属性，用于定义窗体标题栏上显示的内容，或者控件上显示的文本内容。

（3）Enabled 属性

Enabled 属性用于决定对象是否可用，即能否对用户的操作做出响应。该属性的值只有 True 和 False 两种。当属性值为 True 时，允许用户进行操作，并且对象会对操作做出响应；而当属性值为 False 时，对象在运行时将呈现灰色，表示其处于不可用状态。

（4）Visible 属性

Visible 属性用于决定运行时对象是否可见。当属性值为 True 时，对象处于可见状态；而当属性值为 False 时，对象会被隐藏起来，用户无法看到，但控件本身仍然存在。

（5）Height（高）、Width（宽）、Left（左）和 Top（顶）属性

这四个属性决定了对象的大小和位置，单位是 Twip（缇，1 cm=567 缇）。对于窗体而言，Top 属性表示窗体顶部到屏幕顶部的距离，Left 属性表示窗体左边到屏幕左边的距离。而对于控件来说，这两个属性则表示控件相对于其所在窗体的位置，如图 2-1 所示。控件的大小和位置既可以通过鼠标拖动来直观地修改，也可以通过属性窗口进行精确的设置，如图 2-2 所示。

（6）ForeColor 和 BackColor 属性

ForeColor 和 BackColor 属性分别用于表示前景颜色（即正文颜色）和背景颜色。它们的值通常用十六进制常数来表示，同时也可以通过调色板或系统颜色来进行选择和设置。

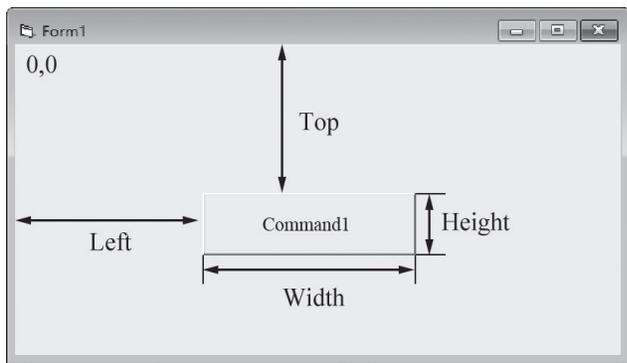


图 2-1 控件位置属性示意图



图 2-2 位置类属性

(7) Font 属性

Font 属性用来设置窗体或控件上文本字体的类型、大小、效果等外观特征，包括 FontName（字体，默认宋体）、FontSize（字号）、FontBold（是否为粗体，值为 True 时为粗体）、FontItalic（是否为斜体，值为 True 或 False）、FontStrikethru（是否加删除线，值为 True 或 False）、FontUnderline（是否加下划线，值为 True 或 False）这 6 个子属性。

(8) Picture 属性

Picture 属性是用于设置窗体或控件中要显示的图片的属性。在属性窗口中进行设置时，用户需要在弹出的对话框中选择想要显示的图片；而在编写程序时，则需要使用 LoadPicture 函数来加载并应用所选的图片，它的一般形式为：

```
对象 .Picture=LoadPicture ( " 图形文件源路径及文件名 " )
```

在程序中清除窗体或控件中显示的图片，仍需使用 LoadPicture，其格式为：

```
对象 .Picture=LoadPicture ( " " )
```

(9) MaxButton 和 MinButton 属性

MaxButton 和 MinButton 属性专门用于设置窗体上的最大化按钮和最小化按钮的显示状态，只能在属性窗口中进行调整。这两个属性在默认情况下值都是设置为 True，当默认值设置为 False 时，窗体上将不会显示最大化和最小化按钮。

(10) Icon 和 ControlBox 属性

Icon 和 ControlBox 是图标和控制菜单属性。在属性窗口中，通过设置 Icon 属性可以选择一个图标文件（格式为 .ico 或 .cur），窗体在最小化状态时将会显示这个图标。ControlBox 属性默认情况下值是 True，当默认值设置为 False 时，窗体左上角将不会显示控制菜单框。

(11) BorderStyle 属性

BorderStyle 属性用于设置窗体的边框样式。该属性值为 0 ~ 5，决定窗体是否可以移动或改变大小等。默认值为 2-Sizable，窗体为双线边框，可移动并可以改变大小。

(12) WindowsState 属性

WindowsState 属性用于设置窗体的状态。该属性值可以为 0、1、2，分别对应正常状态、最小化

状态和最大化状态。

(13) StartUpPosition 属性

StartUpPosition 属性表示窗体的初始位置，该属性值为 0 ~ 3，默认值为 3，设置为 2 时，启动程序时窗体位于屏幕中心位置。

2.2.2 事件

窗体能够响应多种事件，其中 Click 事件、DbClick 事件和 Load 事件是最常用的。此外，窗体还支持其他事件，如 Activate（窗体激活）、Deactivate（未激活）、Initialize、Paint、Resize 和 Unload 等。在代码窗口的过程列表框中，可以查看和编写所有这些事件的处理过程。

(1) Form_Click

单击事件，是在窗体上单击鼠标左键时发生的事件。

(2) Form_DblClick

双击事件，是在窗体上双击鼠标左键时发生的事件，同时激发单击事件。

(3) Form_Load

装载事件，是当窗体从磁盘装入内存时触发的事件，通常运行程序时自动装入启动窗体即触发了 Load 事件，用于对属性和变量进行初始化。Unload 事件在窗体被卸载时发生。

(4) Form_Initialize

当应用程序创建一个窗体时，会触发 Initialize 事件，该事件能够初始化窗体所需的数据。值得注意的是，窗体的 Initialize 事件发生在其 Load 事件之前。

2.2.3 方法

窗体的方法包括 Print、Cls、Move、Show、Hide 和 Refresh 等。

(1) Print 方法

窗体的 Print 方法用于在窗体上输出信息，同样，图形框（PictureBox）也配备了 Print 方法。

Print 方法的一般形式为：

```
[对象名.]Print[定位函数][表达式列表][分隔符]
```

其中 [] 内为可省参数；对象可以是窗体、图形框或打印机，若省略对象名即为在当前窗体上输出；定位函数包括 Tab (n) 和 Spc (n)。

① Tab (n) 表示定位于从对象最左端算起的第 n 列。如果 n<1，则 Tab 将输出位置移动到第 1 列；若 Tab (n) 中 n 值小于当前位置的值，则重新定位于下一行的 n 列。

② Spc (n) 的作用是输出时跳过 n 个空格。

③ 无定位函数时，由对象的当前位置决定输出项的位置。

④ 表达式列表是要输出的数值或字符串表达式。

⑤ 分隔符包括分号和逗号，用于决定输出后光标的定位方式。分号表示以紧凑格式输出，光标将定位在上一个字符后面；逗号表示以标准格式输出，光标将定位到下一个打印区的开始位置，每个打印区宽度为 14 列。若输出列表后无分隔符，则表示输出后换行，光标将定位在下一行的行首。

例如：

```
Print Tab ( 3 ) ; "abcd" ' 定位到第 3 列，输出 "abcd" 后换行
```

```
Print Spc ( 3 ) ; "ef" ' 跳过 3 个空格（定位到第 4 列），输出 "ef" 后换行
```

```
Print "welcome": ' 输出 "welcome"，并把光标定位到最后一个 e 字母之后
```

```
Print "to China" ' 在光标当前位置（即上一行的 e 字母后）输出 "to China" 后换行
```

```
Print " 欢迎学习 "; Tab ( 3 ) ; "VB6.0"
```

’ 【补充说明】执行逻辑分三步：

- ’ 1. 先输出 " 欢迎学习 "，因末尾是分号（；），光标停在 " 习 " 字后面，不换行、不跳列；
- ’ 2. 执行 Tab(3)；从窗体（或图形框）最左端开始计数，强制将光标定位到第 3 列（列序号从 1 开始）；
- ’ 3. 最后输出 "VB6.0"：从第 3 列位置开始显示，最终呈现“欢迎学习 VB6.0”效果（Tab 定位会覆盖之前光标位置）

运行效果如图 2-3 所示。

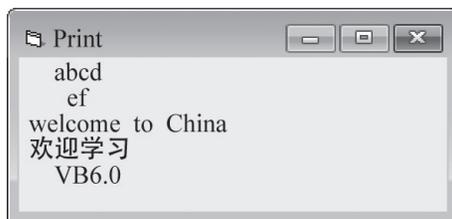


图 2-3 Print 方法练习

Print 方法不仅有输出功能还有计算功能，可以对表达式的值先计算后输出。

(2) Cls 方法

Cls 方法用于程序运行时清除窗体上所有已生成的文本和图形内容。格式为：

```
[ 对象 .] Cls
```

其中，“对象”为窗体或图形框，省略为窗体。例如，清除图 2-3 在运行时输出的内容，代码为：

```
Cls
```

(3) Move 方法

Move 方法用于将窗体或控件移动到指定的位置，并可以同时调整其大小。Move 方法的一般形式为：

```
[ 对象名 .] Move 左边距离 [, 上边距离 [, 宽度 [, 高度 ]]
```

例如：

```
Form1. Move Form1. Left+100, Form1. Top, Form1. Width/2, Form1. Height*2
```

表示将窗体水平向右移动 100 缇（Twip），同时窗体的宽度缩小为原来的一半，高度变为原来的两倍。

如果希望向上移动 100 缇，左右位置不变，大小不变，则代码为：

```
Form1.Move Form1.Left, Form1.Top-100
```

【例 2-1】编写一个标签随鼠标移动的程序。

分析：在窗体上摆放一个标签 Label1，然后在代码窗口输入以下代码即可。

```
Private Sub Form_Load ( )  
    Label1.Caption=" 和鼠标一起移动 "  
End Sub  
  
Private Sub Form_MouseMove ( Button As Integer, Shift As Integer, X As Single, Y As Single )  
    Label1.Move X, Y ' 鼠标移动时使标签移动到鼠标所在的位置  
End Sub
```

(4) Show 方法

Show 方法用来显示窗体，Show 方法的一般形式为：

```
对象 .Show ([ 模式 ])
```

模式分 0 和 1 两种，0-Modeless（非模式），可以对其他窗体进行操作；1-Modal，关闭才能对其他窗体进行操作。默认为 0。

例如：

```
Form2.Show
```

将窗体 Form2 显示出来。Show 方法有自动装载窗体的功能，如果在 Form_Load 事件内打印信息，必须使用 Show 方法或者把窗体的 AutoRedraw 属性设为 True，否则，当程序运行时窗体上什么都不显示。

(5) Hide 方法

Hide 方法用于隐藏窗体，使其变得不可见。当窗体处于隐藏状态时，将无法访问该窗体中的任何控件对象。需要注意的是，Hide 方法仅负责隐藏窗体，并不会卸载窗体，这意味着窗体仍然保留在内存中。若要卸载窗体并释放其占用的内存资源，需要使用 Unload 语句。隐藏窗体和卸载窗体的一般形式为：

```
对象 .Hide  
Unload 对象
```

(6) Refresh 方法

Refresh 方法用于对一个窗体执行全面的重绘操作。如果没有事件发生，窗体或控件对象的绘制是由系统自动处理的，无需使用 Refresh 方法，但是如果希望窗体显示内容、文件列表框或目录列表框等立即更新时，可使用 Refresh 方法。

2.3 控件

VB 提供了种类繁多的控件，这些控件大致可以划分为三类：标准控件、ActiveX 控件以及可插入对象。其中，标准控件也被称为内部控件，是 VB 工具箱中最常见的控件，VB 中共有 20 个这样的控件。接下来将详细介绍几个在 VB 中常用的基本控件。

2.3.1 标签

标签主要用于显示一小段文本，通常用来标注本身不具有 Caption 属性的控件，例如文本框。

(1) 常用属性

① Caption 属性。

Caption 属性用于设置标签中显示的内容。

② Alignment 属性。

Alignment 属性用于设置对齐方式，其中 0 表示标题靠左对齐，1 表示标题靠右对齐，而 2 则表示标题居中对齐。

③ AutoSize 属性。

AutoSize 属性用于设置标签是否自动调整其大小。当此属性值为 True 时，标签会根据其内容自动调整大小；当其值为 False 时，标签将保持设计时的大小不变。

④ BorderStyle 属性。

BorderStyle 属性用于设置控件的边框风格。默认情况下其值为 0，表示标签无边框；若将此属性值设置为 1-Fixed Single 时，则标签将显示边框。

⑤ BackStyle 属性。

BackStyle 属性用于设置背景风格。默认为 1，标签覆盖背景；设置为 0-Transparent 时，标签透明，看起来的效果就像标题直接显示在窗体上。在图 2-4 中显示的是 3 个标签的不同设置效果。

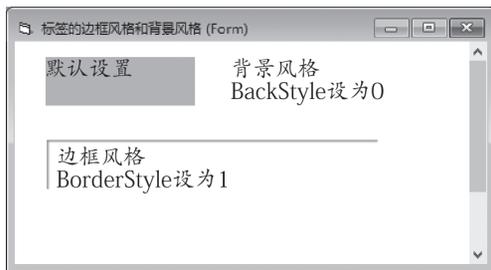


图 2-4 标签属性示例

(2) 常用事件

标签控件虽然能够触发如 Click 和 DblClick 等事件，但由于它通常被用来显示标题或文字说明，

因此很少在事件过程中使用它。

(3) 常用方法

标签也具有 Move 方法，和窗体的 Move 方法一样使用。

2.3.2 文本框

文本框是一个文本编辑区域，为用户提供输入、编辑、修改和显示文本内容的功能。

(1) 常用属性

① Text 属性。

Text 属性用于设置文本框中显示的文字内容。

② MaxLength 属性。

MaxLength 属性用于限定文本框中用户最多可以输入的字符数，其中无论是西文字符还是汉字，都被计为一个字符。缺省值为 0，对于单行显示的文本框，用户能够输入的字符数上限受到内存限制。而对于多行显示的文本框，最多可输入的字符数大约为 32K。将 MaxLength 设置为任何大于 0 的数可设置文本框最多允许输入的字符数，运行过程中输入超过允许的字符个数时，文本框不再接收输入的字符，并有响声提示。

③ MultiLine 属性。

MultiLine 属性决定文本框是否可以多行显示。缺省值为 False，全部内容只能在一行上显示；设置为 True 时字符可以多行显示，即一行放不下时文本可自动换行。

④ ScrollBars 属性。

ScrollBars 属性用于设置滚动条是否显示。当 MultiLine 为 True 时，ScrollBars 属性才有效。属性值可以为 0、1、2 和 3。0 代表没有滚动条；1 代表水平滚动条；2 代表垂直滚动条；3 代表水平和垂直方向都有滚动条。

⑤ SelLength 属性。

SelLength 属性表示文本框中当前选中的字符个数，只可在运行时使用。

⑥ SelStart 属性。

SelStart 属性表示文本框中当前选定字符的起始位置。如果从第一个字符开始选中文本，SelStart 的值应设置为 0。这一属性在运行时可用。

⑦ SelText 属性。

SelText 属性表示文本框中当前选中的文本内容，运行时使用。

⑧ Locked 属性。

Locked 属性值为 True 时表示文本框内容不可编辑。

⑨ PasswordChar 属性。

PasswordChar 属性用于指定用于显示密码的字符。当此属性设置后，文本框中将显示该字符，而不是用户输入的字符，输入的实际值可以通过 Text 属性返回。例如，PasswordChar 设为 “*”，则输入的字符全部显示为 “*”。

(2) 常用方法

SetFocus 方法把输入光标（焦点）移到指定的文本框中。例如：

```
Text1.SetFocus
```

表示把焦点放到文本框 Text1 中。

【例 2-2】 SelStart 等属性练习界面，如图 2-5 所示。



图 2-5 SelStart 等属性练习

程序实现的功能是用户可以在 Text2 文本框中输入一个起始位置，在 Text3 文本框中输入要选择的字符个数。当用户点击“按属性选择”按钮时，程序会根据用户在 Text2 和 Text3 中输入的信息，在 Text1 文本框中选中相应的内容，并将这段选中的文本显示在 Text4 文本框中。另外，在文本框中选择文本后，单击“显示属性值”按钮，在 Text2、Text3、Text4 中分别显示选择文本的起始位置、长度和选中的内容。窗体及各控件的属性设置如表 2-1 所示，其中将 Text1 的 HideSelection 属性设置为 False，目的是使 Text1 失去焦点时选择文本仍然能加亮显示。

表 2-1 窗体及各控件属性

对象	属性名	属性值	对象	属性名	属性值
Form1	Caption	SelStart 等属性的练习	Text4	Locked	True
Text1	Text	欢迎使用 VB6.0	Label1	Caption	SelStart
	HideSelection	False	Label2	Caption	SelLength
Text2	Text	空	Label3	Caption	SelText
Text3	Text	空	Command1	Caption	按属性选择
Text4	Text	空	Command2	Caption	显示属性值

程序代码:

```
Private Sub Command1_Click ( )  
    Text1.SetFocus  
    Text1.SelStart=Text2.Text  
    Text1.SelLength=Text3.Text  
    Text4.Text=Text1.SelText  
End Sub  
  
Private Sub Command2_Click ( )  
    Text2.Text=Text1.SelStart  
    Text3.Text=Text1.SelLength  
    Text4.Text=Text1.SelText  
End Sub
```

(3) 常用事件

① Change。

当文本框的 Text 属性发生变化时触发。

② GetFocus。

当焦点进入文本框时触发。

③ LostFocus。

当焦点离开文本框时触发。

④ KeyPress。

在键盘上按下并释放一个 ANSI 键时会触发 KeyPress 事件，该事件会返回所按键对应的 ASCII 码值。

【例 2-3】在窗体上有 3 个文本框，要求在第 1 个文本框内输入一段英文，在输入的过程中同时在文本框 2 中显示对应的大写字母，在文本框 3 中显示对应的小写字母，运行效果如图 2-6 所示。

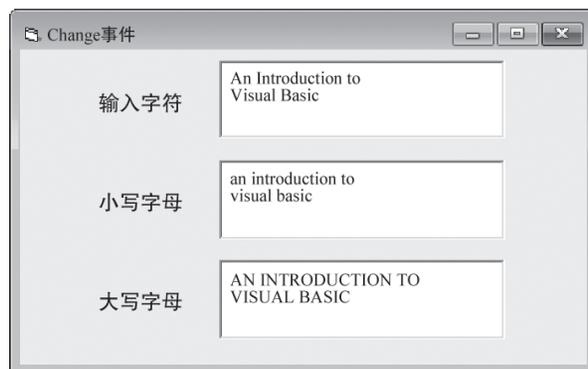


图 2-6 Change 事件

分析：窗体及各控件的属性设置如表 2-2 所示。为了实现在文本框 1 中输入字符时即时进行大小写字母转换的要求，所以决定利用文本框 1 的 Change 事件，因为每输入一个字符，文本框 1 的 Text 属性都发生了变化，就会激发 Change 事件。Ucase 将小写字母转换为大写字母，Lcase 将大写字母转换为小写字母，例如，“Ucase (" abceFS ")”的结果就是 ABCEFS。

表 2-2 窗体及各控件属性

对象	属性名	属性值	对象	属性名	属性值
Form1	Caption	Change 事件	Label1	Caption	输入字符
Text1	Text	空	Label2	Caption	小写字母
Text2	MultiLine	True	Label3	Caption	大写字母
Text3					

程序代码：

```
Private Sub Text1_Change ( )
    Text2.Text= LCase ( Text1.Text )
    Text3.Text= UCase ( Text1.Text )
End Sub
```

【例 2-4】编写一个字符和 ASCII 值相互转换的程序，如图 2-7 所示。



图 2-7 字符和 ASCII 码值相互转换界面

分析：在 Text1 中输入一个字符后，系统会自动将该字符的 ASCII 码值显示在 Text2 中。在 Text2 中输入一个介于 32 到 126 之间的数字（这个范围内的数字对应着键盘上可打印字符的 ASCII 码值），然后按下回车键，Text1 中将会显示出与该 ASCII 码值相对应的字符。窗体及各控件的属性设置如表 2-3 所示。

表 2-3 窗体及各控件属性

对象	属性名	属性值	对象	属性名	属性值
Form1	Caption	字符和 ASCII 码值转换	Text2	Text	空
Text1	Text	空	Label1	Caption	字符
Text1	MaxLength	1	Label2	Caption	ASCII 码值 (32—126)

程序代码：

```
Private Sub Text1_KeyPress ( KeyAscii As Integer )
    Text2.Text=KeyAscii
End Sub

Private Sub Text2_KeyPress ( KeyAscii As Integer )
    If KeyAscii= 13 Then ' 按下回车键
        Text1.Text=Chr ( Text2.Text )
    End If
End Sub
```

【例 2-5】编写一个用户登录的程序，如图 2-8 和图 2-9 所示。



图 2-8 登录界面



图 2-9 欢迎使用界面

分析：该程序需要两个窗体。要添加第二个窗体，可以单击菜单栏中的“工程”选项，然后选择“添加窗体”。在弹出的“添加窗体”对话框中，选择“窗体”选项，并点击“打开”按钮即可。Form1 和 Form2 的窗体及各控件的属性设置如表 2-4 和表 2-5 所示。其中将 Text1 的 TabIndex 属性设置为 0，目的是使窗体加载后，Text1 首先获得焦点。Form2 中使用 2 个标签实现字体阴影的效果。

表 2-4 Form1 及各控件属性

对象	属性名	属性值	对象	属性名	属性值
Form1	Caption	登录	Label1	Caption	用户名
Text1、Text2	Text	空	Label2	Caption	密码
Text2	MaxLength	6	Command1	Caption	确定
Text2	PasswordChar	*	Command2	Caption	取消

表 2-5 Form2 及各控件属性

对象	属性名	属性值	对象	属性名	属性值
Form2	Caption	欢迎	Label2	Caption	欢迎使用程序
Label1	Caption	欢迎使用程序		Left、Top	770、630
	Left、Top	720、600		Font	隶书、粗体、一号
	Font	隶书、粗体、一号		ForeColor	&H0000FF00&
	ForeColor	&H80000012&			

程序代码为：

```
Private Sub Command1_Click ( )
    ' 假设用户名为 admin，密码为 123456
    If Text1.Text="admin" And Text2.Text="123456" Then
        Unload Form1 ' 卸载 form1
        Form2.Show ' 显示 form2
    Else
        MsgBox" 用户名或密码有误，请重新输入！ "
        Text1 .SetFocus
    End If
End Sub

Private Sub Command2_Click ( )
    End
End Sub

Private Sub Text2_LostFocus ( )
    If IsNumeric ( Text2.Text )=False Then ' Text2 中存在非数字字符
        MsgBox" 密码必须为数字！ "
        Text2.SetFocus
    End If
End Sub
```

2.3.3 命令按钮

(1) 常用属性

① Caption 属性。

Caption 属性用于设置命令按钮的标题，在标题中，通过加“&+字母”，可以设置快捷键为“Alt+

该字母”。例如将 Command1 的 Caption 属性设为 “&Open” 后，标题上显示的效果是 Open，运行时按下 “Alt 键 +O”，相当于单击该命令按钮。

② Style 属性。

Style 属性定义了按钮的样式。当默认值为 0 时仅显示文字；当设置值为 1 时，才会显示图片。

③ Picture 属性。

Picture 属性用于设置按钮上显示的图片，但该属性仅在 Style 属性设置为 1 时生效。

④ Cancel 属性。

若 Cancel 属性被设置为 True，则按下键盘上的 Esc 键将等同于单击该按钮。

⑤ Default 属性。

Default 属性设置为 True 时，当窗体中的所有按钮都不具有焦点时，按下键盘上的 Enter 键相当于单击该按钮。

⑥ ToolTipText 属性。

ToolTipText 属性是一种工具提示属性，它可以与 Picture 属性同时使用。当鼠标指针悬停在按钮上时，会显示该属性指定的提示文本。

(2) 事件

命令按钮最常用的是单击事件 Click。



操作演练

编写一个命令按钮的属性和事件练习的程序，如图 2-10 所示。

该程序的功能包括通过单击“显示”按钮来显示文本框；单击“隐藏”按钮来隐藏文本框，以及单击图标按钮来退出应用程序。窗体及各控件的属性设置如表 2-6 所示。

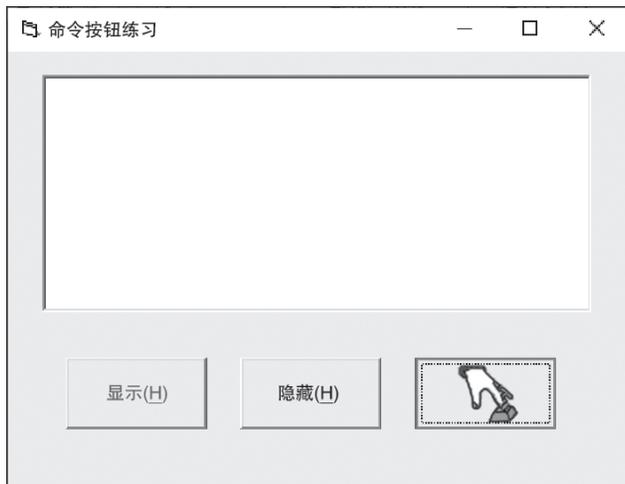


图 2-10 命令按钮练习程序