

张青 王晓伟 编著

工程软件 开发技术

GONGCHENG RUANJIAN KAIFA JISHU



国防工业出版社

National Defense Industry Press

工程软件开发技术

张青 王晓伟 编著

国防工业出版社

·北京·

内 容 简 介

本书系统介绍了工程软件和工程数据库的概念、结构、开发方法,以 Visual Basic 为主要工具,结合工程实践给出大量例子,由浅入深全面阐述了高级程序设计语言的计算原理和基本特征、面向对象程序设计方法与步骤、算法与数据结构、数据库技术、软件工程以及人工智能程序设计。力图通过有限的篇幅,使读者建立起完整、系统和实用的工程软件开发思想。

随着计算机软件技术的发展,程序设计、软件开发对非计算机专业的工程技术人员来说已不再神秘。本书从基础做起,在实践中训练工程技术人员的编程能力,以期实现更多的“一劳永逸”,使我们的设计工作变得快捷而又高效。

本书面向工程,将软件开发技术集成起来,具有知识系统性、内容先进性和思想实用性的特点。既可为工程技术人员提供指导,也可作为高等学校各工程专业的教材。

图书在版编目(CIP)数据

工程软件开发技术/张青,王晓伟编著. —北京:国防工业出版社,2006.11

ISBN 7-118-04790-2

I. 工... II. ①张...②王... III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字(2006)第 119529 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100044)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 21¼ 字数 530 千字

2006 年 11 月第 1 版第 1 次印刷 印数 1—4000 册 定价 39.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)68428422

发行邮购: (010)68414474

发行传真: (010)68411535

发行业务: (010)68472764

前 言

工程软件开发技术是一门理论性和实践性都较强的综合性技术。随着计算机软件技术的发展,掌握程序设计、软件开发技术对非计算机专业的工程技术人员来说,好比如虎添翼。各工程领域的技术人员已感受到了计算机软件的神奇功力,在工程实践中更体会到它所带来的高效和便捷,同时也寄予它更高的要求 and 希望。计算机软件开发有无穷的潜力,但是专业的、自己的事情想让软件代劳,纯粹的拿来主义是肯定不能满足要求的。本书即是立足于此,软件知识零起点,从基础做起,力图通过有限的篇幅,使读者建立起开发工程软件的完整、系统和实用的思想。

工程软件开发技术是工程技术人员面向自己所从事的工程领域,针对重复性的工作和事务进行程序设计或软件开发,从而实现“一劳永逸”的技术。它包括面向对象程序设计方法与步骤(高级程序设计语言的计算原理和基本特征、界面设计、属性设置、代码编写、程序调试与错误处理、图形技术、文件)、算法与数据结构、(工程)数据库技术、软件工程和人工智能程序设计等内容和技术。本书以 Visual Basic 为主要工具,结合工程实践给出了大量例子,由浅入深全面系统地阐述了工程软件和工程数据库的概念、结构、开发方法。读者按照本书的思路,努力训练自己的编程能力,不需多时一定能够实现更多的“一劳永逸”,使我们的设计工作快捷而又高效。

本书面向工程,将软件开发技术集成起来,具有知识系统性、内容先进性和思想实用性的特点。可为工程技术人员提供指导,也可作为高等学校各工程专业的教材。

本书第 1、2、3、4、5、7、8、9、10、12 章由张青撰写,第 6、11 章由王晓伟撰写,全书由张青统一定稿。

为本书的完成做出大量工作的还有吕艳丽、陈国香、王玉玲、王胜春、靳同红、沈孝芹、刘青峰、刘宪琰、姜华、张瑞军、王积永、李楠楠、毛瑞年、闫玉芹、赵文波等。

由于作者水平有限,时间仓促,书中缺点错误在所难免,敬请读者批评指正,提出宝贵意见。

张青 王晓伟
2006 年 6 月

目 录

第1章 导论	1	2.4.4 类继承	39
1.1 软件与程序	1	2.4.5 多态性	42
1.1.1 程序的特性	1	2.4.6 类继承带来的新问题	43
1.1.2 软件的特性	3	2.5 网络计算时代的编程语言	46
1.2 软件的分类与工程软件	3	2.5.1 HTML 和 XML	47
1.3 工程软件开发基本技术	4	2.5.2 脚本语言	49
1.3.1 程序重用技术	5	2.5.3 网络编程语言	51
1.3.2 数据处理技术	6	第3章 VB 程序设计基础	55
1.3.3 程序结构组织	7	3.1 Visual Basic 集成开发环境	55
1.3.4 工程软件集成方法	8	3.1.1 主窗口	55
1.4 软件技术在工程中的应用	10	3.1.2 工具箱	56
第2章 高级程序设计语言	11	3.1.3 窗体设计器	57
2.1 高级程序设计语言概述	11	3.1.4 工程资源管理窗口	57
2.2 高级程序设计语言实现计算的方式	13	3.1.5 属性窗口	57
2.2.1 编译器工作原理	13	3.1.6 代码编辑器窗口	58
2.2.2 高级语言程序的解释执行	15	3.1.7 窗体布局窗口	58
2.3 高级程序设计语言的基本特征	16	3.1.8 对象浏览窗口	59
2.3.1 变量、表达式、赋值	16	3.2 窗体	59
2.3.2 程序的控制结构	17	3.2.1 窗体创建	59
2.3.3 数据类型	22	3.2.2 窗体的属性	60
2.3.4 过程	26	3.2.3 事件	62
2.3.5 过程的数据传递	29	3.2.4 方法	64
2.3.6 变量的生命期和 Static 变量	30	3.2.5 处理多重窗体	65
2.3.7 输入/输出	31	3.3 基本控件	67
2.4 面向对象程序语言的基本特征	32	3.3.1 命令按钮	67
2.4.1 对象概述	32	3.3.2 标签	68
2.4.2 类与对象	34	3.3.3 文本框	69
2.4.3 类定义	38	3.4 应用程序设计	70
		3.4.1 创建工程	71
		3.4.2 界面设计	71

3.4.3 属性设置	72	5.1.2 数据输出	93
3.4.4 代码编写	72	5.1.3 数据输入	94
3.4.5 保存、调试与运行	72	5.2 选择结构	98
3.4.6 生成可执行文件	72	5.2.1 If 条件语句	98
3.5 程序错误类型与调试	72	5.2.2 If 条件语句的嵌套	104
3.5.1 错误类型	72	5.2.3 多分支条件选择语句	105
3.5.2 程序调试	73	5.3 循环结构	106
3.5.3 错误处理	75	5.3.1 Do...Loop 语句	106
第4章 VB 语言基础	77	5.3.2 For...Next 循环语句	109
4.1 数据类型	77	5.3.3 循环嵌套	110
4.1.1 标准数据类型	77	5.4 其他辅助控制语句	112
4.1.2 用户自定义类型	79	第6章 算法与数据结构	114
4.2 常量	79	6.1 概论	114
4.2.1 常量的概念	79	6.1.1 引言	114
4.2.2 符号常量	79	6.1.2 有关概念和术语	115
4.3 变量	80	6.1.3 算法与数据结构研究内容 与关系	117
4.3.1 变量的命名规则	81	6.2 算法	117
4.3.2 变量的声明	81	6.2.1 算法的定义	117
4.4 运算符与表达式	83	6.2.2 算法设计的要求	118
4.4.1 算术运算符与算术表达式	83	6.2.3 算法表示形式	119
4.4.2 关系运算符与关系表达式	84	6.2.4 算法性能分析	120
4.4.3 逻辑运算符与逻辑表达式	84	6.2.5 常用算法	120
4.4.4 字符串运算符与字符串表 达式	85	6.3 数据结构	124
4.4.5 日期运算符与日期表达式	86	6.3.1 数据结构概论	125
4.4.6 运算符的优先顺序	86	6.3.2 线性表	126
4.5 常用内部函数	86	6.3.3 树和二叉树	133
4.5.1 数学运算函数	87	6.3.4 图	138
4.5.2 字符串函数	87	6.4 数组	141
4.5.3 日期与时间函数	88	6.4.1 数组概念	142
4.5.4 数据类型转换函数	89	6.4.2 数组的声明	142
4.5.5 格式输出函数	89	6.4.3 数组赋值	143
4.6 编码规则	90	6.4.4 数组的算法	144
第5章 基本控制结构	92	6.4.5 控件数组	148
5.1 顺序结构	92	6.5 排序	149
5.1.1 几个常用的语句	92	6.5.1 排序概述	150

6.5.2	排序方法	150	8.3.3	Pset 方法	216
6.5.3	排序方法的比较	153	8.3.4	Point 方法	217
第 7 章	界面设计	155	8.4	应用	218
7.1	标准控件	156	8.4.1	工程图形绘制	218
7.1.1	单选按钮和复选按钮	156	8.4.2	简单动画设计	221
7.1.2	框架	158	8.4.3	图形漫游	223
7.1.3	列表框和组合框	159	8.4.4	图形处理技术	225
7.1.4	滚动条和 Slider 控件	164	第 9 章	文件	229
7.1.5	时钟	166	9.1	文件的基本概念	229
7.1.6	图形控件	168	9.1.1	记录	229
7.1.7	文件系统控件	172	9.1.2	文件	229
7.2	菜单设计	174	9.1.3	文件的种类	229
7.2.1	菜单编辑器	174	9.2	顺序文件	230
7.2.2	动态菜单	177	9.2.1	打开文件	230
7.2.3	弹出菜单	178	9.2.2	写操作	230
7.3	高级控件	179	9.2.3	关闭文件	231
7.3.1	通用对话框	179	9.2.4	读操作	231
7.3.2	多重窗体和多文档界面	185	9.3	随机文件	235
7.3.3	工具栏和状态栏	191	9.3.1	文件的打开和关闭	235
7.3.4	RichTextBox 控件	197	9.3.2	文件的读写	235
7.3.5	ProgressBar 控件	200	9.3.3	Seek()函数和 Seek 语句	236
7.3.6	UpDown 控件	201	9.4	二进制文件	236
7.3.7	Animation 控件	202	第 10 章	数据库技术	237
7.3.8	SSTab 控件	204	10.1	数据库概念	237
第 8 章	图形技术	207	10.2	数据库管理器	239
8.1	图形技术基础	207	10.3	数据控件	242
8.1.1	坐标系统	207	10.3.1	数据控件属性	242
8.1.2	自定义坐标系	209	10.3.2	数据控件的事件	245
8.1.3	图形层	211	10.3.3	数据控件的常用方法	246
8.2	绘图属性	212	10.3.4	记录集的属性与方法	246
8.2.1	当前坐标	212	10.3.5	数据库记录的增、删、改 操作	250
8.2.2	线宽与线型	212	10.4	ADO 数据控件	253
8.2.3	填充与色彩	214	10.4.1	ADO 对象模型	253
8.3	图形方法	214	10.4.2	使用 ADO 数据控件	254
8.3.1	Line 方法	214	10.4.3	ADO 控件上新增绑定控件	
8.3.2	Circle 方法	216			

的使用	256	11.5 软件审查与测试	299
10.4.4 使用数据窗体向导	257	11.5.1 软件审查	299
10.5 结构化查询语言	259	11.5.2 软件测试	299
10.5.1 结构化查询语言	259	11.5.3 测试方法	300
10.5.2 使用 SELECT 语句查询 ...	260	11.5.4 测试步骤	300
10.6 工程数据库	263	11.5.5 调试	301
10.6.1 工程数据库的概念	263	11.6 软件维护	302
10.6.2 工程数据库与传统关系型 数据库的比较	273	11.6.1 软件维护概述	302
第 11 章 软件工程	283	11.6.2 决定软件可维护性的 因素	302
11.1 软件工程概述	283	11.6.3 文档	303
11.1.1 软件危机	283	第 12 章 人工智能程序设计	305
11.1.2 软件工程内涵	284	12.1 逻辑型编程语言	305
11.1.3 软件工程的原则	284	12.2 LISP 语言	306
11.1.4 软件工程的基本原理	285	12.2.1 LISP 的特点和数据结构 ...	306
11.2 软件生命周期	286	12.2.2 LISP 的基本函数	308
11.2.1 软件定义时期	286	12.2.3 递归和迭代	311
11.2.2 软件开发时期	287	12.2.4 LISP 编程举例	313
11.2.3 软件运行与维护	287	12.3 PROLOG 语言	315
11.3 软件需求	287	12.3.1 语法与数据结构	316
11.3.1 软件需求分析	287	12.3.2 PROLOG 程序设计原理 ...	316
11.3.2 需求建模	290	12.3.3 PROLOG 编程举例	318
11.3.3 需求分析文档	290	12.4 专家系统	319
11.4 软件设计	291	12.4.1 知识获取及表述	319
11.4.1 软件设计概述	291	12.4.2 知识的运用	323
11.4.2 模块化原理	292	12.4.3 专家系统的结构	326
11.4.3 软件设计过程	292	12.4.4 应用实例	328
11.4.4 软件设计工具	299	参考文献	331

FontUnderLine 属性是逻辑型，决定对象上正文是否加下划线。

例如在运行时将窗体字体设置为黑体，字号为 20，斜体，可用以下语句实现：

Form1.FontName = “黑体”

Form1.FontSize = 20

Form1.FontItalic = True

7. Icon 属性

该属性用于设置窗体左上角的示意图标。在属性窗口中单击选中该属性项，然后单击右边设置框中的带省略号的按钮，此时将弹出加载图框的对话框，在该对话框中选择要加载的图标文件 (*.ico 或 *.cur)。也可在运行时，用 Load Picture 函数装入，其语句格式为：

Form1.Icon = Load Picture (图形文件路径名)

8. Picture 属性

该属性用于设置窗体中要显示的图片。加载图片的操作同加载图标的操作相同。

9. StateUpPosition 属性

该属性用于设置窗体启动时所出现的位置，其取值与对应功能如表 3-2 所列。

表 3-2 StateUpPosition 属性取值与对应功能

属性 设置值	相应功能
0	没有指定初始设置值，由窗体的 Left、Top 决定
1	用户窗体所有者中央
2	屏幕中央
3	屏幕的左上角

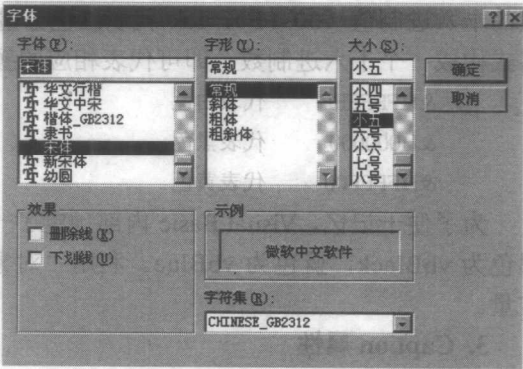


图 3-13 “字体”对话框

10. Visible 属性

该属性为逻辑型值，决定窗体是否可见，其默认值为 True。若将其设置为 False，则窗体运行时不可见。

11. WindowState 属性

该属性用于返回或设置窗体运行的状态，其取值有 0、1、2 这 3 种，分别代表窗体运行时正常化、最小化和最大化，其默认值为 0。

12. Left、Top、Height、Width 属性

Left、Top 属性决定窗体运行时相对于屏幕左上角的坐标值，Height、Width 属性用于设置或返回窗体的高度和宽度，默认单位是缇 (twip)，1 缇 = 1/20 点 = 1/1440 in = 1/567cm。

3.2.3 事件

窗体事件是窗体识别的动作。与窗体有关的事件较多，下面介绍几个常用窗体事件。

1. Click 事件

程序运行时用鼠标单击窗体内的某个位置，Visual Basic 将触发窗体的 Click 事件。事件过程格式为：

```
Private Sub Form_Click ()
```

```
·  
·  
·
```

```
End Sub
```

2. DblClick 事件

程序运行时用鼠标双击窗体内的某个位置，Visual Basic 将触发窗体的 DblClick 事件。事件过程格式为：

```
Private Sub Form_DblClick ()
```

```
·  
·  
·
```

```
End Sub
```

3. Load 事件

当窗体装入内存时引发该事件，它由系统操作触发或通过 Load 语句触发。该事件过程常用于在窗体装入内存时，进行一些初始化处理。事件过程格式为：

```
Private Sub Form_Load ()
```

```
·  
·  
·
```

```
End Sub
```

4. Unload 事件

当窗体从内存中卸载时引发该事件，利用 Unload 事件可在关闭窗体或结束应用程序时做一些必要的善后处理工作。事件过程格式为：

```
Private Sub Form_Unload (Cancel As Integer)
```

```
·  
·  
·
```

```
End Sub
```

其中，Cancel 可设置为 0 或非 0 值。当设置为 0（默认值）时，表示允许关闭窗体；当设置为非 0 值时，表示取消当前关闭窗体的操作。

5. Resize 事件

当窗体首次显示在屏幕上或者窗体大小被改变时触发 Resize 事件。事件过程格式为：

```
Private Sub Form_Resize ()
```

```
·  
·  
·
```

```
End Sub
```

例如，下列代码将窗体 Form1 始终显示在屏幕中央。

```
Private Sub Form_Resize ()
```

```
Form1. Move ( Screen. Width - Form1. Width ) / 2, ( Screen. Height - Form1. Height ) / 2
End Sub
```

3.2.4 方法

下面介绍窗体的几个常用方法。

1. Print 方法

该方法常用于在窗体、图片框或打印机上输出文本信息或表达式的值，其格式为：

```
[ 对象名. ] Print [ 表达式表 ] [, ! ; ]
```

语句说明：

(1) 对象名可以是窗体、图片框或打印机对象 (Printer)。对象名省略时，默认在当前窗体上输出。

(2) 表达式表是数值型表达式或字符型表达式的一个列表。运行时按照原样输出表达式表。若是省略表达式表，则打印一空行。

(3) 表达式表中有多个表达式时，各表达式之间用逗号或分号分隔。若用逗号分隔，则数据项按标准格式输出，每 14 列为一输出区，将一行分为若干输出区，逗号后面的表达式在下一输出区显示输出。若各表达式之间用分号分隔，则以紧凑格式输出。对于数值型数据，输出时前面有一个符号位，后面有一个空格，而字符串输出时，前后都没有空格。

(4) 一般情况下，执行完 Print 语句后系统会自动换行，但若语句尾带有逗号或分号，则 Print 语句执行完后，系统不会换行，下一个 Print 语句的输出仍将在该行继续输出。

例如：

```
Print 1, -1, 2, "abc", "def"
```

```
Print 1; -1; "abc"; "def"
```

```
Print "22 + 33 = ";
```

```
Print 22 + 33
```

其输出结果为：

```
1          -1          2          abc          def
1 -1 2 abcdef
22 + 33 = 55
```

2. Cls 方法

该方法用于清除窗体或图片框中用 Print 方法显示的信息和用绘图方法生成的图形，并将不可见的图形光标重新定位到窗体或图片框左上角 (0, 0)。

语句格式为：

```
[ 对象名. ] Cls
```

对象名省略，则清除当前窗体中所显示的内容。

设计时使用 Picture 属性设置的背景位图和放置的控件不受 Cls 方法影响。

3. Move 方法

该方法用于移动窗体，并可在移动时动态改变其大小。语句格式为：

```
[ 对象名. ] Move X [, Y [, width [, Height ]]]
```

参数 X 和 Y 代表移动到目标位置的坐标；Width 和 Height 代表移动到目标位置后对象的宽度和高度，通过这两个参数实现大小的调整。若省略 Width 和 Height 参数，则移动过程中

保持对象大小不变。

例如, 要将窗体 Form1 移动到屏幕的 (200,200) 处, 并使其大小为高 600、宽 800, 可使用语句为:

```
Form1. Move 200, 200, 600, 800
```

4. Hide 方法

该方法用于隐藏窗体, 使某个已显示的窗体变为不可见, 但仍保留在内存中 (与卸载不同)。以后需要再显示隐藏起来的窗体时, 执行 Show 方法即可。

语法格式为:

```
[窗体名.] Hide
```

省略窗体名称时默认为当前窗体。

5. Show 方法

该方法用于显示窗体, 让窗体成为可见。语法格式为:

```
[窗体名.] Show [窗体显示模式]
```

如果 Show 方法作用的窗体还未装入内存, 则系统会自动将其装入内存, 然后再显示, 也就是说它兼有加载和显示窗体两种功能。“窗体显示模式”参数项为可选项, 用于指定窗体以哪种方式显示。其取值为 0 或 1, 默认为 0。若设为 1, 则表示窗体以模态方式显示, 此时用户只能在本窗体中操作, 不能切换到其他窗体操作, 直到本窗体被关闭为止。如 Office 软件中“帮助”菜单的“关于”命令所打开的对话框窗口即是这种窗口。若参数项设为 0, 则窗体以非模态方式显示, 用户可在本窗体与其他窗体之间任意切换。如“编辑”菜单的“替换”对话框就是一个非模态对话框的实例。

省略窗体名称时默认为当前窗体。

当窗体成为活动窗口时发生窗体的 Activate 事件。

3.2.5 处理多重窗体

在编写实际应用程序时, 常常要用到多个窗体才能实现其设计要求, 下面介绍这些窗体间的处理。

1. 窗体的装载与卸载

(1) 若要装载新窗体, 语句格式为:

```
Load 窗体名
```

该语句用于将指定的窗体装入内存, 但不显示出来。如要显示, 还应调用 Show 方法。执行 Load 语句后, 可以引用窗体中的控件及各种属性。对于窗体而言, 在装载的过程中, 将引发窗体的 Load 事件。

(2) 窗体在运行过程中不再需要时, 应及时卸载掉, 以释放其所占用的系统资源。窗体的卸载语句格式为:

```
Unload 窗体名 | Me
```

该语句用于将窗体从内存中卸载掉。在卸载时, 将引发窗体的 Unload 事件。Unload 语句的一种常见用法是 Unload Me, Me 特指当前窗体。

2. 启动窗体的设计

一个应用程序可以包含多个窗体, 但只能将一个窗体对象设置为启动对象。

选择“工程”菜单中“工程 1 属性”菜单项或者用鼠标指向工程资源管理器中的工程,

单击右键，从弹出的快捷菜单中选择“工程 1 属性”菜单项，弹出如图 3-14 所示的“工程属性”对话框。在“通用”选项卡的“启动对象”下拉框中选择启动对象，默认为 Form1。

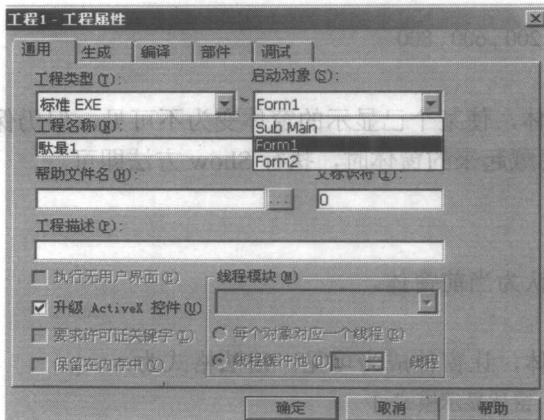


图 3-14 “工程属性”对话框

例 3-1 设计 Form1 和 Form2 两个窗体，Form1 为启动窗体。要求如下：

- (1) 单击 Form1 窗体，在窗体上显示“工程软件开发技术”字样，宋体 28 号字；若用户单击“显示窗体二”命令按钮，则显示窗体 Form2，隐藏 Form1。
- (2) 单击 Form2 窗体，在窗体上显示“山东建筑大学”字样，宋体 28 号字；若用户单击“返回窗体一”命令按钮，则显示窗体 Form1，隐藏 Form2。
- (3) 若用户单击“退出”命令按钮，则结束程序运行。

运行界面如图 3-15 所示。

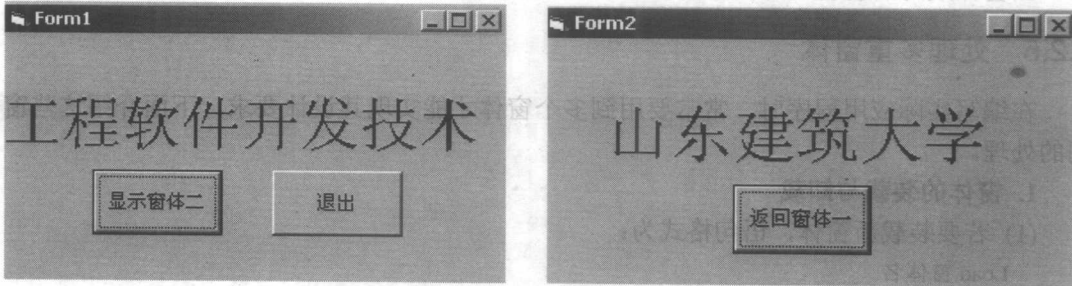


图 3-15 例 3-1 运行界面

添加两个窗体 Form1 和 Form2，Form1 上放置两个命令按钮，Form2 上放置一个命令按钮，其属性设置如表 3-3 所列。

表 3-3 控件属性值

对 象	属性名称	属 性 值	对 象	属性名称	属 性 值
Form1	Caption	Form1	Form2	Caption	Form2
Command1	Caption	显示窗体二	Command3	Caption	返回窗体一
Command2	Caption	退出			

Form1 窗体代码：
Private Sub Form_Load()

```

Form1.FontName = "宋体"
Form1.FontSize = 28
End Sub
Private Sub Form_Click()
    Print "工程软件开发技术"
End Sub
Private Sub Command1_Click()
    Form1.Hide
    Form2.Show
End Sub
Private Sub Command2_Click()
    End
End Sub
Form2 窗体代码:
Private Sub Form_Load()
    Form2.FontName = "宋体"
    Form2.FontSize = 28
End Sub
Private Sub Form_Click()
    Print "山东建筑大学"
End Sub
Private Sub Command3_Click()
    Form2.Hide
    Form1.Show
End Sub

```

3.3 基本控件

3.3.1 命令按钮

命令按钮(CommandButton)在工具箱中的图标是，用于接收用户的操作信息，并引发应用程序的某个操作。在 Visual Basic 应用程序中，命令按钮是使用最多的控件对象之一。默认名称为 Command1、Command2 等。

命令按钮接受用户输入的命令可以有 3 种方式：

- (1) 鼠标单击。
- (2) 按 Tab 键焦点跳转到该按钮，再按回车键。
- (3) 快捷键 (Alt + 有下划线的字母键)。

下面介绍命令按钮的常用属性、方法和事件。

1. 常用属性

Name、Height、Width、Top、Left、Visible、Font 等属性与在窗体中的使用相同。

(1) **Caption** 属性。该属性用来决定显示在命令按钮上的文本即标题。默认值为 **Command1**、**Command2** 等。**Caption** 属性最多包含 255 个字符。

可通过 **Caption** 属性创建命令按钮的访问键,其方法是在欲作为快捷访问键的字母前面加上一个“&”符号。例如,将 **Caption** 属性设置为“打印 (&P)”,则运行时出现  的按钮外观。只要用户同时按下 **Alt** 键和 **P** 键,就能执行按钮功能。

(2) **Default** 属性和 **Cancel** 属性。窗体上放置的命令按钮常会有一个默认按钮和一个取消按钮。所谓默认按钮是指无论当前焦点位于何处,只要按下 **Enter** 键,就能自动执行该命令按钮的 **Click** 事件过程;而取消按钮则是指只要按下 **Esc** 键,就能自动执行该命令按钮的 **Click** 事件过程。

命令按钮的 **Default** 属性和 **Cancel** 属性分别用于设置默认按钮和取消按钮,当其值设置为 **True** 时,表示将对应的命令按钮设定为默认按钮或取消按钮。

注意:一个窗体中只能设置一个默认按钮和一个取消按钮。

(3) **Enable** 属性。该属性决定按钮是否可用,默认值为 **True**。当设置 **Enabled** 属性为 **False** 时,运行时命令按钮将以不可用的浅灰色显示,不能被按下。

(4) **Style** 属性。该属性设置命令按钮的显示形式,默认值 0 表示以标准的 **Windows** 按钮显示,只能显示文字;其值为 1 表示以图形按钮显示,此时可利用 **Picture**、**DownPicture** 和 **DisabledPicture** 属性指定在不同状态下显示的图片。

(5) **Value** 属性。在程序代码中也可以触发命令按钮,只需将该按钮的 **Value** 属性设置为 **True**,即可触发命令按钮的 **Click** 事件,执行命令按钮的 **Click** 事件过程。

2. 常用方法

命令按钮的常用方法为 **SetFocus** 方法,其语法格式为:

命令按钮控件名. **SetFocus**

该方法可用来使命令按钮获得焦点,这时若用户按下 **Enter** 键,就会执行该命令按钮的 **Click** 事件过程。

3. 常用事件

命令按钮最常用事件是 **Click** 事件。

3.3.2 标签

标签(**Label**)在工具箱中的图标是 。标签的功能比较简单,通常用来显示比较固定的提示信息,不允许用户在程序运行时输入数据。标签的默认名称为 **Label1**、**Label2** 等。

1. 常用属性

(1) **Alignment** 属性。该属性用于设置标签中显示文本的对齐方式。其取值有 0、1、2、3 种,分别为左对齐、右对齐和居中,默认值为 0。

(2) **AutoSize** 属性。该属性决定控件是否能自动调整大小以显示所有的文本内容,其取值为 **True** 或 **False**,默认值为 **False**。

(3) **BackStyle** 属性。该属性用于设置标签的背景是否透明。其取值为 0 或 1,分别为透明、不透明,默认值为 1。

(4) **Caption** 属性。这是标签控件最重要的属性,用来设置或返回标签中显示文本,最多为 1024 个字符。

(5) **MousePointer** 属性。该属性设置在运行时当鼠标移动到标签时显示的鼠标指针的类

型。属性值设为 99 时为用户自定义。

(6) **MouseIcon** 属性。该属性设置自定义的鼠标图标。此时 **MousePointer** 属性应为 99。

(7) **WordWrap** 属性。该属性设置为 **True** 时, 标签将在垂直方向变化大小以与标题相适应。此时 **AutoSize** 属性应设置为 **True**, 设置为 **False** (默认值) 时标签不能在垂直方向上扩展。

2. 常用方法

标签常用的方法主要是 **Move** 方法。

3. 常用事件

标签常用的事件主要有 **Click**、**DbClick** 等。

3.3.3 文本框

文本框(**TextBox**)在工具箱中的图标为。文本框是 **Visual Basic** 的一个重要控件, 既可接受用户的输入信息, 又可显示输出信息。默认名称为 **Text1**、**Text2** 等。

1. 常用属性

(1) **Alignment** 属性。该属性设置文本框的文本对齐方式, 默认值 0 表示左对齐; 值 1 表示右对齐; 值 2 表示居中。

(2) **Text** 属性。该属性设置或返回文本框中显示的文本内容。文本框无 **Caption** 属性, 它是利用 **text** 属性来存放文本信息的。在程序运行期间, 用户可向文本框输入文本信息, 输入的信息自动存入 **Text** 属性中。因此, 在编程中, 可通过访问文本框的 **Text** 属性来获得用户的输入值。若要清除文本框中内容, 可将 **Text** 属性设置为空。

(3) **MultiLine** 属性。该属性决定文本框是否以多行方式显示文本, 其默认值为 **False**, 表示文本框只能以单行方式显示或输入文本。若其值设置为 **True**, 则表示文本框可输入或显示多行文本, 当文本长度超过文本框长度时, 文本内容会自动换行。

(4) **MaxLength** 属性。该属性设置文本框中文本的最大长度, 即文本框中最多允许放入的字符个数。默认值为 0, 表示无字符长度限制。若为大于 0 的整数, 表示用户可输入的最多字符数。

(5) **PasswordChar** 属性。该属性决定文本框中是否显示用户输入的字符, 常用于密码输入。只能是一个字符, 如 **PasswordChar** 属性设置为 “*”, 则无论用户在文本框中输入什么字符, 都将以 “*” 代替, 但内部输入的文本内容不会改变。只有在 **MultiLine** 属性设置为 **False** 的前提下, **PasswordChar** 属性才起作用。

(6) **ScrollBars** 属性。该属性设置文本框中是否出现水平或垂直滚动条。共有 4 个可设置值: 默认值为 0, 表示不出现滚动条; 值为 1 表示出现水平滚动条; 值为 2 表示出现垂直滚动条; 值为 3 表示同时出现水平滚动条和垂直滚动条。但是使属性值为 1、2、3 有效的前提是 **MultiLine** 属性必须设置为 **True**。

(7) **Locked** 属性。该属性设置文本框是否可以编辑修改。默认值为 **False**, 表示文本框可以编辑修改; 若设为 **True**, 则表示文本框只读。

(8) **SelStart** 属性、**SelLength** 属性和 **SelText** 属性。当用户选定文本内容时常用到这 3 个属性, 它们只在运行阶段有效, 在设计阶段不能设置。

SelStart 属性用于设置或返回所选定文本的起点, 即从第几个字符开始选定。默认值为 0, 表示从第一个字符开始。

SelLength 属性用于设置或返回所选定文本的长度, 即字符的个数。若属性值为 0, 表示

不选任何字符。

`SelText` 属性用于设置或返回所选定文本的内容。

2. 常用方法

文本框最常用的方法为 `SetFocus` 方法。

3. 常用事件

(1) `Change` 事件。当文本框的内容发生变化时，触发该事件。

(2) `Lost Focus` 事件。当文本框失去焦点时，触发该事件。常用来判断文本框的当前输入内容是否合法，以决定是否转移焦点。

(3) `Key Press` 事件。当文本框接受用户输入时，每一次键盘输入，都将使文本框接受一个 ASCII 字符，发生一次 `Key Press` 事件。该事件过程格式为：

```
Private Sub Text1_KeyPress ( KeyAscii As Integer )  
.  
.  
.  
End Sub
```

其中，参数 `KeyAscii` 值为相应字符的 ASCII 值。

3.4 应用程序设计

一个 Visual Basic 应用程序也称为一个工程，一个工程的一般组成情况如表 3-4 所列。

表3-4 工程的组成

文件类型	说明
工程文件（.vbp）	与该工程有关的全部文件和对象的清单
窗体文件（.frm）	包含窗体及控件的属性设置；窗体级的常量、变量和外部过程的声明；事件过程和自定义过程
窗体的二进制数据文件（.frx）	如果窗体上控件的数据属性含有二进制属性（例如图片或图标），当保存窗体文件时，就会自动产生同名的.frx 文件
标准模块文件（.bas）	包含模块级的常量、变量和外部过程声明；用户自定义的、可供工程内各窗体调用的过程。该文件是可选项
类模块文件（.cls）	用于创建含有方法和属性的用户自己的对象，该文件是可选项
资源文件（.res）	包含着不必重新编辑代码就可以改变的位图、字符串和其他数据。该文件是可选项
ActiveX 控件的文件（.ocx）	ActiveX 控件的文件是一段设计好的可以重复使用的程序代码和数据，可以添加到工具箱，并可像其他控件一样在窗体中使用。该文件是可选项

下面以一具体实例说明创建 Visual Basic 应用程序过程。

创建一应用程序，其功能是求两个数的和。当用户分别在两个文本框中输入数字，单击“计算”按钮后，在第三个文本框中显示两数字之和。程序运行界面如图 3-16 所示。