

VISUAL BASIC 编程 习题与解答

VISUAL BASIC

最佳的复习资料，实用的辅助教材

与国外高校计算机水平保持同步

为考研和出国深造奠定坚实基础

Byron S. Gottfried 著
向隼 周旭 等译

全美经典
学习指导系列

VISUAL BASIC 编程 习题与解答

VISUAL BASIC

Byron S. Gottfried 著

向昶 周旭 等译



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

本书以面向对象程序设计语言为基础,以Visual Basic语言开发为主线,结合大量实例,强调实用性,以通俗易懂的语言,将面向对象程序设计的基础知识和Visual Basic的具体应用展示给读者。本书配有大量的例题和习题,内容由浅入深,循序渐进,是广大计算机爱好者和软件开发人员的良师益友。

Byron S. Gottfried: Visual Basic.

Copyright © 2001 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved.
No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press & CITIC Publishing House.

本书中文简体字翻译版由机械工业出版社、中信出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签,无标签者不得销售
版权所有,侵权必究。

本书版权登记号:图字:01-2002-0358

图书在版编目(CIP)数据

VISUAL BASIC编程习题与解答/(美)戈特弗里德(Gottfried, B. S.)著;向琼等译. -北京:机械工业出版社,2002.8

(全美经典学习指导系列)

书名原文:Visual Basic

ISBN 7-111-10819-1

I. V… II. ①戈… ②B… ③向… III. BASIC语言-程序设计-解题 IV. TP312-44

中国版本图书馆CIP数据核字(2002)第062695号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码100037)

责任编辑:华章

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2002年8月第1版第1次印刷

787mm×1092mm 1/16·22.25印张

印数:0 001-5 000册

定价:29.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

作 者 序

Visual Basic 是一种事件驱动的程序设计语言，它所创建的应用程序可以在微软的 Windows 操作系统下运行。Visual Basic 吸引了两种截然不同的人群：一种是初学程序设计的学生，他们喜欢使用 Visual Basic 立即创建虽然简单、但看上去却很专业的程序，这些程序中还包括图形、对话框和下拉菜单；另一种是富有经验的程序员，他们只需做很少的工作就可以创建复杂的、基于窗口的应用程序。大部分 Visual Basic 的教科书都面向这两种人群。

本书适用于初学者，其风格与 Schaum 系列丛书中其他书的风格一致。它以一种易于被高级、中级或初级的学生所理解的方式来撰写。因此，本书既可以在介绍性的程序设计课里用做教科书，也可以作为编程课程的补充课本或作为有效的自学指南使用。大部分内容所需要的数学知识都没有超过高中代数的范围。

通过阅读本书使读者能尽可能快地写出很完整的初级 Visual Basic 程序。有一点非常重要，读者在阅读本书时，必须同步地写出这样的程序并且执行该程序。这样做可以极大地增强初学者的自信心并且提高他们的学习兴趣（学习在计算机上编程，就像学习演奏钢琴，决不能只学习书本知识）。

书中包含了许多例子。既有简单的，侧重于具体程序结构的举例，也有综合性的编程问题。另外，在每一章的后面都给出了一些复习题、习题和编程习题。复习题既能够帮助读者测试他们对本章内容的掌握情况，也是对相应章节内容有效的总结。

大部分的习题和编程习题都不需要特别的数学背景。学生们可以尽可能多地解决这些问题（本书在后面提供了习题的答案）。当将本书作为编程课的教科书时，教师可以补充额外的习题，以适应特殊的训练要求。

自从首次推出 Visual Basic 以来，它已经被修改了好几次。这本书以 Visual Basic 6 为基础，并且与早期的版本兼容。然而，微软关于下一版本（将被命名为 Visual Basic.NET）Visual Basic 的初步描述指出了与以前的版本有许多不兼容的地方。更多关于这些不兼容的综述以微软的初步描述为基础，作为附录附在本书后。

最后有一点很重要：读完本书后，读者可以学到关于事件驱动编程的一般概念以及 Visual Basic 的特殊规则。希望大部分的读者能够相信：用 Visual Basic 编程不但容易，而且有趣。

目 录

第 1 章 Visual Basic 简介	(1)
1.1 什么是 Visual Basic	(1)
1.2 事件和事件过程	(1)
1.3 与对象相关的概念	(1)
1.4 Visual Basic 程序的开发过程	(2)
1.5 必备的计算机技能	(3)
1.6 合理的程序组织	(3)
1.7 Visual Basic 程序的组成部分	(4)
1.8 Visual Basic 环境	(4)
1.9 打开一个已有的 Visual Basic 工程	(8)
1.10 保存 Visual Basic 工程	(9)
1.11 运行 Visual Basic 工程	(9)
1.12 如何获取帮助	(9)
1.13 Visual Basic 工程举例	(10)
复习题	(17)
编程题	(18)
第 2 章 Visual Basic 基本知识	(19)
2.1 数值常量	(19)
2.2 字符串常量	(20)
2.3 变量	(20)
2.4 数据类型及声明	(21)
2.5 运算符和表达式	(24)
2.6 运算符的优先级	(25)
2.7 插入括号	(26)
2.8 关于算术表达式的特殊规则	(26)
2.9 字符串表达式	(27)
2.10 变量赋值	(28)
2.11 显示输出——Print 语句	(29)
2.12 库函数	(30)
2.13 程序注释	(33)

复习题	(34)
习题	(35)
第 3 章 分支和循环	(41)
3.1 关系运算符和逻辑表达式	(41)
3.2 逻辑运算符	(42)
3.3 用 If-Then 程序块实现分支	(44)
3.4 用 If-Then-Else 程序块实现分支	(45)
3.5 选择语句: Select Case	(48)
3.6 For-Next 循环结构	(51)
3.7 Do-Loop 循环结构	(53)
3.8 While-Wend 循环结构	(56)
3.9 Stop 语句	(56)
复习题	(57)
习题	(58)
第 4 章 Visual Basic 控件基础知识	(63)
4.1 Visual Basic 控件工具	(63)
4.2 控件工具分类	(66)
4.3 使用控件	(66)
4.4 给窗体和控件命名	(67)
4.5 给窗体和控件的属性赋值	(67)
4.6 执行命令 (事件过程和命令按钮)	(68)
4.7 显示输出数据 (标签和文本框)	(70)
4.8 输入数据 (文本框)	(74)
4.9 选择多个选项 (复选框)	(80)
4.10 选择单个选项 (单选按钮和框架)	(84)
4.11 从列表中选择 (列表框和组合框)	(92)
4.12 整体属性赋值 (使用 With 程序块)	(96)
4.13 产生出错消息 (MsgBox 函数)	(98)
4.14 创建定时事件 (定时器控件)	(102)
4.15 滚动条	(105)
复习题	(109)
编程题	(110)
第 5 章 菜单和对话框	(117)
5.1 创建下拉菜单	(117)
5.2 通过键盘访问菜单	(121)
5.3 菜单的增强功能	(125)

5.4 子菜单	(129)
5.5 弹出式菜单	(131)
5.6 对话框	(135)
5.7 关于 MsgBox 函数的更多内容	(144)
5.8 InputBox 函数	(146)
复习题	(150)
编程题	(151)
第 6 章 调试及运行新的工程	(155)
6.1 语法错误	(155)
6.2 逻辑错误	(156)
6.3 设置断点	(158)
6.4 定义监视值	(158)
6.5 逐句执行程序	(162)
6.6 用户导致的错误	(168)
6.7 错误处理	(168)
6.8 产生一个独立的可执行程序	(174)
复习题	(175)
编程题	(177)
第 7 章 过程	(181)
7.1 模块和过程	(181)
7.2 子过程 (子程序)	(182)
7.3 事件过程	(189)
7.4 函数过程	(190)
7.5 作用域	(197)
7.6 可选参数	(210)
复习题	(212)
习题	(213)
编程题	(216)
第 8 章 数组	(219)
8.1 数组性质	(219)
8.2 数组声明	(220)
8.3 处理数组元素 (下标变量)	(222)
8.4 将数组传递给过程	(226)
8.5 动态数组	(237)
8.6 与数组相关的函数	(241)
8.7 控件数组	(242)

8.8 用 For Each-Next 循环	(253)
复习题.....	(254)
习题.....	(256)
第 9 章 数据文件	(265)
9.1 数据文件的性质	(265)
9.2 在 Visual Basic 中访问和保存文件：通用对话框控件.....	(265)
9.3 处理数据文件	(268)
9.4 顺序文件（文本文件）	(270)
9.5 随机存取（直接）数据文件	(293)
9.6 二进制文件	(311)
复习题.....	(312)
习题.....	(314)
编程题.....	(314)
附录 A ASCII 码字符集	(319)
附录 B 与 Visual Basic. NET 的不兼容之处	(321)
部分习题答案	(323)

第 1 章 Visual Basic 简介

1.1 什么是 Visual Basic

Visual Basic 是面向对象的程序开发系统，在 Visual Basic 中创建的应用程序可以在任何 Microsoft Windows 环境下运行。Visual Basic 具有以下两个主要的组件：

1. Visual Basic 提供了一个范围广泛、预先编制好的工具集，称为控件。这些控件可以在图形化的编程环境中作为图标被调用，以创建定制的窗口组件（例如：菜单、对话框、文本框、滚动条等）。
2. Visual Basic 提供了一个完整的程序命令集，这个命令集来源于 Microsoft 的经典 Basic 编程语言的实现。这个命令集包含的特性与流行的编程惯例一致。

Visual Basic 编程的一般方法包括两个方面：

1. 创建一个用户界面，这个用户界面要适合于具体的应用程序。
2. 加入一组 Basic 指令，这组指令可以执行与每个控件相关联的动作。

1.2 事件和事件过程

在传统的计算机程序中，程序的动作按照规定的顺序来执行。程序执行的顺序是连续的，可以与指令书写的顺序有关，也可以依赖于一个或多个逻辑测试的输出。还有一种情况是程序执行的顺序由内部结构所决定。

Visual Basic 是基于事件驱动的。在 Visual Basic 中，只有当用户在用户界面上使用了相应的对象（图标、复选框、单选按钮、菜单选项等）时，程序的功能才被激活。程序对用户操作行为的响应称为事件。请注意，虽然是用户启动了事件，但实际上是程序的响应定义了事件，产生响应的 Basic 命令组称为事件过程。

1.3 与对象相关的概念

除了事件和事件过程，还需要理解下面这些术语的含义：

窗体：在 Visual Basic 中，窗口称为窗体。每一个窗体都包含一个标题栏。窗体中通常也包含菜单栏、状态栏、一个或多个工具栏、滚动条等。用户区（称为客户区）占据了窗体的剩余部分。有些应用程序基于单个窗体，不过更多的应用程序可能

需要两个或者多个窗体。图 1-1 显示了一个简单的窗体，这个窗体包括了两个命令按钮控件和一个文本框控件。

控件：与用户进行交互的图标称为控件。通常使用的控件包括命令按钮、单选按钮、复选框、标签、文本框、图片框和菜单。典型的情况是用户激活控件（例如，单击命令按钮）来产生事件。

对象：窗体和控件都可以归为对象。大多数的对象都与事件相关联。因此，对象可以包含各自的独特事件过程。通常，对象还与它们自身的属性和方法相关联。

属性：对象包含属性。一般而言，属性定义了对对象的外观或行为。属性的选择依赖于对象的类型。举例来说，就一个命令按钮而言，name（名称）、caption（标题）、height（高度）、width（宽度）、background color（背景颜色）、location（位置）和 font（字体）是命令按钮最常见的属性。

方法：有一些对象还包含特殊的程序语句，这些程序语句称为方法。方法给出了一些预先定义好的行为，这些行为可以影响相关的对象。

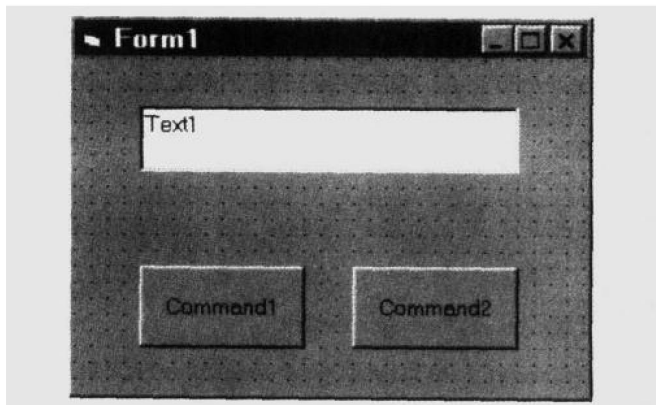


图 1-1 一个包含三个控件的窗体

1.4 Visual Basic 程序的开发过程

一般而言，Visual Basic 程序的编写由以下几个步骤组成：

1. 尽可能明确地确定程序的用途和目标（不过请注意：在完成程序之前，用户可能会改变主意，也许还会改变几次）。
2. 使用 Visual Basic 的程序开发工具创建用户界面。这个步骤通常包括两个相关的动作：
 - (a) 在各个窗体中添加所有的控件。
 - (b) 定义每一个控件的属性。
3. 写出 Visual Basic 指令，以便响应不同的程序事件。一般而言，需要为每一个控件写一些命令组，这些命令组称为事件过程 [但是有一些控件（例如标签控件）没有与之相关联的事件过程]。

4. 运行程序, 检测它是否能够正确执行。
5. 如果结果不正确或者程序不能正确地做出你所期望的响应, 那么就请重复以上的一个或多个步骤, 直到正确为止。

在得到满意的结果之前, 也许需要多次执行上述步骤。请记住: 计算机编程是一个细致而又充满创造性的过程, 在这个过程中, 需要耐心、技巧和独创性。很多时候, 程序开发的过程会令人有挫折感 (例如, 当程序不能正确地执行或者由于一些隐藏的、难于发现的编程错误使程序根本不能执行时)。这时, 通常需要放松一下, 暂时将工作放到一边, 过一段时间之后再重新开始, 你会感觉好得多。

1.5 必备的计算机技能

要使用 Visual Basic 并从这本书中有所获益, 应该熟悉下面的内容:

1. 熟悉 Microsoft Windows 操作系统 (如: Windows 2000/98/95/NT 等), 特别应该熟悉以下操作:
 - (a) 进入 Windows。
 - (b) 使用鼠标。
 - (c) 访问应用程序 (特别是 Visual Basic)。
 - (d) 退出 Windows。
 - (e) 获得联机帮助。
2. 在 Windows 中操作文件的能力 (定位文件、打开文件、编辑文件、保存文件、拷贝文件、移动文件、删除文件等)。
3. 安装新的应用程序 (在 Visual Basic 没有安装好或者需要重新安装的情况下)。

本书将不再讨论这些问题——假定读者已经掌握了这些必不可少的技巧。不过, 在本书后面需要的地方将讨论 Visual Basic 中的文件操作。

1.6 合理的程序组织

实际上, 所有重要的计算机程序都包含以下三个主要任务:

1. 输入数据 (提供将要被处理的信息)。
2. 计算得到的结果 (处理输入的数据)。
3. 输出结果 (输出计算的结果)。

每一步都有可能是很复杂的, 因此完成它们需要相当多的时间和努力。

在 Visual Basic 中, 第一步和最后一步 (数据的输入和数据的输出) 是通过用户界面来完成的, 因此设计用户界面非常重要。对于具体的应用程序, 用户界面应该用合乎逻辑的、直截了当的方式来接收数据和处理输出。在许多应用程序中, 用户界面设计是整个程序开发过程中最复杂的一个部分, 不过利用 Visual Basic 中已有的控件可以极大地简化这个过程。

第二步(计算)通常由一系列 Visual Basic 指令来执行,这些指令嵌入在一个或多个独立的事件过程中。这些指令的选择和顺序的安排由恰当的算法来决定。也就是说由一个合乎逻辑的、有序的计算步骤来决定,这个计算步骤可以将给出的输入数据转换成期望的输出数据。在许多实际的应用程序中,这个步骤(也就是算法的实现)可能很复杂。即使是对于非常熟练的程序员来说,仍然是一种能力的挑战。

1.7 Visual Basic 程序的组成部分

在 Visual Basic 中,程序被称为工程(project)。每一个 Visual Basic 工程至少由两个独立的文件——一个工程文件(该文件的扩展名为 .vbp)和一个窗体文件(该文件的扩展名为 .frm)组成。有些工程文件包含了多个窗体文件和其他类型的文件,例如类模块(.cls)文件、标准模块(.bas)文件、资源(.res)文件以及 Active 控件(.ocx)文件。因此,Visual Basic 工程文件的开发包括开发几个不同的文件,并且需要的时候可以在 Visual Basic 环境中分别访问和使用这些文件。

1.8 Visual Basic 环境

要进入 Visual Basic 编程环境,请用鼠标双击 Visual Basic 图标,这个图标通常位于 Windows 的桌面上。双击后,可以看到如图 1-2 所示的一组打开的窗口。接下来,可以通过

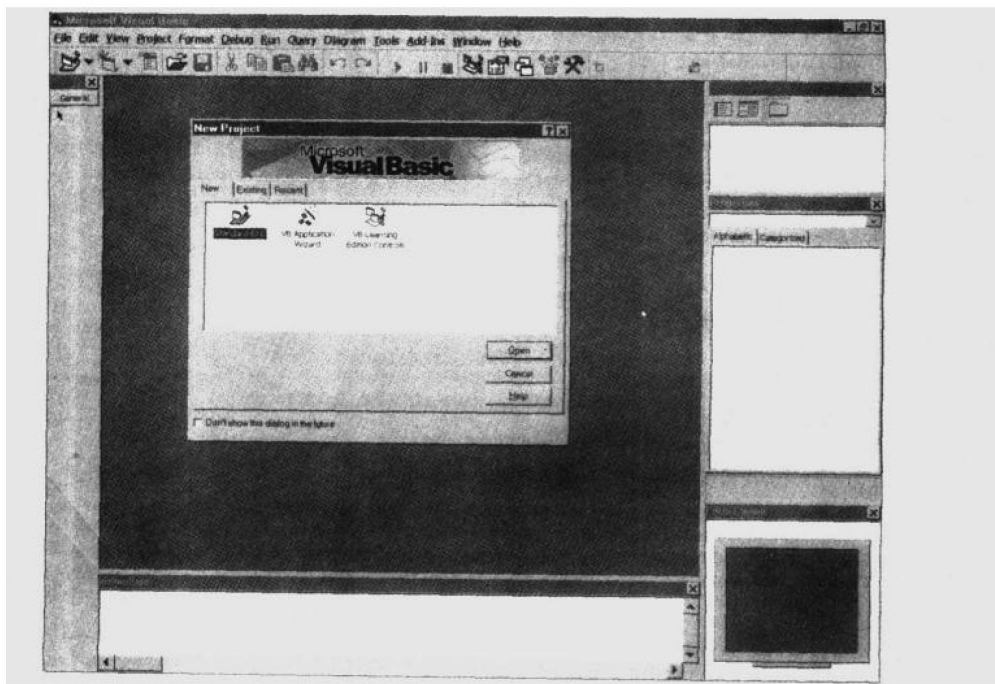


图 1-2 Visual Basic 打开的窗口组

选择 New/Standard.EXE 进入一个新工程的工作区。或者通过选择 File 菜单中的 New Project 项来开始一个新的工程。也可以通过选择 Existing 标签或 Recent 标签, 然后选择感兴趣的工程。还有一种方法是选择 File 菜单中的 Open Project 项, 然后选择感兴趣的工程。

如果选择创建一个新的工程, 将看到类似于图 1-3 所示的一组窗口。主要的组成部分如下:

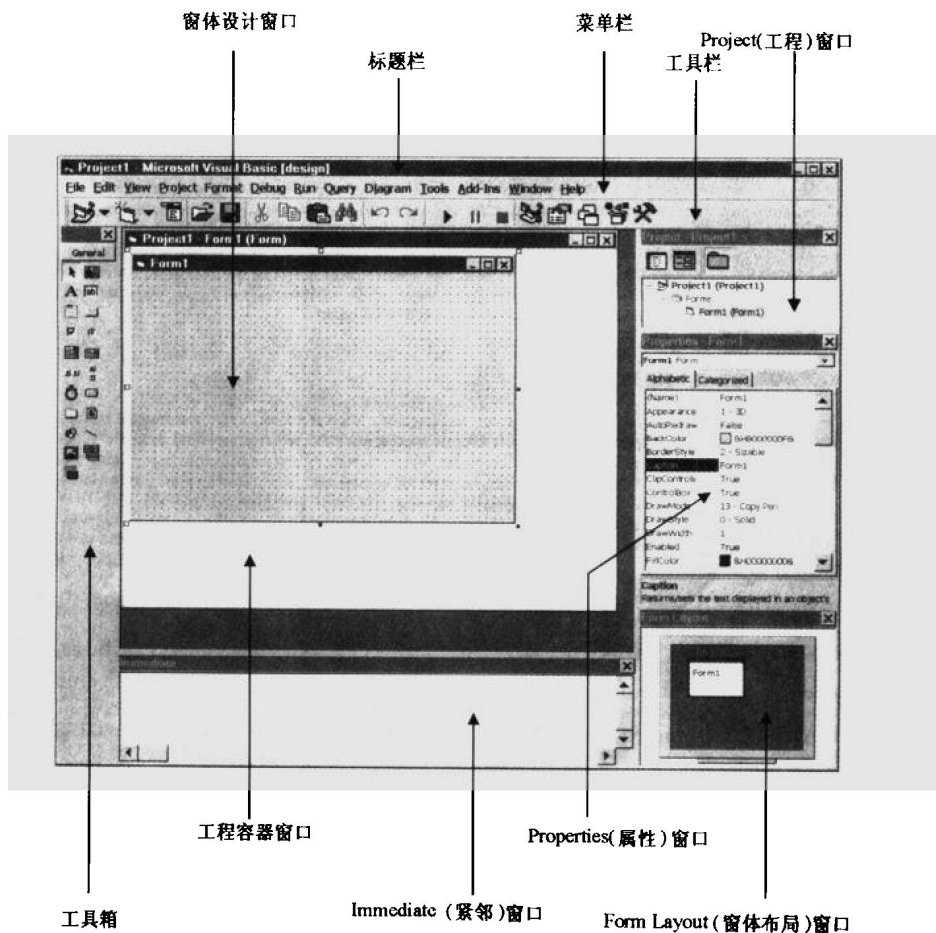


图 1-3 Visual Basic 中新工程的工作界面

标题栏

处于窗口最顶端的是标题栏。标题栏包括了工程的名字, 位于左端的代表 Visual Basic 的图标和位于右端的最小化按钮、还原按钮和关闭按钮。我们将在后面需要的地方讨论这些图标。现在, 可以单击处于左端的图标, 然后从下拉菜单中选择 Close 退出 Visual Basic, 或

者单击最右端的关闭按钮 (×) 退出。

菜单栏

第二行称为菜单栏。选择它们中的任意一项 (File, Edit, View, Project, Format, ... Help) 就会出现 Visual Basic 的某个下拉菜单。这些菜单展示了 Visual Basic 中各个功能的逻辑分组。例如, File 菜单包括了以下一些选项: 打开新的或已有的工程、保存工程、保存当前活动窗口、打印当前活动窗口、运行 (执行) Visual Basic 工程、打开最近访问的工程等等。

工具栏

第三行称为标准工具栏。该行中的一些图标提供了与几个最常使用的菜单选项相同的功能。也可以通过菜单栏的下拉菜单来使用这几个菜单选项。例如: 标准工具栏包括打开已有工程, 保存当前工程, 剪切、复制和删除, 撤销最近的工作, 开始、暂停和结束程序的执行以及从当前环境中增加/删除窗口的图标。所有这些功能都可以通过使用下拉菜单来执行。因此, 虽然工具栏中的图标并不能够提供新的、特别的功能, 但是使用很方便, 因为可以通过简单的单击鼠标来完成相应功能。

要使用其他的工具栏 (Debug、Edit 和 Form Editor), 可以从 View 菜单中选择 Toolbars 菜单项来实现。还可以通过访问 View 菜单的 Toolbars 菜单项来关闭标准工具栏。

窗体设计窗口和工程容器窗口

窗体设计窗口用于设计用户界面。要设计用户界面, 可以从工具箱中选择想要的控件图标, 然后将它们放置在窗体设计窗口中。每一个控件都可以移动 (移动控件的方法: 将鼠标放在图标上, 然后拖动图标, 就可以移动控件), 每一个控件的大小都可以改变 (方法: 激活控件的图标, 然后拖动图标周围的某一手柄), 并且可以根据需要, 对控件的属性重新赋值。

整个窗体 (也就是整个窗体设计窗口) 可以在工程容器窗口内移动 (方法: 将鼠标放置在窗体上, 然后拖动, 就可以移动窗体)。也可以改变窗体的大小 (方法: 激活窗体, 然后拖动窗体周围的某一手柄)。所有的窗口都可以通过单击窗口标题栏中相应的按钮来关闭窗口、最小化窗口和改变窗口大小。

工具箱

工具箱中的图标代表常用的控件, 例如标签、文本框、命令按钮、图片框、框架、复选框、单选按钮、文件列表框等等。可以从工具栏中选择一个控件, 然后双击该控件的图标, 将它放置在当前窗体设计窗口中; 或者单击该控件图标, 然后在窗体设计窗口中想要放置控件的位置上单击一下, 还可以拖动鼠标改变控件的大小。

一旦已经在窗体设计窗口中放置好了控件, 就可以双击该控件, 以便浏览或者编辑与控

件相对应的代码（也就是相关的 Visual Basic 指令）。

属性窗口

我们已经知道，每一个对象都有与之相关联的属性，每一个对象都有独特的属性列表。窗口允许你对与某个具体的对象（比如特定的窗体或者控件）相关联的属性赋值或改变属性值。要改变对象的属性，先单击控件来激活它；然后从属性窗口的左边一栏中选择相应的属性。一旦你选择了某个属性，与这个属性相邻的右边的栏就会改变其外观，显示出一个下拉菜单供用户从中选择合法的属性值。

工程窗口

工程窗口显示了一个分层的文件列表，这个文件列表与确定的工程相联系。这些文件代表了各自的窗体和程序模块，可以通过双击工程窗口中相应的图标来将一个窗体和程序模块放到工程容器中。同时，也可以通过单击工程窗口顶部最左边的两个图标之一来选择 Object View 或者 Code View。

代码编辑窗口

如果选择的是工程窗口中的 Code View，或者如果双击窗体设计窗口中的控件图标，那么将打开代码编辑窗口，显示出与当前活动窗体相对应的 Visual Basic 代码。图 1-4 显示了包含两个不同事件过程的代码编辑窗口（例如：与两个不同的命令按钮相对应的 Visual Basic 代码）。现在请暂时忽略这些 Visual Basic 代码。请注意，窗口顶部有两个列表框。左边的列表框（在图 1-4 中显示出 Command2）是 Object Listbox（对象列表框），它使你可以选择与特定的对象相对应的事件过程（例如：一个特定的命令按钮、一个窗体、一个标签、一个文本框等）。右边的列表框（在图 1-4 中显示为 Click）称为 Procedure Listbox（过程列表框），对于当前的对象来说，它允许选择与特定的事件类型相联系的事件过程（例如：单击、下拉、按下键盘、按下鼠标等）。

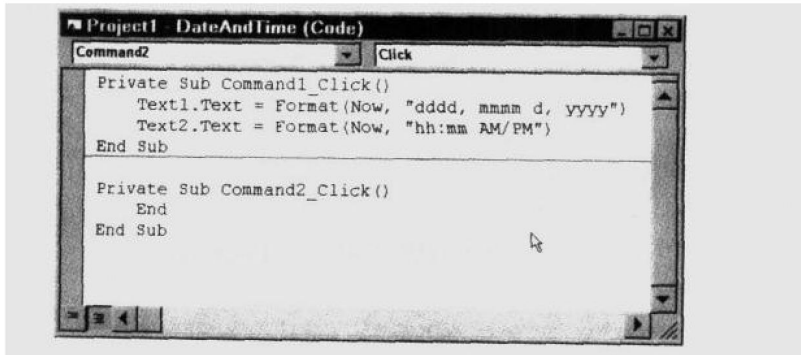


图 1-4 包含两个不同的事件过程的代码编辑窗口

窗体布局窗口

窗体布局窗口使你可以定义工程中窗体在屏幕上的位置。如果想要改变窗体的位置，只要简单地将窗体图标拖动到想要放置的位置就可以了。

紧邻窗口

当调试工程的时候，紧邻窗口是很有用的。在这个窗口中，无论输入哪个变量或者表达式，其对应的值都会立刻显示出来。

改变窗口位置、调整窗口大小、删除窗口、增加窗口和对接窗口

任何窗口都可以被重新定位（通过激活窗口，然后将它拖动到想要放置的位置），或者调整窗口大小（激活窗口，然后拖动边界或者转角）以及删除窗口（单击窗口右上角的关闭按钮）。可以从 View 菜单选择窗口的名字来激活一个窗口。

如果对几个窗口进行了移动和调整了大小，界面可能会变得有些凌乱。当出现这种情况的时候，Visual Basic 允许窗口恢复到预先分配的位置上。这个过程称为对接。对接窗口与其相邻的窗口并排排列，并且所有的窗口都是可见的。

要对接一个已从预先分配的位置上移动了的窗口，只需要简单地双击这个窗口的标题栏。对于任何窗口而言，可以通过选择 Tools 菜单中的 Option/Docking 项，然后选择或取消选定相应的复选框来对接窗口。

1.9 打开一个已有的 Visual Basic 工程

我们已经看到了如何访问一个已经存在的工程，在 New Project 窗口（如图 1-2 所示）的 Existing 或者 Recent 标签下，从工程的名字列表中选择工程，或者通过 File/Open Project 来定位一个工程。打开一个工程后，有可能窗体设计窗口并没有显示在工程容器窗口中。要使用窗体设计窗口，可以打开工程窗口中的 Forms 图标，如图 1-5 所示，然后选择想要打开的窗体。双击窗体的图标，就可以显示窗体设计窗口。

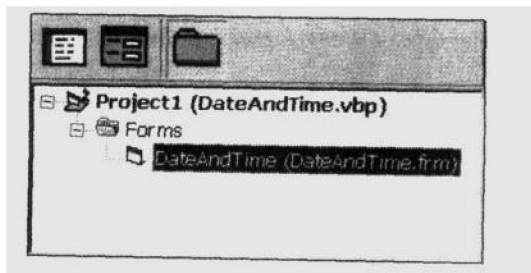


图 1-5 在工程容器窗口中激活一个窗体

1.10 保存 Visual Basic 工程

在 Visual Basic 中保存工程稍微有一点复杂，因为保存工程时要保存多个文件。

如果要首次保存一个新的 Visual Basic 工程，从 File 菜单选择 Save Project As 项。保存时，提示用户给出窗体的名字（.frm）和工程的名字（.vbp）。通常，可以给这些文件取相同的名字。

要保存一个已经保存但又经过更新的工程，在工具栏中单击 Save Project（保存工程）按钮（如图 1-6 所示），或者可以从 File 菜单中选择 Save Project 项。这样做将导致工程文件的当前版本以它们已经存在的名字被保存。

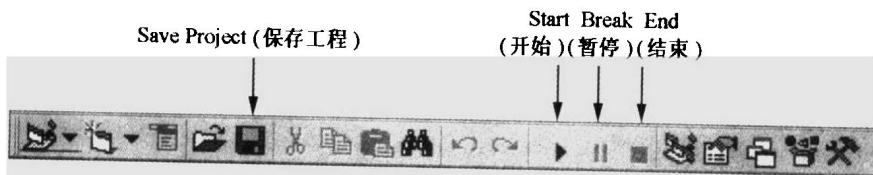


图 1-6 标准工具栏

要以不同的名字来保存先前已经保存过的工程（这是一个复杂的工作），必须用新的名字来分别保存每一个文件。因此，应该首先用新的名字来保存每一个窗体，并且用新的名字保存工程。如果只是简单地用不同的名字保存了工程文件，窗体文件还是保持原来的名字，那么在工程的新版本和旧版本之间进行转换时，将会出现一些问题。

1.11 运行 Visual Basic 工程

要运行一个 Visual Basic 工程，只需要单击工具栏上的 Start 按钮（参见图 1-6），或者从 Run 菜单中选择 Start 菜单项。可以单击 Break 按钮，或者选择 Run 菜单的 Break 菜单项来暂时停止程序的执行。要恢复一个暂停的程序可以通过单击 Run 按钮，或者从 Run 菜单中选择 Continue 项。如果要结束程序的执行，只需要简单地单击 End 按钮，或者从 Run 菜单中选择 End 项。

1.12 如何获取帮助

Visual Basic 包括许多复杂的概念、预先定义好的标志符、详细的语法要求等等。你不可能同时记住所有的内容。幸运的是，Visual Basic 提供了出色的联机帮助功能。联机帮助可以回答大部分的问题，并且可以提供详细的有关 Visual Basic 各种特性的信息，其中还包括一些示例。