

在运行基于 Windows 的应用程序时,一般会接触到菜单、对话框及多文档的设计与应用,本章将介绍这些内容。

6.1 菜单及其应用

在基于 Windows 的图形用户界面的应用程序中,菜单是用户界面最重要的元素之一,菜单包含菜单的名称和菜单项的名称。在进行菜单设计时,最好能与 Windows 应用程序风格一致,或采用一些约定俗成的设计模式和风格,以方便用户使用。菜单提供了用户操作的接口,一个设计良好的菜单会使程序显得实用和美观,从而方便用户使用。

6.1.1 菜单简介

菜单实际上是应用程序提供给用户的一组命令的集合,每个菜单项是一个具体的操作指令,利用菜单,用户不必牢记命令,只要从菜单中找到相关命令,然后单击菜单项即可执行相应程序段,类似于执行一个命令。菜单通常放置在程序界面的顶部。下面以 IE 浏览器为例介绍菜单的一些基本概念,典型的 IE 浏览器界面如图 6-1 所示。

(1) 菜单栏。菜单栏一般出现在窗口标题栏的下面,由一排或多排菜单标题组成。

(2) 菜单标题。菜单标题是菜单命令的一个子集,通常包含多个菜单项。在图 6-1 中,File、Edit、View、Favorites、Tools 和 Help 都是菜单标题。单击菜单标题时,菜单项的列表就会打开,图 6-1 显示的就是已经打开的 Edit 菜单的菜单项列表。

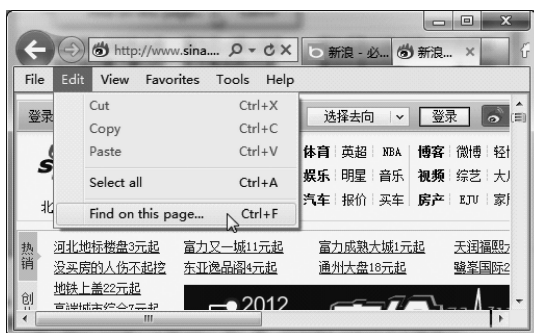


图 6-1 IE 浏览器界面

(3)菜单项。上述列表中的每一项都称为一个菜单项。菜单项可以是一个菜单命令,也可以是分隔线,还可以是子菜单标题。子菜单标题可以引出另外一个子菜单项列表。

(4)访问键。有时在菜单名字后面会带一个加括号的字母,如图 6-2 所示,在菜单名“文件”后面的字符是一个带括号的字母 F,这个字母被称为访问键,用户同时按 Alt+F 组合键就可以激活“文件”菜单,相当于单击“文件”菜单。同理,E 是“编辑”菜单的访问键。

(5)快捷键。快捷键也是组合键,按下时直接执行相应菜单项命令。例如,对于图 6-2 的 IE 浏览器中的“文件”菜单,按 Ctrl+O 组合键就可以打开如图 6-3 所示的 Open 对话框,提示用户输入所要访问的网址。

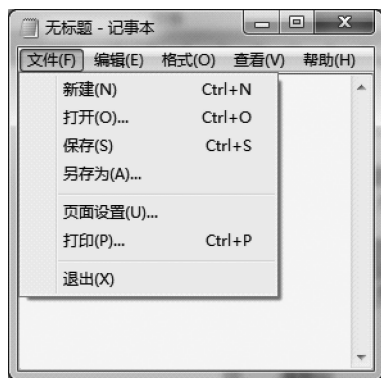


图 6-2 记事本的菜单

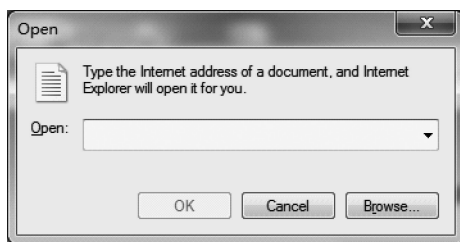


图 6-3 Open 对话框

6.1.2 菜单编辑器的应用

Visual Basic 提供了菜单编辑器,用户可以非常方便地通过菜单编辑器来创建菜单。

1) 启动菜单编辑器

要创建一个菜单,应先打开一个窗体,然后在窗体上右击,在弹出的快捷菜单中选择“菜单编辑器”选项,如图 6-4 所示,也可以执行“工具”→“菜单编辑器”菜单命令,如图 6-5 所示,打开的菜单编辑器界面如图 6-6 所示。



图 6-4 选择“菜单编辑器”选项



图 6-5 通过菜单命令启动菜单编辑器



图 6-6 菜单编辑器界面

用菜单编辑器可以创建新的菜单和菜单栏、在已有菜单上增加新的命令、用新的命令替换已有命令以及修改和删除已有菜单和菜单栏等。

在创建菜单时,应尽量采用符合 Windows 应用系统风格的用户已经熟知的分组方法。例如,“文件”菜单包含“新建”、“打开”、“保存”、“另存为”、“打印”等菜单项,“编辑”菜单包含“剪切”、“复制”、“粘贴”和“撤销操作”等菜单项。

在确定菜单中所要包含的菜单项及其功能后,便可以开始创建菜单了。大多数菜单控件的属性可以通过菜单编辑器设置,所有的菜单属性在属性窗口中也都是可用的。菜单控件的一些基本属性如下。

(1)标题。标题是实际出现在菜单上的文本,即菜单名,如“文件”、“编辑”、“工具”、“帮助”等。

(2)复选。该属性表示菜单项前面是否打勾,一般选中时会有一个勾的标记。

(3)有效。该属性表示菜单是否可用,有效时,菜单项为实体可见文字;无效时,菜单项呈灰色,单击没有任何响应,默认状态是有效的。

(4)可见。该属性表示菜单项是否显示出来,默认状态是可见的。

(5)快捷键。该属性表示菜单项对象的快捷键。

值得注意的是,为了保持 Windows 应用程序的风格,方便用户使用,设置菜单项标题时应尽量遵循以下规则。

- 菜单标题名称应当唯一,而且意义要明确。
- 每个名称都应当有一个可用键盘选取命令的访问字符,两个菜单标题不能用同一个访问字符。



- 如果命令在完成之前还需要附加信息,则在其名称后面应当有一个符号“...”。例如,显示“保存文件”对话框的命令“另存为...”,显示“颜色”对话框的命令“颜色...”等。

2) 使用菜单编辑器

菜单中的每个菜单项实际上都是一个菜单控件,位于菜单编辑器下部的列表框列出了当前窗体的所有菜单控件。

在图 6-6 所示的菜单编辑器中的“标题”文本框中输入一个菜单项名称,该项便会出现在菜单编辑器最下方的菜单控件列表框中,然后可以从该列表框中选取菜单控件并编辑其属性。

菜单控件在菜单控件列表框中的位置决定了该控件是菜单标题、菜单项、子菜单标题还是子菜单项。

在菜单编辑器中创建菜单控件的一般步骤如下。

- (1) 选取一个窗体,菜单栏将创建在该窗体中。
- (2) 打开菜单编辑器。
- (3) 在“标题”文本框中依次输入菜单名称,如“文件”、“编辑”、“工具”、“帮助”等。如果希望为某一菜单定义一个访问键,可以为该菜单定义一个访问键的字符,并在该字符前加上一个字符 &。

(4) 在“名称”文本框中输入该菜单控件的名称。

(5) 用上、下、左、右箭头按钮调整控件的缩进级,从而定义该菜单属于哪一级。

(6) 设置菜单控件的相关属性。

(7) 单击“下一个”按钮可以再创建一个菜单控件,或单击“插入”按钮可以在现有控件之间增加一个菜单控件,也可以通过上、下、左、右箭头按钮,在现有菜单控件之中移动控件。

至此,就在窗体上设计好菜单及菜单项了。

3) 分隔菜单项

为了更好地把某一类操作进行归类,提高视觉效果,方便查找,用户还可以用分隔线将同一菜单中的菜单项进行归类细分。

分隔条是显示在菜单上作为菜单项间的一个水平分割线。在菜单项很多的菜单上,可以使用分隔条将各项划分成一些逻辑组。需要说明的是,分隔线只能起分隔菜单项的视觉作用,并不能响应单击的消息。在菜单编辑器中创建分隔线的步骤如下。

- (1) 单击图 6-6 中的“插入”按钮,在想要分隔的菜单项之间插入一个菜单控件。
- (2) 用上、下、左、右箭头按钮使新菜单项缩进到与它要隔开的菜单项同级。
- (3) 在“标题”文本框中输入一个连字符。
- (4) 设置“名称”属性。

4) 创建多级菜单

在基于 Windows 的应用程序中,经常会用到包含多级的级联子菜单,子菜单中的每一项都代表一个程序命令或下一个子菜单,所创建的每个菜单可以包含最多 5 级子菜单。

在菜单编辑器中,在菜单标题下的菜单控件之下缩进的任何菜单控件都属于子菜单控件。一般来说,子菜单控件可以包括子菜单项、分隔条和子菜单标题,与普通的菜单在外观上没有区别。



要创建子菜单,首先创建作为子菜单标题的菜单项,这与上一级菜单类似,然后再创建子菜单中的各个菜单项,接着单击右箭头按钮将它们缩进,即建立了一个子菜单。如图 6-7 所示是一个子菜单在菜单编辑器中的显示情况。



图 6-7 子菜单在菜单编辑器中的显示情况

从编辑器中可以看到,顶层菜单有“三角函数”和“统计函数”,“三角函数”下面有 sin 和 cos 两个函数,其中 sin 函数又包括 4 个象限的计算子项,这 4 个象限构成了 sin 的子菜单。运行效果如图 6-8 所示。



图 6-8 子菜单在运行时的显示情况

可以根据需要建立不同层次的子菜单,但一般的程序中子菜单不要太多,否则显得杂乱无章,会降低程序的易用性。

6.1.3 快捷菜单

快捷菜单也叫作弹出式菜单,是独立于菜单栏而显示在窗体上的浮动菜单。在 Microsoft Windows 中,可以通过右击来激活快捷菜单。通常情况下,菜单既可以显示为弹出式菜单,又可以显示为窗体主菜单的一部分,任何含有菜单项的菜单都可以作为快捷菜单来显示。

显示快捷菜单可使用 PopupMenu 方法,其语法格式如下。

```
[Object. ]PopupMenu MenuName[,Flags[,x[,y[,BoldCommand]]]]
```

可以使用 Flags 参数在 PopupMenu 方法中定义快捷菜单的属性。表 6-1 中定义了快捷



菜单属性的 Flags 值, X 和 Y 是位置标识。

表 6-1 Flags 参数的菜单位置标识

Flags 参数	值	描述
vbPopupMenuLeftAlign	0	弹出式菜单左对齐于 X
vbPopupMenuCenterAlign	4	弹出式菜单居中于 X
vbPopupMenuRightAlign	8	弹出式菜单右对齐于 X
vbPopupMenuLeftButton	0	仅能通过鼠标左键选择快捷菜单的菜单项
vbPopupMenuRightButton	2	可以通过鼠标左键或右键来选择菜单的菜单项

注: 如果同时选择不相互冲突的几个选项, 可以使用 Or 操作符进行连接

PopupMenu 方法中的 BoldCommand 参数用来指定在显示的弹出式菜单项中以加粗字体显示其名称。

当用 PopupMenu 方法激活弹出式菜单后, 要等到用户选择了弹出式菜单中的某个选项之后(这种情况下, 还要先运行被选择菜单项的相应代码)或单击消除弹出式菜单后, 后面的代码才能继续执行。需要注意的是, 一次只能激活一个弹出式菜单, 当已经有一个弹出式菜单被激活或已经有一个下拉菜单被打开时, 对 PopupMenu 方法的调用将被忽略。

当用户在窗体中右击时, 以下代码显示一个名为 mnuFile 的菜单及该菜单下的各级菜单项。可用 MouseUp 或 MouseDown 事件来检测何时进行了右击, 直到菜单中被选取一项或取消这个菜单时, 调用 PopupMenu 方法后面的代码才会运行, 代码如下。

```
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button=vbRightButton Then      '检查是否进行了右击
        PopupMenu mnuFile            '把文件菜单显示为一个弹出式菜单
    End If
End Sub
```

PopupMenu 方法是 Visual Basic 6.0 内置的方法, 使用起来方便、简单, 但该方法也有不足之处, 当弹出菜单后, 如果不选择菜单项而在其他位置单击, 则菜单并不会在新的单击处显示。这虽然不至于影响应用程序的功能, 但从某种意义上来说降低了程序的易用性。

除了利用 Visual Basic 6.0 内置的 PopupMenu 方法来实现弹出式菜单外, 还可以利用 Windows 应用程序接口(API)函数中的菜单函数来实现弹出式菜单。利用应用程序接口的菜单函数实现弹出式菜单相对使用 PopupMenu 方法要复杂一些, 但可以弥补 PopupMenu 方法的不足之处。

与菜单有关的 API 函数有 GetMenu、GetSubMenu、GetMenuItemInfo、InsertMenuItem、CreatePopupMenu 和 DestroyMenu。下面简单介绍这几个函数的用法。

(1) GetMenu 函数。

语法格式: GetMenu(HWnd)

函数功能: 返回指定窗口菜单的句柄, 其中, hWnd 是该窗口的句柄。



(2) GetSubMenu 函数。

语法格式: GetSubMenu(hMenu, nPos)

函数功能: 返回下拉菜单或被指定菜单项激活的子菜单的句柄, 其中, hMenu 是菜单句柄, nPos 是定义激活下拉菜单或子菜单的菜单项在所给菜单中的相对位置(起始为 0)。

(3) GetMenuItemInfo 函数。

语法格式: GetMenuItemInfo(hMenu, uItem, fByPosition, lpmi)

函数功能: 返回菜单项的信息, 参数说明如表 6-2 所示。

表 6-2 GetMenuItemInfo 函数参数的含义

参 数	取 值	含 义
hMenu	—	包含相应菜单项的菜单句柄
uItem	—	欲获得其信息的菜单项的标识号或位置, 该参数的具体意义由 fByPosition 参数的值来决定
fByPosition	—	用来定义 uItem 参数的意义, 如果此参数的值为 False, 则 uItem 参数为菜单项的标识号, 否则为菜单项的位置
lpmi	指向 MenuItemInfo 结构变量的指针, 该变量指定了需要返回菜单项的信息, 并接受这些信息。MenuItemInfo 结构各元素说明如下	
	CbSize	结构占用的字节数
	Fmask	设置欲获得结构中哪些元素的值
	Ftype	菜单项类型
	Fstate	菜单项状态
	WID	标识菜单项的 16 位值
	Hsubmenu	与菜单项关联的下拉菜单或子菜单的句柄
	HbmpChecked	菜单项被选中时, 显示在菜单项旁边的位图的句柄, 该值为 Null 时使用默认位图
	HbmpUnchecked	菜单项未被选中时, 显示在菜单项旁边的位图的句柄, 该值为 Null 时不使用位图
	DwTypeData	菜单项文字, 长度在元素中给出
Cch	菜单项文字长度	

(4) InsertMenuItem 函数。

语法格式: InsertMenuItem(hMenu, uItem, fByPosition, lpmi)

函数功能: 在菜单的指定位置插入一个新菜单项, 其参数含义如表 6-3 所示。

表 6-3 InsertMenuItem 函数参数的含义

参 数	含 义
hMenu	新菜单项将被插入的菜单的句柄
uItem	新菜单项将被插入在其之前的菜单项的标识号或位置, 该参数的具体意义由 fByPosition 参数的值来决定



(续表)

参 数	含 义
fByPosition	用来定义 ulItem 参数的意义,如果此参数的值为 False,则 ulItem 参数为菜单项的标识号,否则为菜单项的位置
lpMii	指向 MenuItemInfo 结构变量的指针,该变量包含了新菜单项的信息

(5)CreatePopupMenu 函数。

语法格式:CreatePopupMenu()。

函数功能:生成一个下拉菜单、子菜单或弹出式菜单,菜单在初始时是空的,可以使用 InsertMenuItem 函数来插入菜单项。

(6)DestroyMenu 函数。

语法格式:DestroyMenu(hMenu)。

函数功能:清除指定的菜单并释放该菜单所占用的内存空间。

6.1.4 创建菜单控件数组

VB将菜单项视为控件,也存在控件数组的概念,菜单控件数组主要用于动态地增删菜单项,同时可以简化编程,用一段代码处理多个菜单项。

每个菜单控件数组元素都由唯一的索引值进行标识,这个唯一的索引值在菜单编辑器上的“索引”属性框中指定。当一个控件数组成员识别一个事件时,Visual Basic 将其索引属性值作为一个附加的参数传递给事件过程,这样就起到一一对应的关联关系。事件过程必须包含核对 Index 属性值的代码,以正确判断正在使用的控件。

在菜单编辑器中创建菜单控件数组的步骤如下。

- (1)选取窗体并打开菜单编辑器。
- (2)在“标题”文本框中输入第一个菜单标题的文本。
- (3)在“名称”文本框中输入菜单控件的名称。
- (4)在下一个缩进级中通过设定“标题”和“名称”来创建将成为数组中第一个元素的菜单项。
- (5)将数组中第一个元素的“索引”设置为 0,通常第一个元素的索引值都要设置为 0。
- (6)继续创建其他菜单项,索引值依次以 1 为步长递增。
- (7)对于数组中的后续元素重复上述操作。

菜单控件数组的各元素在菜单控件列表框中必须是连续的,而且必须在同一缩进级上。创建菜单控件数组时,要把在菜单中出现的分隔符条也包括进去。按照表 6-4 可以设计一个使用菜单控件数组的多级菜单,设计出来的菜单如图 6-9 所示。

表 6-4 使用菜单控件数组设计的多级菜单

标 题	名 称	索 引	说 明
文本设置	Menu	—	主菜单项 1
字体设置	Font	—	子菜单项 11



(续表)

标题	名称	索引	说明
宋体	FontName	0	子菜单项 111
楷体	FontName	1	子菜单项 112
仿宋	FontName	2	子菜单项 113
黑体	FontName	3	子菜单项 114
风格设置	Style	—	子菜单项 12
粗体	FontStyle	0	子菜单项 121
斜体	FontStyle	1	子菜单项 122
下划线	FontStyle	2	子菜单项 123



图 6-9 使用菜单控件设计的菜单

6.1.5 菜单的有效性控制

在运行应用程序的过程中,在某特定状态下,某个菜单或菜单项颜色变成浅灰色,且不能响应键盘或鼠标的操作,即不能运行,这就涉及菜单的有效性控制问题。

与按钮的有效性相同,菜单项也有有效与无效两种状态。处于无效状态时,菜单项以浅灰色显示,不能响应用户的任何操作。一些菜单项只有在满足一定条件时才有效。例如,在 Visual Basic 中的“编辑”菜单中的“粘贴”菜单项,只有在当前剪贴板中有内容时,该选项才有效,否则无效,并以浅灰色显示;在记事本应用程序中,如果没有选中任何文本,则“编辑”菜单中的“复制”菜单项变为浅灰色,不能被使用。在新打开的记事本中,“编辑”菜单下的菜单项几乎都是浅灰色的,不能被使用,当在记事本中输入一段文字,选取其中的部分内容并执行“复制”命令后,“编辑”菜单下的菜单项就变成可操作状态,以此说明系统进行了可操作性控制,以避免非法操作。

1) 有效性控制

菜单控件也有 Enabled 属性,该属性用来设置菜单在程序运行时是否有效。在设计阶段,该属性可以在“菜单编辑器”对话框中进行设置(见图 6-6),其中,“有效”复选框用来设置其初始状态是否有效。在默认情况下,菜单控件的 Enabled 属性的值为 True,即有效,如果要定义其为无效,那么在“菜单编辑器”对话框中就取消选中“有效”复选框,此时其 Enabled 属性值为 False,菜单命令无效,不能响应单击动作,同时快捷键的访问也无效。



如果希望把应用程序整个“编辑”菜单设置为无效,那么可以通过以下代码进行设置。

```
mnuEdit.Enabled=False
```

要将其恢复为有效,只需要将上述代码中的 False 改为 True 就可以了,如果将整个菜单设置为无效,那么该菜单下的所有菜单项均是无效的。如果只将某个菜单的特定菜单项设置为无效,如将“编辑”菜单中的“复制”菜单项设置为无效,那么可以通过以下简单代码进行设置。

```
mnuEditCopy.Enabled=False
```

2) 可见性控制

类似于上述有效性操作,还可以定义可见性,某个菜单项如果被设置为无效,仅是颜色变成灰色,但还是可见的,通过设置其 Visible 属性的值可以定义其是否可见,读者也可以在菜单编辑器中,通过“可见”复选框设置菜单项的可见性。

在运行时,如果要使一个菜单控件可见或不可见,可以在代码中进行如下设置。

```
mnuFileArray(0).Visible=True '使控件可见
```

```
mnuFileArray(0).Visible=False '使控件不可见
```

当菜单控件不可见时,该控件同时也是无效的。因为该控件通过菜单、访问键或快捷键都无法访问。如果菜单标题不可见,则该菜单上的所有控件均无效。

3) 显示菜单控件上的复选标志

在菜单项的操作过程中,经常会发现菜单项前有“√”标记(复选标志),以显示是否选中该项,这个操作可以通过设置 Checked 属性来实现,复选标志的作用主要是表示“打开”或“关闭”条件的状态,还可以用来指示正在起作用的模式。

在运行时要从一个菜单控件上增加或删除复选标志,可以从代码中设置它的 Checked 属性,这一点与前面有效或无效的设定方法完全相同。

4) 运行时添加/删除菜单控件

运行时可以根据需要添加/删除菜单项,该操作是通过加载/卸载菜单控件数组元素来实现的,因此创建菜单控件时必须使用控件数组。

【例 6-1】 设计程序,在“减”菜单下增加一级菜单项,并在主菜单上增加主菜单项,如图 6-10 所示,在“减”菜单下单击各菜单项可删除相应的菜单项。(本例题主要练习菜单的动态添加与删除)

(1) 建立程序界面,各菜单项属性设置如表 6-5 所示,创建菜单控件数组 xiangmu 和 del。

表 6-5 程序设计属性设置

标 题	名 称	索 引	说 明
加	jia	—	主菜单 1
减	jian	—	主菜单 2
无须帮助	nohelp	—	子菜单项 21
项目 0	del	0	子菜单项 22



(续表)

标 题	名 称	索 引	说 明
项目	xiangmu	0	主菜单 3
帮助	help	—	主菜单 4
帮助文档	helpfile	—	子菜单项 41
1	first	—	子菜单项 411
2	second	—	子菜单项 422

程序运行界面如图 6-10 所示。



图 6-10 动态添加/删除菜单程序运行界面

(2) 在“加”菜单项的 jia_Click 事件中添加以下程序代码。

```
Public num As Integer          '定义公共变量 num
Private Sub jia_Click()
    Load xiangmu(num)
    Load del(num)
    xiangmu(num).Caption="新增项目"& num
    del(num).Caption="删除项目"& num
    xiangmu(num).Visible=True
    del(num).Visible=True
    num=num+1
End Sub
```

新增 3 个项目的程序运行界面如图 6-11 所示。

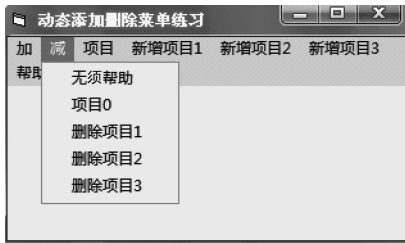


图 6-11 新增 3 个项目的程序运行界面

(3) 在“减”菜单下的 del_Click 事件中添加以下程序代码。

```
Private Sub del_Click(Index As Integer)
```



```
If Index <> 0 Then      '单击“删除项目 1”即可删除“新增项目 1”，以下类似
    Unload xiangmu(Index)
    Unload del(Index)
ElseIf Index=0 Then    '单击“项目 0”，提示不可删除
    MsgBox("不能删除我!")
End If
End Sub
```

(4)单击 xiangmu 控件数组中的各控件时,弹出提示信息,添加以下程序代码。

```
Private Sub xiangmu_Click(Index As Integer)
    MsgBox("我是项目"& Index)
End Sub
```

6.2 对话框及其应用

在基于 Windows 的应用程序中,对话框被用来提示用户提供应用程序继续运行所需的数据或向用户显示信息,是用户界面不可缺少的一部分,为用户和应用程序的交互提供了接口。

对话框是一种特殊类型的窗体对象,可以使用 MsgBox 或 InputBox 函数来创建预定义对话框,也可以使用 CommonDialog 控件创建标准对话框,如“保存”、“另存为”和“打印”等对话框。

Windows 系统中的对话框主要有 3 种,分别是模式对话框、非模式对话框和系统模式对话框。模式对话框允许用户将焦点转移到其他应用程序上,但是不允许将焦点转移到该对话框的父窗口上;非模式对话框既允许将焦点转移到其他应用程序上,也允许用户将焦点转移到其父窗口上;系统模式对话框既不允许将焦点转移到其父窗口上,也不允许将焦点转移到其他应用程序上。

1) MsgBox 函数

在程序执行过程中,需要交互的相关信息很多,用来提示程序进程和系统状态等,MsgBox 函数是经常使用的一种提示方法。MsgBox 函数用来获得“是”或“否”的响应,并显示简短的消息,如错误、警告等。之后可通过一个按钮来关闭该对话框。MsgBox 函数的语法格式如下。

```
MsgBox(prompt[, buttons] [, title] [, helpfile, context])
```

其中的一些选项说明如下。

(1)prompt。prompt 是字符串表达式,用来保存显示在对话框中的消息。prompt 的最大长度不超过 1 024 个字符,如果 prompt 的内容超过一行,则可以在行之间用回车符(Chr(13))、换行符(Chr(10))或是回车与换行符的组合(Chr(13)&Chr(10))将行分隔开来,或用 vbCrLf 进行换行操作。读者可以运行以下代码来体会换行的操作。

①使用 vbCrLf 进行换行操作的运行结果如图 6-12 所示,其代码如下。



```
Private Sub Form_Load()
```

```
MsgBox"以热爱祖国为荣,以危害祖国为耻;"& vbCrLf &_
    "以服务人民为荣,以背离人民为耻;"& vbCrLf &_
    "以崇尚科学为荣,以愚昧无知为耻;"& vbCrLf &_
    "以辛勤劳动为荣,以好逸恶劳为耻;"& vbCrLf &_
    "以团结互助为荣,以损人利己为耻;"& vbCrLf &_
    "以诚实守信为荣,以见利忘义为耻;"& vbCrLf &_
    "以遵纪守法为荣,以违法乱纪为耻;"& vbCrLf &_
    "以艰苦奋斗为荣,以骄奢淫逸为耻。"
```

```
End Sub
```

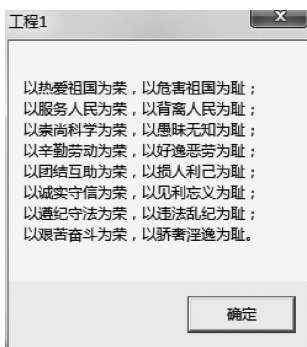


图 6-12 用 vbCrLf 进行换行操作

②不用 vbCrLf 进行换行控制的运行结果如图 6-13 所示,代码如下。

```
Private Sub Form_Load()
```

```
MsgBox"以热爱祖国为荣,以危害祖国为耻;"&_
    "以服务人民为荣,以背离人民为耻;"&_
    "以崇尚科学为荣,以愚昧无知为耻;"&_
    "以辛勤劳动为荣,以好逸恶劳为耻;"&_
    "以团结互助为荣,以损人利己为耻;"&_
    "以诚实守信为荣,以见利忘义为耻;"&_
    "以遵纪守法为荣,以违法乱纪为耻;"&_
    "以艰苦奋斗为荣,以骄奢淫逸为耻。"
```

```
End Sub
```

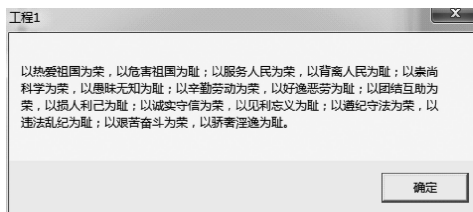


图 6-13 不用 vbCrLf 进行换行操作

③用 Chr(13)&Chr(10)控制换行的运行结果如图 6-14 所示,代码如下。

```
Private Sub Form_Load()
MsgBox"以热爱祖国为荣,以危害祖国为耻;"& Chr(13)&_
    "以服务人民为荣,以背离人民为耻;"& Chr(10)&_
    "以崇尚科学为荣,以愚昧无知为耻;"& Chr(13)& Chr(10)&_
    "以辛勤劳动为荣,以好逸恶劳为耻;"&_
    "以团结互助为荣,以损人利己为耻;"&_
    "以诚实守信为荣,以见利忘义为耻;"&_
    "以遵纪守法为荣,以违法乱纪为耻;"&_
    "以艰苦奋斗为荣,以骄奢淫逸为耻。"
End Sub
```

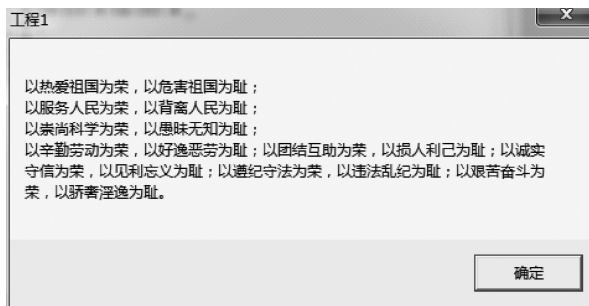


图 6-14 使用 Chr(13)&Chr(10)控制换行的效果

(2)buttons。buttons 是数值表达式值的总和,用于指定显示按钮的数目、形式,使用的图标样式等,它是一个可选项,默认值为 0,具体如表 6-6 所示。

表 6-6 Buttons 参数

常 数	值	描 述	备 注
vbOKOnly	0	只显示 OK 按钮	描述了对话框中显示的按钮类型与数目
vbOKCancel	1	显示 OK 及 Cancel 按钮	
vbAbortRetryIgnore	2	显示 Abort、Retry 及 Ignore 按钮	
vbYesNoCancel	3	显示 Yes、No 及 Cancel 按钮	
vbYesNo	4	显示 Yes 及 No 按钮	
vbRetryCancel	5	显示 Retry 及 Cancel 按钮	
VbCritical	16	显示 Critical Message 图标	描述了图标的样式
VbQuestion	32	显示 Warning Query 图标	
VbExclamation	48	显示 Warning Message 图标	
VbInformation	64	显示 Information Message 图标	



(续表)

常 数	值	描 述	备 注
vbDefaultButton1	0	第一个按钮是默认值	说明哪一个按钮是默认值
vbDefaultButton2	256	第二个按钮是默认值	
vbDefaultButton3	512	第三个按钮是默认值	
vbDefaultButton4	768	第四个按钮是默认值	
vbApplicationModal	0	应用程序强制返回;应用程序一直被挂起,直到用户对消息框作出响应才继续工作	决定消息框的强制返回性
vbSystemModal	4 096	系统强制返回;全部应用程序都被挂起,直到用户对消息框作出响应才继续工作	
vbMsgBoxHelpButton	16 384	将 Help 按钮添加到消息框	—
VbMsgBoxSetForeGround	65 536	指定消息框窗口作为前景窗口	—
vbMsgBoxRight	524 288	文本为右对齐	—
vbMsgBoxRtlReading	1 048 576	指定文本应为在希伯来和阿拉伯语系统中的从右到左显示	—

(3)title。title 是在对话框标题栏中显示的字符串表达式,也是可选项,如果省略,则将应用程序名放在标题栏中。

(4)helpfile。helpfile 是字符串表达式,用来识别向对话框提供上下文相关帮助的帮助文件,它也是可选项,如果提供了 helpfile,则必须提供 context。

(5)context。context 是数值表达式,由帮助文件的作者指定给适当的帮助主题的帮助下上下文编号,它也是可选项,如果提供了 context,则必须提供 helpfile。

【例 6-2】 根据表 6-2 中的 buttons 参数,体验不同参数的运行效果。

新建一个工程,在窗体中依次添加 6 个按钮,按钮的 Caption 设置如图 6-15 所示。

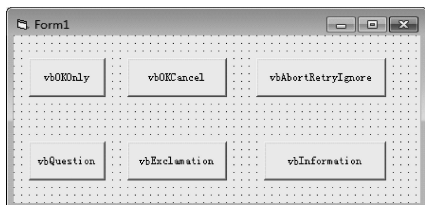


图 6-15 各个按钮的 Caption 属性

(1)双击 vbOKOnly 按钮,编写如下响应代码。

```
Private Sub Command1_Click()
    MsgBox"只显示 OK 按钮!",vbOKOnly
End Sub
```

单击这个按钮将打开如图 6-16 所示的对话框。

(2) 双击 vbOKCancel 按钮, 编写如下响应代码。

```
Private Sub Command2_Click()  
    MsgBox" 显示 OK 和 Cancel 按钮!", vbOKCancel  
End Sub
```

单击这个按钮将打开如图 6-17 所示的对话框。



图 6-16 只显示 OK 按钮



图 6-17 显示 OK 和 Cancel 按钮

(3) 双击 vbAbortRetryIgnore 按钮, 编写如下响应代码。

```
Private Sub Command3_Click()  
    MsgBox" 显示 Abort, Retry, Ignore 按钮!", vbAbortRetryIgnore  
End Sub
```

单击这个按钮将打开如图 6-18 所示的对话框。

(4) 双击 vbQuestion 按钮, 编写如下响应代码。

```
Private Sub Command4_Click()  
    MsgBox" 这个是问号图标!", vbQuestion  
End Sub
```

单击这个按钮将打开如图 6-19 所示的对话框。



图 6-18 显示 Abort、Retry 和 Ignore 按钮

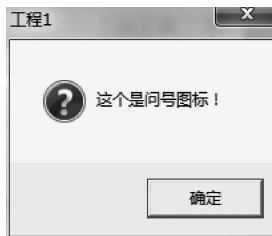


图 6-19 显示问号图标

(5) 双击 vbExclamation 按钮, 编写如下响应代码。

```
Private Sub Command5_Click()  
    MsgBox" 这个是惊叹号图标!", vbExclamation  
End Sub
```

单击这个按钮将打开如图 6-20 所示的对话框。

(6) 双击 vbInformation 按钮, 编写如下响应代码。

```
Private Sub Command6_Click()
```




```
MsgBox"这个是信息图标!",vbInformation
End Sub
```

单击这个按钮将打开如图 6-21 所示的对话框。



图 6-20 显示重要信息图标

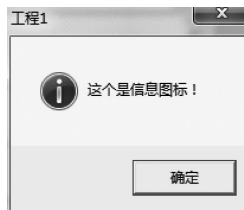


图 6-21 显示信息图标

MsgBox 函数的返回值如表 6-7 所示。

表 6-7 MsgBox 函数的返回值

常 数	值	按 钮	常 数	值	按 钮
vbOK	1	确定	vbCancel	2	取消
vbAbort	3	放弃	vbRetry	4	重试
vbIgnore	5	忽略	vbYes	6	是
vbNo	7	否	—	—	—

2) InputBox 函数

在程序运行过程中,经常需要与用户进行交互,如需要输入某些信息等。虽然消息框也可以让用户作出选择,但供选择的内容是指定的,不允许输入指定内容之外的信息。因此,必须用其他方式获取信息,可以用 InputBox 函数实现在对话框中显示提示、等待用户输入文本或单击按钮并返回文本框内容等操作。

InputBox 函数的语法格式如下。

```
InputBox(prompt[,title][,default][,xpos][,ypos][,helpfile,context])
```

InputBox 函数各参数的含义如表 6-8 所示。

表 6-8 InputBox 函数的参数含义

参 数	描 述
prompt	作为消息显示在对话框中,是一个字符串表达式,其最大长度为 1 024 个字符。如果 prompt 中包含多行,则可在各行之间用回车符(Chr(13))、换行符(Chr(10))或回车符与换行符的组合(Chr(13)&Chr(10))来分隔各行
title	显示在对话框标题栏中的字符串表达式,若省略,则标题栏将显示应用程序的名称
default	显示在文本框中的字符串表达式,在没有其他输入时作为默认的响应值。如果省略 default,则文本框为空
xpos	用于指定对话框的左边缘与屏幕左边缘的水平距离。如果省略 xpos,则对话框会在水平方向居中
ypos	用于指定对话框的上边缘与屏幕上边缘的垂直距离

(续表)

参 数	描 述
helpfile	用于标识为对话框提供上下文相关帮助的帮助文件,是一个字符串表达式
context	用于标识由帮助文件的作者指定给某个帮助主题的上下文编号,是一个数值表达式

【例 6-3】 用 InputBox 函数输入一个圆的半径,然后用 MsgBox 函数输出圆的面积。

(1)在 Form 上设置一个按钮,按钮的 Caption 属性设置为“请输入圆的半径”,如图 6-22 所示。

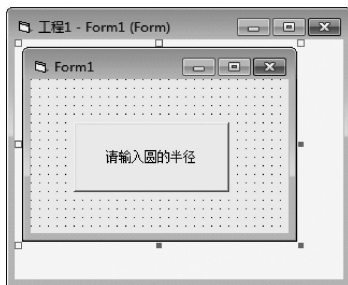


图 6-22 带按钮的界面设计

(2)双击这个按钮,在 Command1_Click()事件过程中输入以下代码。

```
Private Sub Command1_Click()
    Dim r As Single 'r 为圆的半径
    r = Val(InputBox("请输入圆的半径:", "输入"))
    MsgBox "当半径为 "& r & " 时圆的面积为: "& r * r * 3.14
End Sub
```

运行程序,在弹出的如图 6-23 所示的界面中输入圆的半径 50 并单击“确定”按钮,弹出如图 6-24 所示的对话框。

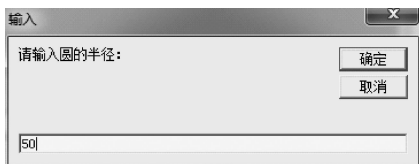


图 6-23 提示输入圆的半径

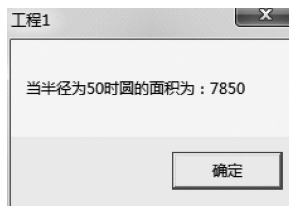


图 6-24 MsgBox 函数输出计算结果

6.3 多文档窗体设计

在实际应用程序中,一个主程序窗口下通常包含一个或多个文档界面,多个文档界面随着主窗口的关闭而关闭,也可以在不关闭主程序窗口的情况下关闭单个文档界面。下面主



要介绍多文档界面的设计和实现。

Windows 应用程序的用户界面主要有两种样式：单文档界面 (single document interface, SDI) 和多文档界面 (multi document interface, MDI)。值得注意的是, 单文档界面并不是指只有一个窗体的界面, 而是指应用程序的各窗体之间是相互独立的, 它们在屏幕上独立显示、移动、最小化或最大化, 而与其他窗体无关。

单文档界面程序只能打开一个文档, 想要打开另一个文档, 必须先关闭已打开的文档。典型的单文档编辑工具是 Windows 系统自带的记事本。

对于多文档编辑软件, 如 Office 系列办公软件中的 Word、Excel、PowerPoint 等, 都可以同时打开多个不同的文档。需要注意的是, 多文档界面由多个窗体组成, 这些窗体不是独立的, 其中有一个窗体称为多文档的父窗体, 其他窗体称为多文档的子窗体。多文档的子窗体的活动范围不能超出父窗体范围, 即不能将其移动到多文档父窗体之外。

通常情况下, 在运行期间如果应用程序只与一个文档有关联, 则建议使用单文档, 如果应用程序能与多个文档关联并进行处理, 这时就应该使用多文档。

6.3.1 MDI 应用程序的建立

要创建 MDI, 首先需要为应用程序创建一个 MDI 窗体。执行“工程”→“添加 MDI 窗体”菜单命令, 打开“添加 MDI 窗体”对话框, 在“新建”选项卡中选择“MDI 窗体”选项, 单击“打开”按钮即可在当前工程中创建一个 MDI 窗体, 如图 6-25 所示。

此时在工程资源管理器中增加了一个 MDI 窗体的图标, 如图 6-26 所示。值得注意的是, 一个应用程序只能创建一个 MDI 窗体。如果工程已经有了一个 MDI 窗体, 那么“工程”菜单中的“添加 MDI 窗体”菜单项就会变成灰色, 即不可再创建。

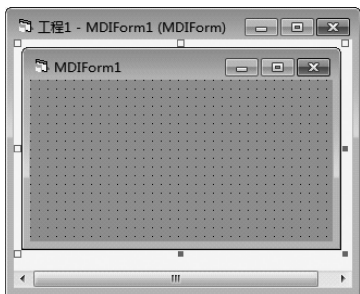


图 6-25 新建一个多文档窗体 图 6-26 在工程资源管理器中增加一个 MDI 窗体图标

创建好 MDI 窗体后, 还可以根据需要在工程资源管理器中创建一个 MDI 子窗体或打开一个存在的窗体, 然后把它的 MDIChild 属性设置为 True, 这样该窗体就成为 MDI 窗体的子窗体了。同理, 用户可以添加更多的子窗体, 下面在此基础上添加子窗体 Form1 和 Form2, 如图 6-27 所示。

当程序中有多个窗体时, 需要设置一个启动窗体。执行“工程”→“工程 1 属性”菜单命令, 打开如图 6-28 所示的“工程 1-工程属性”对话框, 切换到“通用”选项卡, 在“启动对象”下拉列表框中包括 Form1、Form2、Sub Main 以及 MDIForm1 等选项, 选择 MDIForm1 选项, 便将 MDI 窗体设置为启动窗体, 如图 6-28 所示。

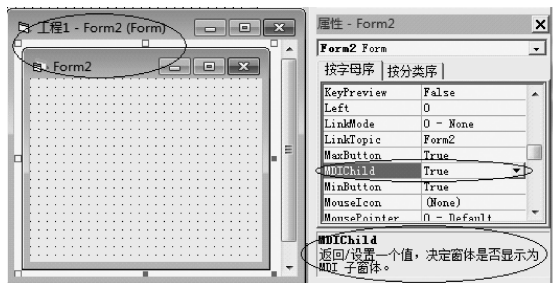


图 6-27 添加子窗体

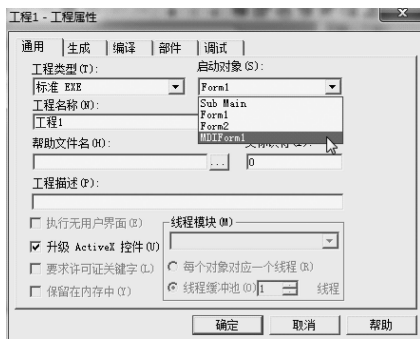


图 6-28 设置启动窗体

下面进行一个测试,响应主窗口的 MDIForm_Load 事件,用于显示子窗口,代码如下。

```
Private Sub MDIForm_Load()  
    Form1.Show  
    Form2.Show  
End Sub
```

运行该程序,可以看到多文档程序的运行效果,如图 6-29 所示。用户可以尝试移动 Form1 或 Form2,结果是子窗体都不能超出 MDIForm1 的边界,拖动 MDIForm1 的边界可以改变其大小。

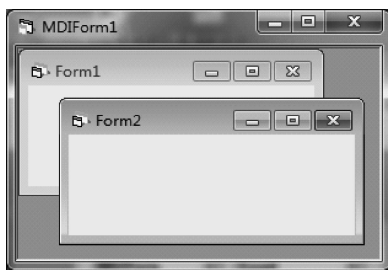


图 6-29 多文档程序测试运行结果

6.3.2 MDI 窗体应用举例

前面介绍了 MDI 窗体、普通窗体和 MDI 子窗体 3 个窗体概念。MDI 子窗体与普通窗体实际上没有太大区别,在 MDI 子窗体中同样可以添加控件、设置属性、编写代码等。

在加载 MDI 子窗体时,其父窗体(MDI 窗体)会自动加载并显示;而加载 MDI 窗体时,其子窗体不会自动加载。

MDI 窗体的 AutoShowChildren 属性可用来加载隐藏状态的 MDI 子窗口,使它们处于隐藏状态,直至用 Show 方法使它们显示出来。MDI 窗体的 AutoShowChildren 属性设置如图 6-30 所示。

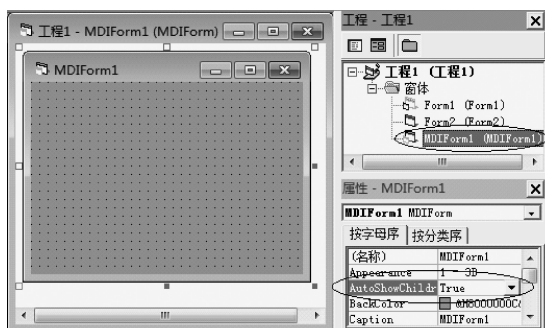


图 6-30 设置 MDI 窗体的 AutoShowChildren 属性

对于 MDI 窗体,为了使其界面友好,可能会涉及子窗体的排列问题,Visual Basic 提供了 Arrange 方法来实现对子窗体的排列,该方法有 4 个可选参数值,如表 6-9 所示。

表 6-9 Arrange 方法的相关参数

参 数	值	含 义
vbCascade	0	层叠所有非最小化的子窗体
vbTileHorizontal	1	水平平铺所有非最小化的子窗体
vbTileVertical	2	垂直平铺所有非最小化的子窗体
vbArrangeIcons	3	重排最小化子窗体的图标

用户需要退出 MDI 应用程序时,通常需要对相关信息进行保存,保存之前需要确定自上次保存以来子窗体中的数据是否有改变,这个功能可以通过在相应子窗体中定义一个公用变量来实现,如声明一个变量 ifchange,代码如下。

```
Public ifchange As Boolean
```

Text1 中的文本每改变一次,子窗体文本框的 Change 事件就会将 ifchange 设置为 True,反之设置为 False。可添加以下代码来提示自上次保存以来 Text1 的内容已经改变。

```
Private Sub Text1_Change()
```

```
    ifchange = True
```

```
End Sub
```

【例 6-4】 以学生档案管理系统的设计为例,说明菜单、工具栏、状态栏、对话框的综合应用。学生档案管理系统由 1 个 MDI 主窗体与 6 个 MDI 子窗体组成,MDI 主窗体由下拉式菜单、工具栏、状态栏组成,用户单击菜单项命令能调用相应的子窗体程序,且已打开的子窗体标题能在启动窗口主菜单项中列出,效果如图 6-31 所示。



图 6-31 学生档案管理系统主菜单设计界面



(1) 学生档案管理系统 MDI 主窗体与子窗体设计。建立 MDI 多文档学生档案管理系统主窗体以及“校历”子窗体、“学生档案编辑”子窗体、“学生档案统计”子窗体、“学生档案打印”子窗体和“系统维护”子窗体。具体操作步骤如下。

① 新建学生档案管理系统目录 C:\EX_MDI。

② 新建工程 EX_MDI.vbp, 将工程文件保存到目录 C:\EX_MDI 下, 文件名为 EX_MDI.vbp。

③ 执行“工程”→“添加窗体”菜单命令, 添加一个 MDI 窗体并设置其属性, 具体为: Name 设置为 MDIForm_XSDA; Caption 设置为学生档案管理系统; WindowsState 设置为 0-Normal。

④ 将主菜单窗体 MDIForm_XSDA.frm 作为启动主窗体, 设置方法为: 执行“工程”→“工程属性”菜单命令, 在打开的对话框中切换到“通用”选项卡, 在“启动对象”下拉列表框中选择 MDIForm_XSDA.frm 选项。

⑤ 在此例中默认生成的 Form1 窗体现在没用了, 可以将其删除。

⑥ 创建 6 个 MDI 子窗体, 其属性如表 6-10 所示, 同时设置 6 个 MDI 子窗体的 MDIChild 属性均为 True。

表 6-10 窗体属性的设置

控件名称	控件类型	属 性	设 定 值
Form1	窗体	Name	Form_XSDA1
		Caption	校历
Form2	窗体	Name	Form_XSDA2
		Caption	学生档案编辑
Form3	窗体	Name	Form_XSDA3
		Caption	学生档案查询
Form4	窗体	Name	Form_XSDA4
		Caption	学生档案统计
Form5	窗体	Name	Form_XSDA5
		Caption	学生档案打印
Form6	窗体	Name	Form_XSDA6
		Caption	系统维护

(2) 学生档案管理系统 MDI 主菜单设计。建立学生档案管理系统窗体的下拉式菜单, 其内容如表 6-11 所示, 打开工程 EX_MDI.vbp, 在工程资源管理器中选择 MDIForm_XSDA 并使用菜单编辑器建立菜单项, 然后建立校历主菜单, 在“标题”文本框中输入“校历”, 在“名称”文本框中输入 M1, 单击“下一个”按钮产生新的菜单项, 在“标题”文本框中输入“校历管理”, 在“名称”文本框中输入 M11, 单击按钮使“校历管理”成为下一级菜单, 单击“下一个”按钮产生新的菜单项; 依此类推建立所有菜单与菜单项。

菜单项名字的命名规则是: 主菜单从 M1~M7, 一级子菜单(含分隔条)是按二维数组 M(8,7)的元素 M_{ij} 方式命名的。从校历子菜单项 M11 到排列子菜单项 M74, 如表 6-11 所示。



表 6-11 学生档案管理系统窗体的菜单项标题

标 题	名 称	标 题	名 称
校历	M1	文本编辑	M51
校历管理	M11	多媒体设置	M52
退出	M12	音乐	M6
学生档案编辑	M2	多媒体播放	M61
学生档案录入	M21	窗口排列	M7
学生档案查询	M22	层叠	M71
学生档案统计	M3	平铺	M72
学生档案打印	M4	垂直	M73
系统维护	M5	排列图标	M74

(3) 建立菜单项事件过程。

① 在窗体设计器中, 执行“退出”菜单命令, 进入代码编辑器, 在其中输入以下代码。

```
Private Sub M12_Click()
    End
End Sub
```

② 编写调用“校历管理”子窗体的事件过程, 代码如下。

```
Private Sub M11_Click()
    Form_XSDA1.Show
End Sub
```

其余 5 个子窗体的调用事件过程与此类似, 读者可自行编写, 此处不再一一给出。

③ 编写“层叠”子菜单事件处理过程, 代码如下。

```
Private Sub M71_Click()
    MDIForm_XSDA.Arrange(0)
End Sub
```

“平铺”、“垂直”与“排列图标”菜单项的事件过程由读者自己编写。

(4) 学生档案管理系统主窗体中工具栏的设计。在界面上增加一个工具栏, 用来放置 calendar、write、find、stat、print、repair、edit 和 music 八个按钮, 如图 6-32 所示, 具体操作步骤如下。

① 打开所建工程 EX_MDI.vbp, 在 MDIForm_XSDA 窗体中放置一个 ToolBar 控件和一个 ImageList 控件。

② 在 ImageList 控件中添加 8 个图标的图像文件, 读者可以选择自己喜欢的图标。

③ 将 ToolBar 控件“通用”选项卡中的图像列表属性设置为 ImageList1。在 ToolBar 控件按钮页中添加 8 个按钮, 其标题分别为“校历”、“录入”、“查询”、“统计”、“打印”、“维护”、“编辑”和“音乐”, 关键字可以选择按钮的名称, 图像索引号分别为 1~8; 提示为“日历管理”、“学生档案录入”、“学生档案查询”、“学生档案统计”、“学生档案打印”、“系统维护”、“文本编辑器”和“多媒体播放器”。

④ 编写单击工具栏按钮的事件过程。



要求单击前 6 个按钮后能调用对应子窗体,并在状态栏第 1 个窗格上显示激活子窗体的标题名称。例如,单击“校历管理”按钮,则状态栏显示“激活窗体:校历管理”。事件过程的部分程序代码如下。

```
Private Sub ToolBar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    Select Case Button.Key
        Case "Date"
            Form_XSDA1.Show
            StatusBar1.Panels(1).Text = "激活窗体:校历管理"
    End Select
End Sub
```

(5) 学生档案管理系统主窗体中的状态栏设计。在界面上添加状态栏控件,用于显示当前打开的子窗体及当前日期。具体操作步骤如下。

① 在 MDIForm_XSDA 窗体上执行“工程”→“部件”菜单命令,默认打开“控件”选项卡,在列表框中选择 Microsoft Windows Common Controls 6.0 选项,然后在工具箱中双击 StatusBar 控件,将状态栏放置在窗体下方。

② 在状态栏上放置时间控件 Timer,并设置其 Interval 属性为 1 000(系统以毫秒作为单位,1 000 ms 即为 1 s)。

③ 设置状态栏属性。打开状态栏的属性页面,在“窗格”选项卡中添加两个窗格,其“文本”属性分别设置为“激活窗体”和“日期:时间”。

④ 编写代码显示当前时间,代码如下。

```
Private Sub Timer1_Timer()
    StatusBar1.Panels(2).Text = "日期:时间"& Now
End Sub
```

在 Form_Load 事件中添加以下代码。

```
StatusBar1.Panels(2).Text = "日期:时间"& Now
```

程序运行界面如图 6-32 所示。



图 6-32 程序运行完整界面



6.4 习 题

1) 选择题

(1) 多窗体程序由多个窗体组成。在默认情况下执行应用程序时,总是把()指定为启动窗体。

- A. 不包含任何控件的窗体 B. 设计时的第一个窗体
C. 包含控件最多的窗体 D. 命名为 Form1 的窗体

(2) 在用菜单编辑器设计菜单时,必须输入的项是()。

- A. 快捷键 B. 标题 C. 索引 D. 名称

(3) 以下关于通用对话框的叙述错误的是()。

- A. CommonDialog1.ShowFont 显示字体对话框
B. 在文件“打开”或“另存为”对话框中,用户选择的文件名可以经 FileName 属性返回
C. 在文件“打开”或“另存为”对话框中,用户选择的文件名及路径可以经 FileName 属性返回
D. 通用对话框可以用来制作和显示“帮助”对话框

(4) 窗体从内存卸载的过程中会发生许多事件,这些事件发生的顺序是()(其中 Terminate 事件在窗体及对象的所有引用都被从内存删除后发生)。

- A. Terminate, QueryUnload, Unload B. QueryUnload, Unload, Terminate
C. QueryUnload, Terminate, Unload D. Unload, QueryUnload, Terminate

(5) 以下关于多重窗体的叙述中,正确的是()。

- A. 作为启动对象的 Main 子过程只能放在窗体模块内
B. 如果启动对象是 Main 子过程,则程序启动时不加载任何窗体,以后由该过程根据不同情况决定是否加载以及加载哪个窗体
C. 没有启动窗体,程序不能执行
D. 以上都不对

(6) 当用户将焦点转移到另一个应用程序时,当前应用程序的活动窗体将()。

- A. 发生 DeActivate 事件 B. 发生 DeActivate 和 LostFocus 事件
C. 发生 LostFocus 事件 D. DeActivate 和 LostFocus 事件都不会发生

(7) 假定有一个名为 Menu1 的菜单项,为了在运行时使该菜单项变灰(失效),应使用的语句为()。

- A. Menu1.Enabled=False B. Menu1.Enabled=True
C. Menu1.Visible=False D. Menu1.Visible=True

(8) 以下操作不能向工程添加窗体的是()。

- A. 执行“工程”→“添加窗体”命令
B. 单击工具栏上的“添加窗体”按钮
C. 右击窗体,在弹出的快捷菜单中选择“添加窗体”命令



D. 右击工程资源管理器,在弹出的快捷菜单中选择“添加”命令,然后在其子菜单中选择“添加窗体”命令

(9)为菜单命令 Run 设计一个访问键 Alt+U,正确的方法是在其标题中输入()。

- A. &Run B. R&un C. Ru&n D. Run&

(10)以下 4 个选项中正确的是()。

- A. 只能通过右击打开弹出式菜单
B. 只能使用 PopupMenu 方法打开弹出式菜单
C. PopupMenu 方法只能放在鼠标事件 MouseUp 或 MouseDown 中使用
D. 任何菜单项都可以作为弹出式菜单使用

(11)在利用通用对话框建立“打开”或“另存为”对话框时,如果需要指定列表框所列出的文件类型是文本文件(.txt 文件),则正确的描述格式是()。

- A. text(>txt)|(*.txt) B. 文本文件(.txt)|(.txt)
C. text(.txt)||(*.txt) D. text(.txt)(*.*.txt)

(12)以下叙述错误的是()。

- A. 在同一窗体的菜单中,不允许出现标题相同的菜单项
B. 在菜单的标题栏中,“&”所引导的字母指明了访问该菜单项的访问键
C. 在程序运行过程中,可以重新设置菜单的 Visible 属性
D. 弹出式菜单也在菜单编辑器中定义

(13)在窗体上添加通用对话框 CDg1,执行语句“CDg1.Action=2”可以建立一个专用对话框,在下面的选项中,与该语句等效的语句是()。

- A. CDg1.ShowOpen B. CDg1.ShowFont
C. CDg1.ShowColor D. CDg1.ShowSave

(14)假设在窗体上已经添加了通用对话框 CDg1,并设置了相关属性,在程序中执行语句“CDg1.Filter="All Files(*.*)|*. * |文本文件(*.txt)|*.txt”后,在对话框的文件列表框中出现的选项个数是()。

- A. 1 B. 2 C. 3 D. 4

(15)MdiForm 为 MDI 窗体,其中包含菜单 MunTest,Form1 为子窗体,也包含菜单 MunTest,执行下列程序代码后,单击 MunTest 菜单,输出结果为()。

其中 MdiForm 的相关代码如下。

```
Private Sub MdiForm_Load()  
    Form1.Show  
End Sub  
Private Sub MunTest_Click()  
    MsgBox"mdi"  
End Sub
```

Form1 的相关代码如下。

```
Private Sub Muntest_Click()  
    MsgBox"Child"  
End Sub
```



- A. 显示 Child 对话框
- B. 显示 MDI 对话框
- C. 显示 Child 对话框后再显示 MDI 对话框
- D. 显示 MDI 对话框后再显示 Child 对话框

(16) 下列说法正确的是()。

- A. 一个应用程序只能创建一个窗体
- B. 一个应用程序只能创建一个模块
- C. 一个应用程序只能创建一个 MDI 窗体
- D. 一个应用程序只能创建一个 MDI 子窗体

(17) 某程序执行后显示了一个 MDI 窗体和两个子窗体, 其中, 某子窗体中有一个按钮, 则执行下列程序代码后的输出结果为()。

```
Private Sub Command1_Click()  
    Print DoEvents()  
End Sub
```

- A. 0
- B. 1
- C. 2
- D. 3

(18) MDI 窗体 MDIForm1. Arrange2 的作用是()。

- A. 层叠所有非最小化 MDI 子窗体
- B. 水平平铺所有非最小化 MDI 子窗体
- C. 垂直平铺所有非最小化 MDI 子窗体
- D. 重排最小化 MDI 子窗体的图标

(19) 以下关于 MDI 子窗体在运行时特性的叙述错误的是()。

- A. 子窗体在 MDI 窗体的内部区域显示
- B. 子窗体可在 MDI 窗体的外部区域显示
- C. 当子窗体最小化时, 它的图标在 MDI 窗体内显示
- D. 当子窗体最大化时, 其标题与 MDI 窗体标题合并, 并显示在 MDI 窗体的标题栏中

(20) 若要显示一个指定窗体, 所使用的方法是()。

- A. Show
- B. Open
- C. Hide
- D. Load

(21) 为了使窗体从屏幕上消失但仍在内存中, 所使用的方法为()。

- A. Show
- B. Open
- C. Hide
- D. Load

(22) 层叠所有非最小化 MDI 子窗体的语句是()。

- A. MDIForm1. Arrange0
- B. MDIForm1. Arrange1
- C. MDIForm1. Arrange2
- D. MDIForm1. Arrange3

(23) 在 MDI 应用程序的主窗体上, 子窗体分别有各自的菜单, 运行该 MDI 应用程序并打开一个窗体后, 在 MDI 主窗体的菜单条上显示的是()。

- A. MDI 主窗体上定义的菜单
- B. MDI 子窗体上定义的菜单
- C. MDI 主窗体菜单和子窗体菜单的简单组合
- D. MDI 主窗体菜单和子窗体菜单组合到一起时, 相同的部分只出现一次

(24) 要在工程中添加一个 MDI 窗体, 可采用的方法是()。



- A. 单击工具栏上的“添加窗体”按钮
- B. 执行“工程”→“添加窗体”菜单命令
- C. 执行“视图”→“添加 MDI 窗体”菜单命令
- D. 执行“工程”→“添加 MDI 窗体”菜单命令

2) 填空题

- (1) SDI 指 _____ 界面, MDI 指 _____ 界面。
- (2) 要将一个窗体定义为一个 MDI 子窗体, 需要将其 _____ 属性设置为 True。
- (3) 最大化一个子窗体时, 它的标题会与 MDI 窗体的标题组织在一起并显示于 _____ 上。
- (4) 一个工程可以有 _____ 个 MDI 窗体, 设置子窗体必须将该窗体的 _____ 属性设置为 _____。
- (5) 如果一个窗体的 MDIChild 属性被设置为 _____, 则该窗体为子窗体。
- (6) 在 MDI 应用程序中处理菜单的一般方法是: 把始终显示的菜单控件放在 _____ 上, 而把用于窗体的菜单控件放在 _____ 上。
- (7) 在 VB 中, 整理 MDI 窗体中的子窗体一般有以下几种形式: 层叠、_____、垂直平铺及排列图标。