



Visual Basic

程序设计教程上机实验指导

陈 歆 周淑秋 主编



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

Visual Basic 程序设计教程

上机实验指导

陈 歆 周淑秋 主编

上海交通大学出版社

内 容 提 要

本书是本社出版的《Visual Basic程序设计教程》的配套实验教材。编者结合多年教学经验精选出有代表性的实验素材,帮助读者边学边练,学以致用,迅速提高操作能力。全书主要内容包括程序设计基础、数据的输入输出、程序的控制结构、数组与过程、常用的内部控件、菜单界面设计、文件管理、键盘与鼠标事件过程、多重窗体设计编程等。

本书内容丰富,重点突出,文字叙述简明易懂,图文并茂,实验题目注重实际应用和可操作性。每一章节都配有相应的练习题,书末配有解题提示和参考,以帮助读者反复理解。本书适合作为高等院校计算机公共课教材,也可作为各类 Visual Basic 培训班及全国计算机等级考试读者的学习参考书。

图书在版编目(CIP)数据

Visual Basic 程序设计教程上机实验指导/陈歆,周淑秋
主编. —上海:上海交通大学出版社,2013

ISBN 978-7-313-09617-3

I. V... II. ①陈... ②周... 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(2013)第 081282 号

Visual Basic 程序设计教程上机实验指导

陈 歆 周淑秋 主编

上海交通大学出版社出版发行

(上海市番禺路 951 号 邮政编码 200030)

电话:64071208 出版人:韩建民

常熟市大宏印刷有限公司 印刷 全国新华书店经销

开本:787mm×1092mm 1/16 印张:13.5 字数:311 千字

2013 年 5 月第 1 版 2013 年 5 月第 1 次印刷

印数:1~2030

ISBN 978-7-313-09617-3/TP 定价:29.00 元

版权所有 侵权必究

告读者:如发现本书有印装质量问题请与印刷厂质量科联系
联系电话:0512-52621873

目 录

第 1 章 Visual Basic 6.0 概述	1
1.1 知识要点	1
1.2 实验	7
1.3 练习题	12
第 2 章 Visual Basic 程序设计基础	16
2.1 知识要点	16
2.2 实验	24
2.3 练习题	34
第 3 章 Visual Basic 数据的输入与输出	37
3.1 知识要点	37
3.2 实验	42
3.3 练习题	51
第 4 章 程序的控制结构	55
4.1 知识要点	55
4.2 实验	60
4.3 练习题	72
第 5 章 数组与过程	85
5.1 知识要点	85
5.2 实验	90
5.3 练习题	107
第 6 章 常用的内部控件	120
6.1 知识要点	120
6.2 实验	122
6.3 练习题	133
第 7 章 菜单界面设计	139
7.1 知识要点	139
7.2 实验	140
7.3 练习题	151

第 8 章 文件管理	154
8.1 知识要点	154
8.2 实验	158
8.3 练习题	164
第 9 章 键盘与鼠标事件过程	167
9.1 知识要点	167
9.2 实验	169
9.3 练习题	175
第 10 章 多重窗体程序设计	180
10.1 知识要点	180
10.2 实验	182
10.3 练习题	185
参考答案与提示	186

第1章 Visual Basic 6.0 概述

1.1 知识要点

1.1.1 一个简单 Visual Basic 应用程序的组成

通常, Visual Basic 的应用程序由三类模块组成, 即窗体模块、标准模块和类模块。

(1) 窗体模块 (.frm)。窗体模块由界面窗体和代码组成。一个应用程序包含一个或多个窗体模块 (其文件扩展名为 .frm)。每个窗体模块均分为两部分: 一部分是作为用户界面的窗体; 另一部分是执行具体操作的代码。

(2) 标准模块 (.bas)。标准模块完全由不与具体的窗体或控件相关的代码组成。标准模块中, 可以声明全局变量, 也可以定义函数过程或子程序过程。

(3) 类模块 (.cls)。类模块由代码和数据组成。每个类模块都定义了一个类, 可以在窗体模块中定义类的对象, 调用类模块中的过程。类模块可以视为没有物理表示的控件。

一个工程中的文件可分为六类:

(1) 窗体文件 (.frm), 每个窗体对应一个窗体文件, 窗体及其控件的属性, 以及其他信息和代码都存放在窗体文件中, 一个应用程序可以有多个窗体, 但最多为 255 个。

(2) 工程文件 (.vbp), 每个工程对应一个工程文件。

(3) 工程组文件 (.vbg), 当一个程序包括两个以上的工程时, 这些工程构成一个工程组。

(4) 标准模块文件 (.bas), 它是一个纯程序代码性质的文件, 不属于任何一个窗体, 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, 主要在大型应用程序中使用。

(5) 类模块文件 (.cls), Visual Basic 中大量的预定义的类以及用户通过类模块来定义自己的类, 都可以用一个文件来保存。

(6) 资源文件 (.res), 它是一个纯文本文件, 可以同时存放如文本、图片、声音等多种“资源”的文件, 资源文件由一系列独立的字符串、位图及声音文件组成, 可用简单的文字编辑器, 如 NotePad 编辑。

Visual Basic 采用事件驱动的编程机制, 因此, Visual Basic 应用程序的工作方式主要通过事件驱动来实现。

(1) 事件驱动。所谓事件驱动 (编程机制) 就是指触发 (激发) 对象的某个事件, 对象将对该事件的触发 (激发) 作出响应, 从而操作执行一段事先编写好的相应程序代码 (事件过程)。事件的触发可以通过用户的操作触发, 也可以通过操作系统 (计时器) 或其他应用程序的消息触发, 还可以由应用程序本身的消息触发。

(2) 事件驱动应用程序的典型操作顺序 (序列)。操作顺序为: 启动应用程序, 加载和

显示窗体的用户界面,对象(窗体、控件等)接收事件并执行相应的事件代码,执行完再等待下一次事件的触发。

1.1.2 用 Visual Basic 语言设计开发应用程序的步骤

用 Visual Basic 语言设计开发简单应用程序时,一般主要包括三大步骤,详细涉及十个具体步骤。

1) 三大步骤

- (1) 界面设计:建立可视化用户界面。
- (2) 属性设置:设置可视化用户界面的特性。
- (3) 代码编写:编写事件驱动代码。

2) 十个具体步骤

- (1) 启动 Visual Basic。
- (2) 新建(打开)工程(一个工程包含两部分内容:对象和代码)。
- (3) 用户界面设计。
- (4) 对象的属性设置。
- (5) 事件驱动的代码编写。
- (6) 调试、运行。
- (7) 保存(窗体文件——*.frm、标准模块文件——*.bas、工程文件——*.vbp 等)。
- (8) 编译生成可执行文件(*.exe)。
- (9) 退出 Visual Basic。
- (10) 在 Windows 环境下运行可执行文件(*.exe)。

必须指出,设计开发应用程序(包括大型程序)时,并非要完全按上述步骤的顺序进行,而且,上述步骤也并非全面。但上述步骤对于学习和掌握运用 Visual Basic 语言设计开发应用程序的过程是非常有效的。

1.1.3 窗体、标签、命令按钮、文本框的常用属性、方法和事件

1) 窗体

(1) 窗体的属性。

① 描述窗体外观的属性。

Caption 属性:表示窗体标题栏上的文本内容。取值:字符串。

BorderStyle 属性:只读属性,取值如下:

0——None,无边框;

1——FixedSingle,固定单边框且大小只能用最大化和最小化按钮改变;

2——Sizable,默认有双线边界的可改变大小的边框;

3——FixedDialog,按设计时的大小固定边框且没有最大化和最小化按钮;

4——FixedToolWindow,固定工具窗口、大小不能改变且只显示关闭按钮并用缩小的字体显示标题栏;

5——SizableToolWindow,可改变大小且只显示关闭按钮并用缩小的字体显示标题栏。

BackColor 属性: 窗体的背景色, 取值: 代表颜色的十六进制数或调色板或 vbRed 等。

ForeColor 属性: 窗体的正文或图形的前景色, 取值: 代表颜色的十六进制数或调色板或 vbRed 等。

Picture 属性: 窗体上显示的图片, 取值: 设计时属性窗口中加载或运行时用 LoadPicture() 函数装入。

② 描述窗体位置和大小的属性: 左边位置 (坐标) Left、上边位置 (坐标) Top 和高 Height、宽 Width。取值都是数值。

③ 描述窗体行为的属性: 是否可移动 Moveable、是否激活 (可用状态) Enabled、是否可见 Visible (运行时起作用)。取值都是 True (默认) 或 False。

④ 描述窗体字形 (字体、大小) 的属性: 包括字体名称 FontName、字体大小 FontSize。

⑤ 描述窗体的其他属性:

Name 属性。只读属性, 在代码中用来代表窗体。取值: 字符串。

ControlBox 属性。取值: True 或 False。

WindowState 属性。取值: 0——默认正常、1——最小化图标、2——最大化窗口。

MinButton 属性。取值: True 或 False。

MaxButton 属性。取值: True 或 False。

AutoRedraw 属性。主要用于多窗体程序设计中, 取值: True 或 False。

Icon 属性。窗体最小化时的图标, 取值: 设计时属性窗口中加载或运行时用 loadPicture 函数装入或通过另一窗体最小化时的图标属性来赋值。

(2) 窗体的事件。

Load 事件: 当窗体被装入内存时, VB 系统自动触发该事件。

Unload 事件: 当窗体被关闭后, 将触发该事件。

Click 事件: 在运行时, 当用户在窗体的空白区域上单击鼠标时触发该事件。

DbClick 事件: 在运行时, 当用户在窗体的空白区域上双击鼠标时触发该事件。

Activate 事件: 当本窗体变为活动窗口时触发该事件。

Deactivate 事件: 当另一窗体变为活动窗口前, 触发该事件。

Paint 事件: 当本窗体被移动或放大, 或窗体移动时覆盖了一个窗体时, 触发该事件。

Resize 事件: 在运行时, 当窗体的大小改变时, 触发该事件。

(3) 窗体的方法。

Hide 方法: 在运行时, 用来隐藏窗体。

Show 方法: 用于激活窗体, 使被激活的窗体成为当前活动窗体。

Cls 方法: 用于清除窗体上的所有图形和正文。

Print 方法: 用于在窗体上显示信息。

2) 标签

(1) 标签的属性。标签的部分属性与窗体及其他控件相同, 包括 FontBold, FontItalic, FontName, FontSize, FontUnderline, Height, Left, Name, Top, Visible, Width。对其他属性说明如下:

① **Alignment** 属性: 标签和文本框均有该属性, 该属性决定怎样放置标签的标题或文本框

的内容,规定文本的对齐方式,具体取值如下:

0——左对齐;

1——右对齐;

2——居中。

② **AutoSize** 属性: 仅标签具有该属性,该属性用于设置标签的大小。如果该属性设置为 **True**,则系统自动改变标签的大小,以适应由 **Caption** 属性制定的文本;如果该属性设置为 **False**,则标签保持设计时定义的大小,此时,如果标题太长,则系统会自动进行剪裁,以适应标签的大小。

③ **BorderStyle** 属性: 标签和文本框均有此属性,此属性用于设置边框类型。具体取值如下:

0——设置标签或文本框无边框(默认值);

1——设置标签或文本框为单线边框。

④ **Caption** 属性: 标签中的文本只能用 **Caption** 属性显示,**Caption** 属性用来显示标签的文本。

⑤ **Enabled** 属性: 标签和文本框都有该属性。该属性用于设置标签文本框是否接收各种鼠标事件。该属性一般设置为 **True**,可接收鼠标事件;但当该属性设置为 **False** 时,屏蔽各种鼠标事件,而且使标签或文本框对象变灰。

⑥ **BackStyle** 属性: 该属性可以取两个值,即 0 和 1。具体取值如下:

1——标签将覆盖背景(默认值);

0——则标签为“透明”的。

⑦ **WordWrap** 属性: 该属性用来决定标签的标题(**Caption**)属性的显示方式。该属性取两种值: 即 **True** 和 **False**。默认为 **False**。具体取值如下:

True——则标签将在垂直方向上变化大小,以适应标题文本,水平方向上的大小与原来所画的标签相同。

False——则标签将在水平方向上扩展到标题中最长的一行,在垂直方向上显示标题的所有各行。

注意,为使 **WordWrap** 起作用,应把 **AutoSize** 属性设置为 **True**。

(2) 标签的事件。与图片框、图像框一样,标签对象能接收 **Click**、**DblClick** 事件。此外,标签主要用来显示一小段文本,可以通过 **Caption** 属性定义,不需要使用其他方法。

(3) 标签的方法。标签对象的主要作用是显示一小段文本,且文本是由 **Caption** 属性设置的。与此有关的一些方法,对于一般用户来说,用处不大,因此这里不再介绍。

3) 文本框

(1) 文本框的属性。文本框的属性包括 **BorderStyle**, **Enabled**, **FontBold**, **FontItalic**, **FontSize**, **FontUnderline**, **Height**, **Left**, **Name**, **Top**, **Visible**, **Width**。此外,还有一些其他属性,归纳如下:

① **Text** 属性。**Text** 属性用于接收在文本框中输入的文本。程序读入此属性,用户可以查看自己输入的内容。该属性也可以由程序进行修改,以改变其中显示的文本。设计时使 **Text** 属性为空字符串时,则可使正文框空白。

② **MaxLength** 属性。该属性用于设置文本框中显示的字符数。当该属性值为 0（默认值）时，表示文本框可以接收任意多个输入字符；当该属性值设置为非 0 数值时，系统会将用户输入的字符限制在该数值的范围之内，即该非 0 值是最大输入字符数。

③ **MultiLine** 属性。该属性用于设置文本框是单行显示还是多行显示。这是一个布尔属性，具体规定如下：

True——允许多行（通过回车）；

False——禁止多行。

④ **ScrollBars** 属性。该属性用于设置滚动条。具体规定如下：

0——无；

1——水平；

2——垂直；

3——水平和垂直两种。

⑤ **PasswordChar** 属性。该属性可用于口令输入。在缺省状态下，该属性被设置为空字符串（不是空格）。用户从键盘上输入时，每个字符均可以在文本框中显示出来。如果把 **PasswordChar** 属性设置为一个字符，例如，星号（*），则在文本框中键入字符时，显示的不是键入的字符，而是所设置的字符。但文本框中的实际内容仍是输入的文本，只是显示结果被改变了。利用这一特性，可以设置口令。

⑥ **SelLength** 属性。该属性用来定义当前选中的字符数。当在文本框中选择文本时，该属性值会随着选择字符的多少而改变。也可以在程序代码中把该属性设置为一个整数值，由程序来改变选择。如果 **SelLength** 属性值为 0，则表示未选中任何字符。该属性以及下面的 **SelStart**、**SelText** 属性只有在运行期间才能设置。

⑦ **SelStart** 属性。该属性用来定义当前选择的文本的起始位置。0 表示选择的开始位置在第一个字符之前，1 表示从第二个字符之前开始选择，依此类推。该属性也可以通过程序改变。

⑧ **SelText** 属性。该属性含有当前所选择的文本字符串，如果没有选择文本，则该属性值是一个空字符串。如果在程序中设置该属性，则用该值代替文本框中选中的文本。

⑨ **Locked** 属性。该属性用来指定文本框是否可以被编辑。当设置值为 False（默认值）时，可编辑文本框中的文本；当设置值为 True 时，可以滚动和选择控件中的文本，但不能编辑。

（2）文本框的事件。文本框支持 Click、DbClick 等鼠标事件，同时也支持 Change、GotFocus、LostFocus 等事件。

① **Change** 事件。当用户向文本框中输入新信息，或当程序把 **Text** 属性设置为新值，从而改变文本框 **Text** 属性时，将触发 **Change** 事件。程序运行后，在文本框中每键入一个字符，就会引发一次 **Change** 事件。

② **GotFocus** 事件。当文本框具有输入焦点（即处于活动状态）时，键盘上输入的每个字符都将在该文本框中显示出来。只有当一个文本框被激活，并且可见性为 True 时，才能接收到焦点。

③ **LostFocus** 事件。从表面看，**LostFocus** 是“失去指针”（即光标离开），也就是说，当光标离开时，就执行该事件的请求，而所谓的“指针离开”，实际上是光标离开文本框，即“失去

输入控制权”。当按下 Tab 键使光标离开当前文本框，或者用鼠标选择窗体中的其他对象时，就会触发该事件。为了检查用户输入的内容是否符合要求，通常使用 LostFocus 事件，而不使用 Change 事件，因为后者的发生过于频繁。

④ KeyPress 事件。该事件与键盘输入有关，适用于窗体和大部分控件，用来识别键入的字符。当在键盘上按下某个键时，触发该事件。

(3) 文本框的方法。SetFocus 方法是文本框常用的方法。

格式：[对象.]SetFocus。

功能：该方法可以把光标移动到指定的文本框中，使指定的文本框获得焦点。当在窗体上建立了多个文本框后，可以用该方法把光标置于所需要的文本框上。

4) 命令按钮

(1) 命令按钮的属性。在应用程序中，命令按钮通常用于单击时执行指定的操作。以前介绍的大多数属性都可以用于命令按钮，包括 Caption, Enabled, FontName, FontSize, ForeColor, Height, Left, Name, Visible, Top 和 Width 等。除此以外，还有以下一些属性：

① Cancel 属性。该属性用于设置命令按钮的作用是否等同于按 Esc 键的功能。当该属性设置为 True 时，按该命令按钮的效果与按 Esc 键的效果等同。只有命令按钮具有这个属性，而且在一个窗体窗口中至多只能有一个命令按钮的 Cancel 属性可以设置为 True。

② Default 属性。该属性用于设置命令按钮的作用是否等同于按回车键的功能。当该属性设置为 True 时，按该命令按钮的效果与按回车键的效果等同。只有命令按钮具有这个属性，而且在一个窗体窗口中至多只能有一个命令按钮的 Default 属性可以设置为 True。

③ Style 属性。Style 属性用于设置或返回一个值，这个值用来指定控件的显示类型和操作。该属性在运行期间是只读的。Style 属性可用于多种控件，包括复选框、组合框、列表框、单选按钮和命令按钮等。当用于命令按钮（或者复选框和单选按钮）时，可以取以下两种值：

第一种，0（符号常量 vbButtonStandard）：标准样式。控件按 Visual Basic 老版本中的样式显示，即在命令按钮中只显示文本（Caption 属性），没有相关的图形。此为默认设置。

第二种，1（符号常量 vbButtonGraphical）：图形格式。控件用图形样式显示，在命令中不仅显示文本（Caption 属性），而且可以显示图形（Picture 属性）。

④ Picture 属性。该属性可以用来给命令按钮指定一个图形。为了使用这个属性，必须把 Style 属性设置为 1（图形格式），否则 Picture 属性无效。

⑤ DownPicture 属性。该属性用来设置当控件被单击并处于按下状态时在控件中显示的图形，可用于复选框、单选按钮和命令按钮。为了使用这个属性，必须把 Style 属性设置为 1（图形格式），否则 DownPicture 属性将被忽略。如果没有设置 DownPicture 属性的值，则当按钮被按下时将显示赋值给 Picture 属性的图形。如果既没有设置 Picture 属性的值，也没有设置 DownPicture 属性的值，则在按钮中只显示标题（Caption 属性）。如果图形太大，超出按钮边框，则只显示其中的一部分。

⑥ DisabledPicture 属性。该属性用来设置对一个图形的引用，当命令按钮禁止使用（即 Enabled 属性被设置为 False）时在按钮中显示该图形。和 Picture、DownPicture 属性一样，必须把 Style 属性设置为 1，才能使 DisabledPicture 属性生效。和图片框的 Picture 属性一样，在设计阶段可以从属性窗口中设置命令按钮的 Picture、DownPicture 或 DisabledPicture 属性，也

可以通过 LoadPicture 函数装入图形。

(2) 命令按钮的事件。命令按钮最常用的事件是单击 (Click) 事件, 当单击一个命令按钮时, 触发 Click 事件。

注意: 命令按钮不支持双击 (DblClick) 事件。

(3) 命令按钮的方法。因为在命令按钮、复选框和单选按钮上不能显示任何字符 (用 Caption 属性设置除外), 所以, 前面的所有方法对它们均不适用。

1.2 实验

1.2.1 实验 1: 一个简单 Visual Basic 应用程序的建立

1) 实验目的

- (1) 掌握 Visual Basic 6.0 的启动与退出。
- (2) 熟悉 Visual Basic 的集成开发环境。
- (3) 了解在 Visual Basic 集成开发环境中编写一个 Visual Basic 应用程序的一般步骤。

2) 实验内容

编一个程序, 程序运行的初始界面如图 1-2-1 所示。单击“隐藏”按钮, 标签中的文本内容隐藏, 单击“显示”按钮, 标签中显示“北京欢迎你”。

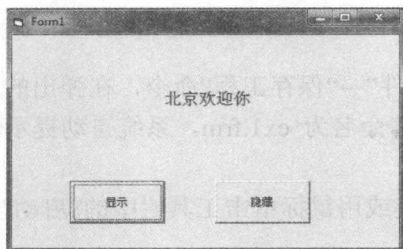


图 1-2-1 程序运行的初始界面

3) 实验步骤

(1) 启动 Visual Basic 6.0。执行“开始”→“程序”→“Microsoft Visual Studio 6.0 中文版”→“Microsoft Visual Basic 6.0 中文版”命令, 打开“新建工程”对话框。选择“新建”选项卡中的“标准 EXE”项, 进入如图 1-2-2 所示 Visual Basic 6.0 应用程序集成开发环境。

(2) 在 E 盘上建立一个文件夹 (如: EX1) 以保存该实验生成的各种文件。

(3) 用鼠标单击工具箱中的标签控件, 在窗体的合适位置绘制标签。设置标签控件属性, Caption 属性设置为“北京欢迎你”, 单击 Font 属性右边列表, 在弹出的字体对话框中将字号设置为四号。

(4) 在窗体中的合适位置绘制一命令按钮, 设置其 Caption 属性为“显示”。

(5) 在“显示”按钮的右边再绘制一命令按钮, 设置其 Caption 属性为“隐藏”。

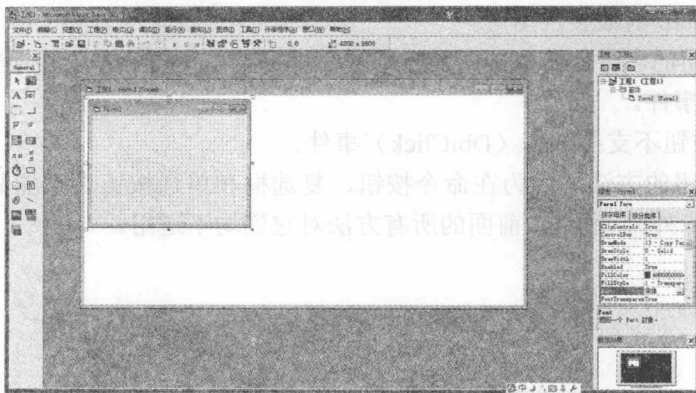


图 1-2-2 Visual Basic 6.0 应用程序集成开发环境

(6) 双击“显示”命令按钮打开代码窗口，在代码窗口中添加如下代码：

```
Private Sub Command1_Click()  
    Label1.Visible = True  
End Sub
```

(7) 在代码窗口的对象列表框中选择对象“Command2”，在事件列表框中选择事件“Click”，输入如下代码：

```
Private Sub Command2_Click()  
    Label1.Visible = False  
End Sub
```

(8) 保存文件。执行“文件”→“保存工程”命令，在弹出的对话框中将窗体文件保存在 E 盘 EX1 文件夹下，该窗体文件命名为 ex1.frm，系统自动提示保存工程文件，并将工程文件 ex1.vbp 保存在该文件夹下。

(9) 运行程序。按 F5 键或用鼠标单击工具栏中的“启动”按钮。单击“隐藏”按钮隐藏文字，单击“显示”按钮显示文字。

(10) 生成可执行文件。执行“文件”→“生成 ex1.exe”命令，将该文件保存在 EX1 文件夹下。

1.2.2 实验 2: 一个 Visual Basic 显示程序

1) 实验目的

- (1) 熟悉窗体、标签、命令按钮的最常用属性、事件和方法。
- (2) 熟悉 Visual Basic 中面向对象程序设计的一般方法，理解对象、对象的属性、事件和方法的含义。

2) 实验内容

编一程序，在屏幕上显示“你好，Visual Basic 系统”，字体大小设为三号、颜色为红色（见图 1-2-3），单击窗体后，在窗体上显示“初次见面，请多关照！”，同时窗体上出现“继续”和“结束”两个命令按钮（见图 1-2-4），如果单击“继续”按钮，则又回到初始运行状态；单击“结束”按钮，结束程序运行。

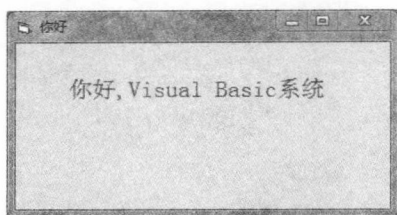


图 1-2-3 程序运行界面 1

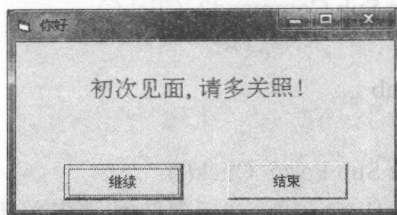


图 1-2-4 程序运行界面 2

3) 实验步骤

(1) 界面设计。在窗体上绘制一个标签和两个命令按钮，并调整位置，如图 1-2-5 所示。

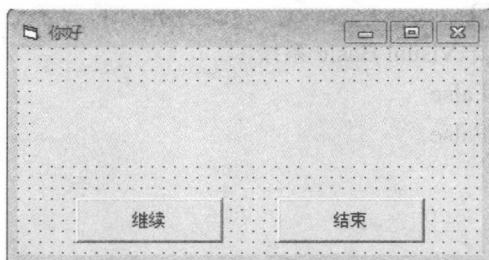


图 1-2-5 界面设计参考图

(2) 属性设置。按照表 1-2-1 设置各对象的属性。

表 1-2-1 属性设置

控件对象	属 性	属性值
窗体	Name	Form1
	Caption	你好
标签	Name	Label1
	Forecolor	红色
	Font	三号
命令按钮 1	Name	Command1
	Caption	继续
命令按钮 2	Name	Command2
	Caption	结束

(3) 代码编写:

```
Private Sub Command1_Click()  
    Label1.Caption = "你好, Visual Basic 系统"  
    Command1.Visible = False  
    Command2.Visible = False  
End Sub
```

```
Private Sub Command2_Click()  
    End  
End Sub  
  
Private Sub Form_Click()  
    Label1.Caption = "初次见面,请多关照!"  
    Command1.Visible = True  
    Command2.Visible = True  
End Sub  
  
Private Sub Form_Load()  
    Label1.Caption = "你好,Visual Basic 系统"  
    Command1.Visible = False  
    Command2.Visible = False  
End Sub  
(4) 保存工程。  
(5) 运行调试。
```

1.2.3 实验 3: 改变字体程序

1) 实验目的

- (1) 熟悉文本框的最常用属性、事件和方法。
- (2) 熟悉属性设置的两种途径。

2) 实验内容

要求: 编一程序(见图 1-2-6), 其中文本框的背景色为黄色, 前景色为红色, 单击“放大”按钮, 文本框显示的文字放大 3 倍; 单击“加粗”按钮, 文本框显示的文字加粗; 单击“斜体”按钮, 文本框文字加粗; 单击“还原”按钮, 文本框中的文本格式恢复到初始状态。

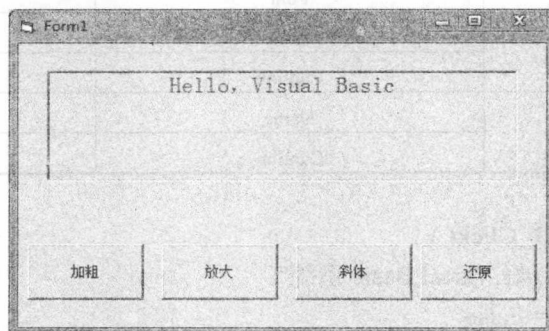


图 1-2-6 程序运行界面

3) 实验步骤

- (1) 界面设计。在窗体上绘制一个文本框和四个命令按钮，调整其大小和位置。
- (2) 属性设置。按照表 1-2-2 设置各对象的属性。

表 1-2-2 属性设置

控件对象	属 性	属性值
文本框	Name	Text1
	Text	Hello, Visual Basic
	Alignment	2
	Font	14
	BackColor	&H0080FFFF&
	ForeColor	&H000000FF&
命令按钮 1	Name	Command1
	Caption	加粗
命令按钮 2	Name	Command2
	Caption	放大
命令按钮 3	Name	Command3
	Caption	斜体
命令按钮 2	Name	Command4
	Caption	还原

(3) 代码编写:

```
Private Sub Command1_Click()  
    Text1.FontBold = True  
End Sub
```

```
Private Sub Command2_Click()  
    Text1.FontSize = Text1.FontSize * 3  
End Sub
```

```
Private Sub Command3_Click()  
    Text1.FontItalic = True  
End Sub
```

```
Private Sub Command4_Click()  
    Text1.FontSize = 14  
    Text1.FontBold = False  
    Text1.FontItalic = False
```

End Sub

(4) 保存工程。

(5) 运行调试。

1.2.4 实验 4: 改变窗体大小程序

1) 实验目的

熟悉窗体的最常用属性、事件和方法。

2) 实验内容

要求: 编一程序窗体颜色为绿色, 窗体初始大小为 4000×6000, 窗体标题上显示“单击窗体改变窗体大小”。单击窗体, 窗体大小改变为 5000×5000, 并且窗体标题上显示“双击窗体还原”。双击窗体, 窗体恢复初始大小, 窗体标题上显示“单击窗体改变窗体大小”。

3) 实验步骤

(1) 设置窗体属性 BackColor 为 &H0000C000&。

(2) 添加如下代码:

```
Private Sub Form_Click()
```

```
Form1.Height = 5000
```

```
Form1.Width = 5000
```

```
Form1.Caption = "双击窗体还原"
```

```
End Sub
```

```
Private Sub Form_DblClick()
```

```
Form1.Height = 4000
```

```
Form1.Width = 6000
```

```
Form1.Caption = "单击窗体改变窗体大小"
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Form1.Height = 4000
```

```
Form1.Width = 6000
```

```
Form1.Caption = "单击窗体改变窗体大小"
```

```
End Sub
```

(3) 保存工程。

(4) 调试运行。

1.3 练习题

1) 选择题

(1) 和其他的传统程序设计语言相比较, Visual Basic 最突出的特点是 ()。