

Visual Basic

高等学校Visual Basic课程系列教材

Visual Basic

课程设计

(第2版) 王 栋 主编



清华大学出版社

高等学校Visual Basic课程系列教材

Basic Visual Basic 课程设计 (第2版) 王 栋 主编

清华大学出版社
北 京

内 容 简 介

Visual Basic 课程设计是高等学校“Visual Basic 程序设计”课程的后续实践环节,目的是使学生在掌握了 Visual Basic 的基础知识和基本概念之后提高综合应用所学知识编制实用程序的能力。本书是专为 Visual Basic 课程设计编写的指导教材。

书中提供了作者精心挑选的 51 个设计题目,内容涵盖了 Visual Basic 程序设计各个方面的基础知识,既具有实用性、针对性、典型性,又不失趣味性。每个题目均给出了详细的背景介绍、设计目标和功能要求;对于题目中的重点和难点,做了适当的提示、讲解或给出了部分关键代码。所附光盘提供了每个题目的标准样例源程序和所需资源供读者参考和借鉴。

通过对书中题目的学习和练习,读者必将深化对 Visual Basic 基本概念和算法的理解,提高对应用程序设计的总体把握能力,树立将所学知识应用于解决实际问题的信心。

本书既可以用做 Visual Basic 课程设计的教材和指导书,也可以作为“Visual Basic 程序设计”课程的上机参考书及程序设计爱好者的学习参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Visual Basic 课程设计/王栋主编.--2 版.--北京:清华大学出版社,2014
高等学校 Visual Basic 课程系列教材
ISBN 978-7-302-34722-4

I. ①V… II. ①王… 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(2013)第 292346 号

责任编辑:闫红梅

封面设计:傅瑞学

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.5 字 数:449 千字
(附光盘 1 张)

版 次:2004 年 4 月第 1 版 2014 年 1 月第 2 版 印 次:2014 年 2 月第 1 次印刷

印 数:1~2000

定 价:29.50 元

产品编号:042714-01

前言

Visual Basic 程序设计被许多高校作为程序设计语言课程已有若干年了。在这几年里，和其他的语言一样，学生们学完了、考完了、等级证书也拿到了，但因为缺少实践，接下来没多长时间就忘光了。学生普遍的反映是：学过了程序设计，仍然不会设计程序，就像学习外语，记了一大堆的词汇和语法却不能张口表达。而后续课程的教师也在抱怨：如今大学生计算机接触得越来越多，可编程能力却未有大的长进！

究其原因，有多方面，笔者认为，程序设计课的学时少，缺少综合性的归纳和练习；学生上机实习时间短，实际操作不多，没有时间编制完整的程序是最重要的原因。

不少高校已经认识到了这个问题，并采取了相应的措施，其中比较科学有效的办法是增加课程设计环节。课程设计是指学生在程序设计课之后，用规定的时间相对独立地完成一个实用程序开发的过程。课程设计对于学生深化理解 Visual Basic 语言的基础知识、提高综合运用所学知识的能力有着非常显著的作用。

在课程设计实施过程中，教师们遇到的最大问题是没有足够的设计题目。所以，我们组织了一批有丰富教学和实践经验的教师编写了这本《Visual Basic 课程设计》。本书分为 8 章，提供了 8 种类型共 51 个设计题目，每章均有“编程指导”一节，讲述本章要用到的知识和技巧；每个题目都有详细的背景介绍、设计目标和功能要求，并对题目中的重点和难点做了适当的提示和讲解。配套光盘提供了每个题目的标准样例程序（由 Visual Basic 6.0 生成）及所需的数据、图片和声音等资源 and 全部源代码供读者参考和借鉴。

我们在筛选题目时，坚持了以下原则：注重针对性、实用性和趣味性；每个题目都尽量多地涉及基础知识，同时又各有侧重点；选题内容覆盖面广、种类齐全、互相不重复；杜绝了太难、太易、太偏和太繁的题目。为了便于读者选题和评估，51 个设计题目按难度分为五个等级（以“☆”个数表示，个数越多，难度越大）。除了基本要求之外，每个题目还有一个名为“更上一层楼”的小节，鼓励读者将程序做得更完善。

本教材第 1 版经过了近 10 年实际教学使用，取得了良好的效果，积累了不少经验。第 2 版对第 1 版进行了修改，删除了个别题目，又新增了一些题目。附带的光盘中包含了完整的源程序。

参加本书编写的还有王芳、王育荣、马银忠、袁红兵、吴志林、陆静、王涛、符意德、宋斌、李向东等教师，在此表示真诚的感谢。

欢迎广大读者对本书提出意见和建议，可通过电子邮件：wangdong@njust.edu.cn 与作者联系。



2013 年 9 月 于紫金山麓

使用说明

如果本书用做大学第一门编程语言的课程设计教材，请参考以下使用说明。

1. 各学校根据学生情况、课程进度和学时数来安排学生选题，可以一人一题，也可以一人多题或多人一题，原则上由学生自由选题，也可以使用题目难度进行控制。譬如，规定一个难度总和（如4级），学生可以选做一个4级题目，也可以选做两个2级题目，或3级、1级各一个，等等（每一章最多选一题）。如果完成的题目难度总和超过规定难度，酌情加分。

2. 课程设计过程中，学生应能相对自由地支配时间、查阅相关资料、相互讨论，教师安排答疑。设计的过程应是一个开拓思路、互相协作、自主学习、分析问题和解决问题的过程，对学生是全方位训练。

3. 在配套光盘中，每个题目都有两个对应的文件夹，其中一个文件夹（可执行文件）提供了样例程序的可执行文件和运行所需的所有其他文件。**Resource**子文件夹中提供了该样例程序设计时用到的数据文件、图标文件、图片文件和声音文件等素材，该文件夹在程序运行时不需要。

因为大多数程序有写文件的操作，所以在光盘中运行会出错，应将题目的整个文件夹复制到硬盘中，去掉所有文件的只读属性，然后再运行。光盘根目录下的**Gifts**文件夹中提供了大量的图片、图标、光标和声音文件，可以用来创建工具栏、定制窗口图标、鼠标指针和调试程序。

另一个文件夹（源程序）中包含了生成这些可执行文件的源程序。希望学生能先自己编程序，提供的代码仅供参考。

4. 对于所有的设计题目，有一些统一的要求，其中包括：为每个窗口指定一个合适的图标；运行程序时窗口自动位于屏幕中央；合理设置窗口的边框类型，没有特别要求的程序，窗口应为单线不可缩放的边框；文件操作时应使用相对路径，确保可执行文件及其相关数据文件放置在硬盘的任何目录下均能正常运行。

5. 本书提供的样例程序仅作为参考，题目中的设计要求是最基本的要求，读者在完成这些基本要求之后，应考虑使程序更加合理、完善，功能更强。

6. 编程时，要求模块名、过程名、变量名具有描述性，源程序中添加必要的注释内容，语句嵌套采用缩进格式。尽量采用结构化和模块化的设计思想，将共用的、相对独立的代码设计为通用过程或函数。

7. 课程设计结束时，提交完成的所有源程序、相关文件和可执行文件。同时撰写《课程设计报告》，其中主要包括对题目的分析，采用的解决方案，使用的模块、过程、变量数组及相互关系，设计中遇到的主要问题及解决方法，设计中尚存的不足，对设计的感想和心得体会，等等。

8. 教师在考核学生的设计时，应有面试、答辩和上机测试过程。这样可以比较全面、准确地了解学生对基础知识的掌握和对设计题目所做的工作，同时有效地杜绝作弊现象的发生。

编 者

目 录

第 1 章 基本控件应用	1
1.1 编程指导 (一)	1
1.1.1 Visual Basic 编程的一般步骤与注意事项	1
1.1.2 为窗体对象设置正确的边框样式	2
1.1.3 为窗口设置合适的图标	3
1.1.4 窗体的启动位置和状态	3
1.1.5 使用工具提示	4
1.1.6 多模块编程的注意事项	4
1.1.7 正确设置工程属性	4
1.1.8 控件数组的使用	5
1.1.9 添加 ActiveX 控件	6
1.1.10 使用 SSTab 控件	7
1.1.11 使用 TabStrip 控件	8
1.1.12 使用对象型变量	9
1.1.13 使用 Tag 属性和 TypeOf 关键字	9
1.2 题目 1 按揭购房还款计划 [☆]	10
1.3 题目 2 彩票选号小助手 [☆]	13
1.4 题目 3 滚动字幕板 ^{☆☆}	16
1.5 题目 4 打冰雹游戏 ^{☆☆}	18
1.6 题目 5 计算器 ^{☆☆}	20
1.7 题目 6 排序演示 ^{☆☆}	22
1.8 题目 7 围棋 ^{☆☆}	25
第 2 章 字符串处理	29
2.1 编程指导 (二)	29
2.1.1 定长字符串与变长字符串	29
2.1.2 特殊字符的表示	29
2.1.3 常用字符串处理函数与运算符	29
2.1.4 认识与使用文本文件	31
2.1.5 常用的公共对象	33
2.1.6 使用公共对话框控件	35
2.1.7 多文档界面	38
2.2 题目 8 邮编区号查询 [☆]	40
2.3 题目 9 列车时刻查询 ^{☆☆}	44
2.4 题目 10 记事本程序 ^{☆☆}	47
2.5 题目 11 多文档文本编辑器 ^{☆☆}	49
2.6 题目 12 电子书 ^{☆☆}	52
2.7 题目 13 背单词 ^{☆☆}	56

2.8 题目 14 英汉词典 ^{☆☆}	58
第 3 章 图片应用	63
3.1 编程指导(三)	63
3.1.1 Visual Basic 支持的图片格式	63
3.1.2 图片的使用	63
3.1.3 大量图片的管理方法	65
3.1.4 图片的掩码	69
3.1.5 鼠标指针样式的设置	69
3.1.6 使用 MMControl 控件播放声音	70
3.1.7 使用 ProgressBar 进度条控件	72
3.2 题目 15 图片浏览器 [☆]	73
3.3 题目 16 快速配对游戏 ^{☆☆}	75
3.4 题目 17 多功能计时器 ^{☆☆}	78
3.5 题目 18 存包柜 ^{☆☆}	81
3.6 题目 19 拼图游戏 ^{☆☆☆}	85
3.7 题目 20 动物管理员游戏 ^{☆☆☆☆}	90
第 4 章 数据管理	96
4.1 编程指导(四)	96
4.1.1 自定义数据类型	96
4.1.2 动态数组	96
4.1.3 随机文件的使用	97
4.1.4 如何使用 API 函数	99
4.1.5 配置文件的应用	103
4.1.6 使用 MSFlexGrid 控件	105
4.1.7 使用 MonthView、DTPicker 和 Calendar 控件	108
4.1.8 使用 MaskedTextBox 控件	109
4.2 题目 21 机房管理程序 ^{☆☆}	111
4.3 题目 22 电子台历 ^{☆☆}	115
4.4 题目 23 通讯录管理 ^{☆☆☆}	118
4.5 题目 24 个人收支管理 ^{☆☆☆}	123
4.6 题目 25 学生信息管理 ^{☆☆☆}	126
4.7 题目 26 打印准考证 ^{☆☆☆}	129
第 5 章 游戏设计	137
5.1 编程指导(五)	137
5.1.1 处理键盘事件	137
5.1.2 随机值的生成	140
5.2 题目 27 记忆力测试 ^{☆☆}	140
5.3 题目 28 数字拼图游戏 ^{☆☆}	143
5.4 题目 29 打字练习 ^{☆☆☆}	145
5.5 题目 30 贪吃蛇游戏 ^{☆☆☆☆}	148

5.6	题目 31 俄罗斯方块游戏☆☆☆	151
5.7	题目 32 扫雷游戏☆☆☆	154
5.8	题目 33 纸牌接龙游戏☆☆☆	158
第 6 章 图形绘制		166
6.1	编程指导 (六)	166
6.1.1	使用颜色	166
6.1.2	绘制曲线	167
6.1.3	输出文字	168
6.1.4	显示图片	170
6.1.5	保存位图文件	170
6.1.6	坐标系统	171
6.1.7	创建非矩形窗口	172
6.1.8	使窗口位于最顶层	175
6.1.9	使用二进制文件	175
6.1.10	使用 UpDown 微调按钮控件	176
6.2	题目 34 虚拟波形发生器☆☆	177
6.3	题目 35 模拟监控程序☆☆	180
6.4	题目 36 时钟设计☆☆	182
6.5	题目 37 曲线显示程序☆☆☆	184
6.6	题目 38 显示区位点阵汉字☆☆☆	187
6.7	题目 39 绘图程序☆☆☆	190
6.8	题目 40 模拟手机短消息编辑器☆☆☆☆☆	195
6.9	题目 41 五笔字型测试系统☆☆☆☆☆	198
第 7 章 科学计算		204
7.1	编程指导 (七)	204
7.1.1	使用 Sleep 函数	204
7.1.2	使用 DoEvents 函数	204
7.1.3	错误的捕获与处理	205
7.1.4	使用 MSChart 图表控件	207
7.2	题目 42 汉诺塔问题☆☆☆	214
7.3	题目 43 回转体体积计算☆☆☆	217
7.4	题目 44 线性方程组求解☆☆☆	220
7.5	题目 45 直线拟合☆☆☆	224
7.6	题目 46 圆弧插补☆☆☆	227
第 8 章 高级控件应用		233
8.1	编程指导 (八)	233
8.1.1	使用 RichTextBox 控件	233
8.1.2	使用 ToolBar 控件	235
8.1.3	使用 CoolBar 控件	238
8.1.4	使用 StatusBar 控件	240

8.1.5	使用 TreeView 控件.....	242
8.1.6	使用 ListView 控件.....	247
8.1.7	使用 ImageCombo 框控件.....	252
8.1.8	拖放操作	253
8.1.9	使用文件系统对象	255
8.1.10	使用 DAO 对象访问.mdb 数据库.....	260
8.2	题目 47 格式文本编辑器☆☆☆	265
8.3	题目 48 自测题系统☆☆☆☆	268
8.4	题目 49 中图分类号查询系统☆☆☆☆	273
8.5	题目 50 学生成绩管理☆☆☆☆☆	277
8.6	题目 51 文件管理器☆☆☆☆☆	281

注：“☆”号表示题目的难度。

第1章

基本控件应用

1.1 编程指导（一）

1.1.1 Visual Basic 编程的一般步骤与注意事项

（1）启动 Visual Basic，创建新的工程。对于本书的所有题目，均应选择“标准 EXE”工程类型。

（2）通过“属性”窗口将工程名和窗体名改为具有描述意义的值。例如，编制排序程序时，可将工程名改为 Sort，将主窗体名改为 frmSortMain。尽量不使用默认值（如“工程 1”和 Form1 等）。

（3）通过 Windows 资源管理器创建一个专用文件夹（如 Sort），将整个工程保存到该文件夹中。默认情况下，工程文件名和窗体文件名与工程名和窗体对象名同名，如 Sort.vbp 和 frmSortMain.frm。

（4）通过 Visual Basic 集成开发环境“工程”菜单中的“引用”和“部件”对话框将要使用的外部引用或 ActiveX 控件添加到工程中；使用 API 函数浏览器将要调用的 API 函数的声明添加到工程中。

（5）在窗体上添加控件，将控件的默认名称改为具有描述性的名称，如“确定”按钮使用 cmdOK 作为对象名。切勿使用 Command1 和 Text1 这样的默认对象名。

（6）编写代码。包括定义自定义数据类型，定义全局和模块级变量、常量和数组，编写通用过程和通用函数以及事件过程。代码中的各种名称（如变量名、数组名和过程名）也应采用描述性的命名规则，保证通过名称便能判断出其作用。

在编程前应设置 Option Explicit 和 Option Base 1|0 等语句。“强制变量定义”不论对于初学者还是有经验的编程人员来说都是很重要的。一般情况下，不使用变体数据类型（Variant）。

（7）对于多模块程序，添加多个窗体模块或标准模块。保存时应将一个工程的所有文件保存在同一文件夹中。复制工程时，只需将整个文件夹复制即可，既方便又不易出错。

（8）编程时既要注意代码简洁，又要注意结构合理，更要注意程序的执行效率。一般情况下，对象属性的存取速度比变量的存取速度慢得多，所以应避免将对象属性用于循环语句中。

建议将大段的代码分割改写为相互独立的通用过程或函数，因为这样更利于程序的排错和维护。不过，过程或函数的调用需要占用大量系统资源，所以对于经常调用、反复执

行的代码不一定非要分解成多个小过程，因为这样会使执行效率降低。

有时，执行效率和代码大小、内存占用量是相互矛盾的。下面是两个功能相同的过程，第①个过程比第②个过程少用了一个变量 **n**，因此占用内存更小；而第②个过程只调用了一次 **UBound** 函数，当数组 **a** 很大时，第②个过程的执行速度明显会更快。到底该侧重于哪一方面，应根据具体的程序来定夺。

①

```
Private Sub p(a() As Integer)
    Dim s As Long : Dim i As Integer
    i = LBound(a)
    Do
        s = s + a(i)
        i = i + 1
        If i > UBound(a) Then Exit Do
    Loop
    Print s
End Sub
```

②

```
Private Sub p(a() As Integer)
    Dim s As Long : Dim n As Integer : Dim i As Integer
    i = LBound(a) : n = UBound(a)
    Do
        s = s + a(i)
        i = i + 1
        If i > n Then Exit Do
    Loop
    Print s
End Sub
```

1.1.2 为窗体对象设置正确的边框样式

窗体的 **BorderStyle** 属性不但决定其边框和标题栏的宽度和可缩放性，而且决定该窗体是否会在 Windows 任务栏上显示相应的图标。表 1.1 列出了 **BorderStyle** 属性的取值和意义。

表 1.1 窗体 **BorderStyle** 属性的取值和意义

属性值	常数	意义
0	vbBSNone	无边框和标题栏等元素，无法使用鼠标拖动和缩放，不在任务栏上显示图标
1	vbFixedSingle	固定单边框。有标题栏，但无最大化和最小化按钮。不能通过鼠标操作改变其大小。在任务栏上显示图标
2	vbSizable	（默认值）可缩放边框。可通过鼠标拖动边框或单击最大化、最小化按钮改变其大小，也可通过拖动其标题栏移动。在任务栏上显示图标

续表

属性值	常数	意义
3	vbFixedDialog	固定对话框边框。无最大化和最小化按钮，不能改变大小，可以拖动。不在任务栏上显示图标
4	vbFixedToolWindow	固定工具栏窗口。标题栏的高度较窄，无最大化最小化按钮，不能改变大小，可以拖动。不在任务栏上显示图标
5	vbSizableToolWindow	可变大小的工具栏窗口。除了可以拖动边框改变大小，其他与前一个相同

BorderStyle 常用的取值为 1、2 和 3。一般情况下，可将程序的主窗体设置为 1 或 2，将其他窗体设置为 3。如果主窗体上以常用的控件为主，则没有必要支持缩放，应设为 1；对于绘图类程序，总希望尽量使用屏幕的显示能力，应支持缩放，设为 2。如果将不应支持缩放的窗体使用了可缩放的边框，在最大化时就会出现大块没有控件的空闲区域，很不美观。对于一般的程序，除了主窗体，其他窗体都是作为对话框使用的，因此，其边框一般不需要缩放，也没有必要在任务栏上显示图标，所以应设为 3。

对于大多数程序，其主窗口应在 Windows 任务栏上显示一个图标，以方便用户在不同的程序之间切换，否则只有将其他程序全部关闭或最小化，才能显示出该程序的窗口。

不建议使用属性值为 0 的无边框模式。

1.1.3 为窗口设置合适的图标

大多数的商业化应用程序都有专门的窗口图标，例如 Word 和 Excel 等软件的图标和对于大多数的计算机用户来说是很熟悉的。即使作为简单的练习程序，也应该为程序指定专门的图标，如果所有程序都使用 VB 的默认图标，既不美观也无特色。因为大多数程序的图标在运行过程中保持不变，所以在设计阶段为窗体的 Icon 属性赋一个合适的.ico 文件即可。

本书每个样例程序的对应文件夹中提供了一个可用的图标文件。

1.1.4 窗体的启动位置和状态

初学者在练习编程时往往不注意窗体的显示位置，其实这是一个很重要的问题，用户启动一个程序总是希望窗口出现在最显眼、操作起来最方便的位置上，一般来说应是屏幕中央。如果在设计时通过设置窗体的 Left 和 Top 属性来决定其显示位置是费力而不讨好的，因为用户的显示分辨率各不相同。最简单的方法是将窗体的 StartUpPosition 属性设置为 2，窗体在启动时自动位于屏幕中心，无论显示器的分辨率为多少。对于一个用做对话框的窗体，将其 StartUpPosition 属性设置为 1（所有者中心），在使用 Show 方法显示该窗体时为其指定所有者窗体可以使之自动定位在指定的窗体中央（一般为主窗口）。例如，下面语句自动将 frmDlg 窗体（StartUpPosition 属性设为 1）显示在当前窗体的中央。

```
frmDlg.Show 1, Me
```

有些程序需要在启动时将窗口最大化，可以在窗体的 Load 事件过程中将其 WindowState 属性设置为 2。

1.1.5 使用工具提示

一个优秀的程序总是在任何细微处为用户的使用提供方便，其中“工具提示”就是一个途径。譬如在 Word 软件中，如图 1.1 所示，将鼠标指针在某个工具栏按钮上停留片刻，会在鼠标指针附近弹出一个框，其中的文字简要介绍该按钮的功能，这便是“工具提示”窗口。



图 1.1 工具提示

Visual Basic 为工具提示功能提供了完善的支持，使用起来非常方便。大多数控件都有 ToolTipText 属性，只要为该属性赋一个工具提示字符串便可。程序运行时，当用户将鼠标放到控件上，会自动显示工具提示窗口。

1.1.6 多模块编程的注意事项

(1) 多模块程序的关键问题是模块间的数据传递和共享。对于简单的多模块程序（如只有两三个窗体模块的小程序），可以在主窗体模块中定义全局变量，在所有的模块中可以访问，但必须在变量名前加上模块名称来修饰。如在 frmDialog 模块中访问 frmMain 模块中定义的全局变量 Var，必须写成 frmMain.Var 的形式。如果程序稍大一些或者需要使用全局的常量、全局数组、全局自定义数据类型时，必须使用标准模块。在其他模块中调用标准模块中定义的全局变量、数组、过程时可以直接使用其名称，而不需要加以修饰。

(2) 在使用多个窗体时，应注意显示窗体时的模态性。以 1 为参数调用窗体的 Show 方法会显示为模态窗口，不带参数调用则为非模态窗口。非模态窗口不能阻止用户操作本程序的其他窗口，一般不合作对话框，所以主窗体之外的其他窗体应显示为模态窗口。

(3) 一个 Visual Basic 程序的所有窗体只有卸载之后才会真正关闭整个程序，而有的初学者习惯使用 Visible 属性或 Hide 方法将某些窗体隐藏，这样关闭后实际程序仍然占用内存。多次运行这样的程序可能会使内存耗尽。

1.1.7 正确设置工程属性

打开一个 Visual Basic 工程，从“工程”菜单中选择最后一个菜单项“XX 属性”（其中“XX”为工程名），可以弹出图 1.2 所示的“工程属性”对话框。对于标准 EXE 类型的工程，通过“通用”和“生成”两个选项卡可以对工程进行常规设置。

在“通用”选项卡上，从“启动对象”组合框中选择要作为工程启动对象的窗体或过程，这对于多模块的程序是有用的。默认情况下，第一个添加到工程中的窗体（即新建工程时 Visual Basic 自动创建的窗体）是启动对象，如果要改变，则必须通过此对话框设置。

在“生成”选项卡上，可以对工程指定所编程序的版本号、添加产品名称、公司名

称、版权声明等信息。这些信息会被编译到生成的可执行文件中，通过 Windows 资源管理器中的“属性”功能可以查看可执行文件的这些信息。“标题”中输入的内容一般也是提供给操作系统的，譬如调用 MsgBox 消息框函数时若未指定第三个参数（即“标题”参数），会以这里设置的文字显示。如果已为工程中的每个窗体设置 Icon 属性（即窗口图标），可以通过“图标”组合框从中选择一个用作应用程序的图标（也就是可执行文件的图标）。

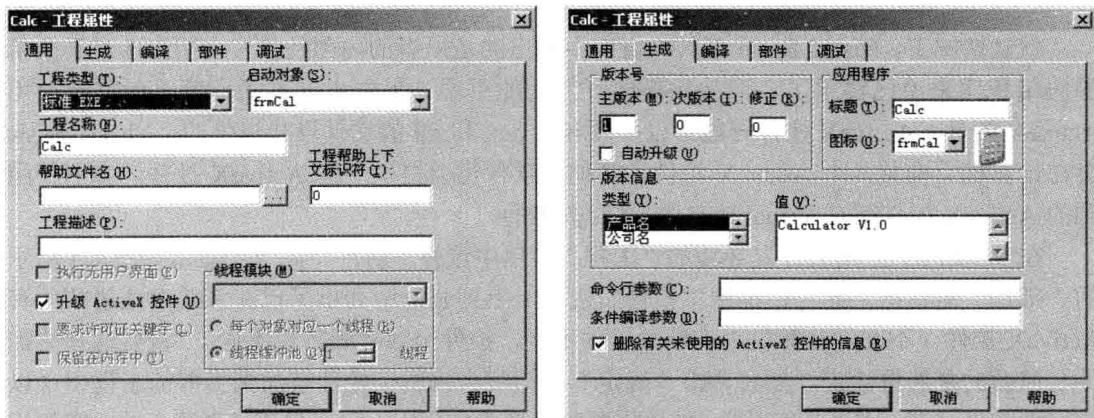


图 1.2 “工程属性”对话框的“通用”和“生成”选项卡

1.1.8 控件数组的使用

利用控件数组是使用多个同种类型控件的有效方法，一个控件数组中各个控件的 Name 属性相同，下标（即 Index 属性）不同。与常规数组不同，一个控件数组的下标可以不连续。假设一个控件数组 cmdArray 由三个控件组成，其 Index 属性可以分别是 1、2 和 3，也可以是 2、5 和 8。

在窗体上添加控件数组的方法有以下几种：（1）在设计时添加一个控件，使用复制和粘贴的操作将生成下标从 0 开始且连续的控件数组。（2）在设计时添加多个控件，然后通过“属性”窗口将这些控件的名称改为相同、Index 属性改为不同的下标值（可以连续也可以不连续）。（3）通过 Load 语句在运行时为已有的控件数组动态地添加新元素。如：

```
Load cmdArray(5)
```

使用 Load 添加的新控件是不可见的，一般先通过程序设置其位置和外观然后将 Visible 属性设为 True 使其可见。

在运行时，使用 Unload 语句卸载由 Load 语句添加的元素（不能卸载设计时添加的元素）。如：

```
Unload cmdArray(5)
```

控件数组被 Visual Basic 看作是一个集合，可以通过类似 cmdArray.Count、cmdArray.Ubound 和 cmdArray.Lbound 的方式得到一个控件数当前的控件个数、最大和

最小的下标。也可以使用 `cmdArray.Item(i)` 的方式访问指定下标的控件，等价于 `cmdArray(i)`。

注意：不能使用 `Ubound` 和 `Lbound` 函数得到控件数组的下标上界和下界，如 `Ubound(cmdArray)` 是错误的。

1.1.9 添加 ActiveX 控件

默认情况下，Visual Basic 工具箱窗口中列出的是所有的内部控件，这些控件对于简单的应用程序来说已够用了，但是如果要开发功能更强、界面更漂亮的程序，就需要使用 ActiveX 控件。ActiveX 控件一般是以扩展名为 `.ocx` 和 `.dll` 的文件形式提供的。Visual Basic 会在计算机上安装几个 ActiveX 控件，其他的软件也会提供一些 ActiveX 控件。本书使用到的 ActiveX 控件都是 Visual Basic 安装时提供的。

在 Visual Basic 集成开发环境的“工程”菜单中选择“部件”命令会显示如图 1.3 所示的“部件”对话框。此对话框中“控件”选项卡上列表框中列出了已在系统中注册的所有 ActiveX 控件（不同的计算机因安装的软件不同，控件个数和版本也不同）。将要选择的控件名前面的复选框选定，然后单击“确定”按钮就会将所选控件添加到当前的工程中（以图标形式显示在工具栏窗口中）。如果想要的控件未被列出，可以单击“浏览”按钮查找相应的文件。有时，“部件”窗口中的一项代表一组控件（如 Microsoft Windows Common Controls）。

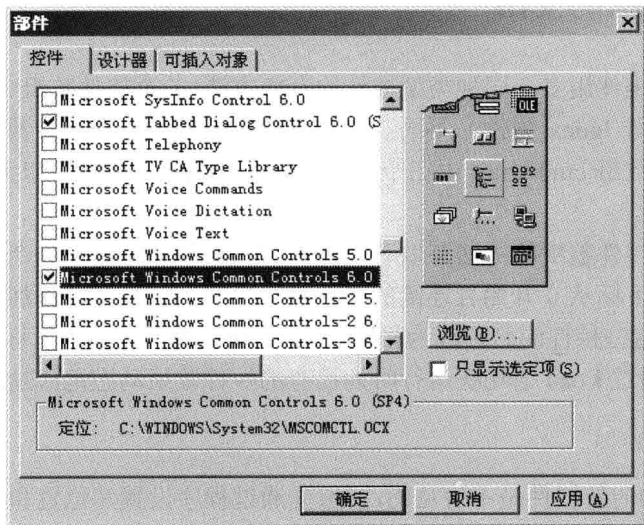


图 1.3 添加 ActiveX 控件时的“部件”对话框

把 ActiveX 控件添加到工具箱之后，便可以像使用内部控件一样地使用了：放置到窗体上、设置其属性、调用其方法、编写其事件过程。

对于多数 ActiveX 控件，在设计阶段有一种更好的办法设置其属性：在已经放置到窗体上的控件上单击鼠标右键，从菜单中选择“属性”会弹出“属性页”对话框。在此对话框中可以方便地、可视化地设置该控件的多数常用属性，与在属性窗口中设置相应的属性

是等价的。如果属性页对话框是中文的，使用会更方便。

1.1.10 使用 SSTab 控件

许多功能较强的软件在窗口上，特别是对话框上使用了选项卡，如图 1.4 所示是 Visual Basic 的“选项”对话框。使用选项卡之后，会以标签的形式组织显示大量的控件，使得本来需要多窗体的程序可以用一个窗体来实现其功能。在 Visual Basic 中使用选项卡可以使用两个常用的 ActiveX 控件：SSTab 控件或 TabStrip 控件。TabStrip 是 Windows 公共控件中的一个，将在 1.1.11 小节中介绍，与其相比，SSTab 控件使用起来更方便。

使用前一节介绍的方法，从“部件”对话框中选择 Microsoft Tabbed Dialog Control 项，将 SSTab 控件添加到当前工程中，该控件在工具箱窗口中显示为图标。在窗体上放置一个 SSTab 控件，既可以通过属性窗口，也可以通过“属性页”对其进行设置，使用其“属性页”（见图 1.5）更加直观。

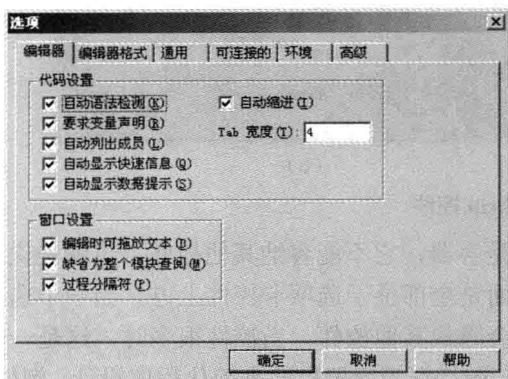


图 1.4 一个使用了选项卡的对话框

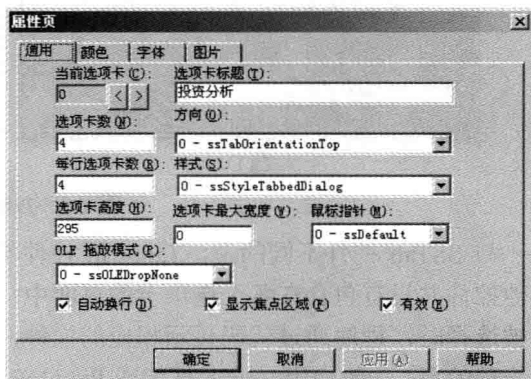


图 1.5 SSTab 控件的属性页

先设置“选项卡数”（Tabs 属性）、“每行的选项卡数”（TabsPerRow 属性，当选项卡很多时可以自动以多行显示）。然后通过单击按钮，切换当前选项卡（Tab 属性）。选项卡的编号是从 0 开始的，如果有 6 个选项卡，则相应的编号为 0~5。将某个选项卡设为当前选项卡之后，便可以编辑“选项卡标题”（Caption 属性）。还可以设置选项卡的样式（Style 属性）和标签位置（TabOrientation 属性）。

对于多数程序，选项卡的个数和标题是确定的，可以在设计时设置完毕，运行时不必改变。如果有特殊需要，可以使用以下方法通过编程进行设置。

SSTab1.Tabs = 6	' 设置选项卡数目
SSTab1.Tab = 0	' 将第 1 个选项卡设置为当前
SSTab1.Caption = "选项卡 1"	' 设置第 1 个选项卡的标题
SSTab1.Tab = 1	' 将第 2 个选项卡设置为当前
SSTab1.Caption = "选项卡 2"	' 设置第 2 个选项卡的标题

选项卡是用来分组和管理其他控件的，所以每个选项卡是一个“容器”。在设计时，单击一个选项卡的标签将其设为当前，然后可以把其他控件放置到该选项卡里面。无论在设

计时还是在运行时, SSTab 控件只显示当前选项卡上的控件, 这正是我们需要的。

在程序中, 通过代码可以访问不同的选项卡上的各种控件。

1.1.11 使用 TabStrip 控件

TabStrip 控件属于 Windows 公共控件中的一员, 使用 1.1.9 小节介绍的方法将“Microsoft Windows Common Controls”部件添加到工程中, 其工具箱图标为。与上一小节介绍的 SSTab 控件类似, TabStrip 控件的作用是在窗体上实现“选项卡”的功能。如图 1.6 所示, 除了可以定义选项卡个数、标题、位置, TabStrip 控件与 ImageList 控件(参见 3.1.3 小节)配合可以在其标签上显示图标。

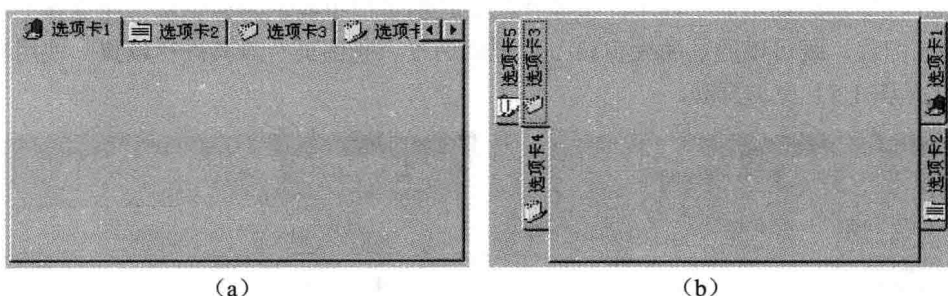


图 1.6 TabStrip 控件

与 SSTab 控件不同的是, TabStrip 控件不是容器, 它不能容纳其他控件。也就是说, 其他控件并没有包含在各个选项卡的页面中, 而是全部位于选项卡控件上方。当一个选项卡被选择时, 程序通过代码显示相应的控件, 会隐藏其他控件。当控件很多时, 这是一件很繁琐的事。一般使用 Frame 框架或 PictureBox 图片框等容器控件来简化程序设计。例如, 为每个选项卡页面准备一个框架控件, 将 BorderStyle 属性设置为 0 使其边框不可见, 然后将各种要用到的控件放置在相应的框架控件中。在选项卡控件的 Click 事件过程中根据其 SelectedItem 属性值判断哪个选项卡被选择, 使相应的框架控件可见, 其他框架不可见(容器的可见性决定其内部控件的可见性)。使用选项卡控件的 ClientTop、ClientLeft、ClientWidth、ClientHeight 属性可以判断当前选项卡客户区(即页面区域)的大小和位置, 将框架控件移到客户区上即可。

```
Private Sub TabStrip1_Click()
    Select Case TabStrip1.SelectedItem.Index
        Case 1
            '显示选项卡 1 上的控件
        Case 2
            '显示选项卡 2 上的控件
        ...
    End Select
End Sub
```