

高等学校计算机科学与技术应用型教材

Visual Basic
CHENGXU SHEJI
SHANGJI ZHIDAO
YU XITI JIEDA

Visual Basic

程序设计

上机指导与习题解答

主 编◎彭文艺



NLIC2970949968

高等学校计算机科学与技术应用型教材

Visual Basic 程序设计 上机指导与习题解答

主 编 彭文艺



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书是与北京邮电大学出版社出版的《Visual Basic 程序设计》一书配套的教学辅导用书。为了更好地与《Visual Basic 程序设计》教材内容相衔接,本书的章节组织与该教材相同,每章包含有学习指导、实验、例题解析和课后习题答案 4 部分。

本书可作为高等院校非计算机专业可视化程序设计相关课程的辅导书,也可作为高职高专计算机应用专业面向对象程序设计相关课程的辅导书。

图书在版编目(CIP)数据

Visual Basic 程序设计上机指导与习题解答 / 彭文艺主编. --北京:北京邮电大学出版社, 2013. 8
ISBN 978-7-5635-3664-1

I. ①V… II. ①彭… III. ①BASIC 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 206423 号

书 名: Visual Basic 程序设计上机指导与习题解答
主 编: 彭文艺
责任编辑: 王丹丹
出版发行: 北京邮电大学出版社
社 址: 北京市海淀区西土城路 10 号 (邮编: 100876)
发 行 部: 电话: 010-62282185 传真: 010-62283578
E-mail: publish@bupt.edu.cn
经 销: 各地新华书店
印 刷: 北京联兴华印刷厂
开 本: 787 mm×1 092 mm 1/16
印 张: 10.75
字 数: 264 千字
印 数: 1—3 000 册
版 次: 2013 年 8 月第 1 版 2013 年 8 月第 1 次印刷

ISBN 978-7-5635-3664-1

定 价: 22.00 元

• 如有印装质量问题, 请与北京邮电大学出版社发行部联系 •

前 言

本书是与北京邮电大学出版社出版的《Visual Basic 程序设计》一书配套的教学用书。为了更好地与《Visual Basic 程序设计》教材内容相衔接，本书的章节组织与该教材相同，每章包含有学习指导、实验、例题解析和课后习题答案 4 部分。

本书的特点如下：

(1) 实验部分包含实验内容和实验扩展内容，实验内容详细地给出了实验的步骤，有利于学生对基本理论和基本操作的掌握，而实验扩展内容则要求学生根据前面所掌握的基本理论和基本操作自己完成，从而有利于培养学生运用知识的能力。

(2) 充分考虑了学生参加全国计算机等级水平考试的需要，认真研究了全国计算机等级水平考试大纲和历年计算机等级考试的特点，选择实验和理论习题，而且最后一章中包含了最近几年全国计算机等级考试二级 Visual Basic 语言笔试试题和答案。不仅能帮助学生掌握 Visual Basic 6.0 基本内容起到良好的巩固作用，还能帮助学生顺利通过计算机等级水平考试。

本书的出版得到了华中科技大学武昌分校信息科学与工程学院领导、计算机基础教研室全体教师以及北京邮电大学出版社的大力支持和帮助，在此表示衷心的感谢！

在本书的编写过程中，参考了多部优秀教材，从中吸取了许多有益的知识，在此，谨向他们表示诚挚的谢意。由于时间仓促，加上水平有限，书中难免有不妥之处，敬请广大读者批评指正。



目 录

第 1 章 Visual Basic 开发环境	1
1.1 学习指导	1
1.2 实验	3
1.3 例题解析	8
1.4 课后习题答案	8
第 2 章 程序设计基础	9
2.1 学习指导	9
2.2 实验	14
2.3 例题解析	17
2.4 课后习题答案	19
第 3 章 顺序结构程序设计	20
3.1 学习指导	20
3.2 实验	23
3.3 例题解析	29
3.4 课后习题答案	30
第 4 章 选择结构程序设计	31
4.1 学习指导	31
4.2 实验	33
4.3 例题解析	37
4.4 课后习题答案	39
第 5 章 循环结构程序设计	40
5.1 学习指导	40
5.2 实验	42
5.3 例题解析	50
5.4 课后习题答案	53
第 6 章 数组	54
6.1 学习指导	54

6.2 实验	55
6.3 例题解析	60
6.4 课后习题答案	64
第7章 常用标准控件	65
7.1 学习指导	65
7.2 实验	74
7.3 例题解析	87
7.4 课后习题答案	91
第8章 菜单设计与多文档界面	92
8.1 学习指导	92
8.2 实验	98
8.3 例题解析	105
8.4 课后习题答案	108
第9章 过程	109
9.1 学习指导	109
9.2 实验	112
9.3 例题解析	119
9.4 课后习题答案	124
第10章 数据文件	125
10.1 学习指导	125
10.2 实验	129
10.3 例题解析	135
10.4 课后习题答案	138
第11章 历年二级考试真题	139
参考文献	164

第1章

Visual Basic开发环境

1.1 学习指导

1. 程序设计语言的分类

- (1) 机器语言。
- (2) 汇编语言。
- (3) 面向过程的程序设计语言。
- (4) 面向对象的程序设计语言。

2. Visual Basic 的特点

- (1) 可视化的开发环境。
- (2) 事件驱动的编程机制。
- (3) 友好的集成开发环境。
- (4) 结构化的程序设计语言。
- (5) 强大的数据库访问功能。
- (6) 支持 DDE、DLL 和 OLE 技术。
- (7) 完备的联机帮助功能。

3. Visual Basic 6.0 启动

Visual Basic 6.0 的启动有以下三种方式。

- (1) 依次单击“开始”→“程序”→“Microsoft Visual Basic 6.0 中文版”→“Microsoft Visual Basic 6.0 中文版”。
- (2) 使用“资源管理器”或“我的电脑”找到 Visual Basic 6.0 可执行文件所在文件夹,在文件夹中找到可执行文件 VB6.exe,双击则可启动 Visual Basic 6.0。
- (3) 用鼠标双击桌面上的 Microsoft Visual Basic 6.0 快捷图标(如果桌面上有)。

4. Visual Basic 6.0 退出

Visual Basic 6.0 的退出有以下三种方式。

- (1) 单击 Visual Basic 集成开发环境窗口标题栏中的“关闭”按钮。
- (2) 单击 Visual Basic 集成开发环境窗口“文件”菜单下的“退出”命令。
- (3) 使用组合键“Alt+F4”。

5. Visual Basic 6.0 集成开发环境组成

- (1) 标题栏:窗口最上面的一栏为标题栏,显示窗口的标题名字和工作模式,由于工作

状态的不同,内容有所不同,有设计模式、运行模式、中断模式。

(2) 菜单栏:菜单栏包括文件、编辑、视图、工程、格式、调试、运行、查询、图表、工具、外接程序、窗口、帮助 13 个菜单项,它包含了程序设计中常用的命令。

(3) 工具栏:工具栏提供了许多常用命令的快捷访问按钮,单击某个按钮,即可执行相应的操作。

(4) 窗体设计窗口:也称为对象窗口,用于应用程序的用户界面设计,通过在窗体上添加各类控件并设置相应的属性来完成窗体的设计。

(5) 工具箱:位于窗体的左侧,用于存放和显示建立 Visual Basic 6.0 应用程序所需要的各种控件,标准的工具箱包含有 20 个 Visual Basic 6.0 标准的控件和 1 个指针图标。

(6) 工程资源管理器窗口:用来管理工程中所包含的文件,可以通过“查看代码”、“查看对象”和“切换文件夹”3 个按钮查看不同的文件。

(7) 属性窗口:用于设置所选对象的属性,如大小、标题、颜色、字体等,主要由对象列表框、属性显示方式、属性列表框和属性说明 4 个部分组成。

(8) 窗体布局窗口:用于指定程序运行时窗体的初始位置。

(9) 代码编辑窗口:用于编写程序代码和修改程序代码,由对象列表框、过程列表框和代码区组成。

6. 向窗体添加控件

(1) 单击工具箱中的控件,选中控件,将鼠标移动到窗体上,此时鼠标的指针变成“+”号,把“+”号移动到窗体的适当位置,按下鼠标的左键拖动鼠标,就会在窗体上创建控件。

(2) 双击工具箱中某个需要的控件,则在窗体中会自动出现该控件后,但此方法添加的控件的大小和位置是固定的。

7. 设计 Visual Basic 应用程序的步骤

(1) 建立用户界面:根据用户界面的需要,向窗体中添加所需控件。

(2) 设置窗体和控件属性:根据用户界面的需要,设置窗体和各个控件属性。在实际的应用程序设计中,创建控件对象和设置控件属性可同时进行,即每添加一个控件后,接着就设置该控件的属性。

(3) 编写代码:根据程序的功能需要,针对窗体中各个控件所能支持的方法或事件编写相应的程序代码,实现各个功能。

(4) 程序调试和运行:检查程序的用户界面是否符合要求,程序的各个功能是否运行正确,如果不能满足要求,修改用户界面和程序代码,直到满足要求为止。

(5) 程序保存:程序运行满足要求后,为了永久地保存程序,需要对程序进行保存。在 Visual Basic 中,应用程序都是以工程的形式保存在磁盘中。一个工程通常包含了多个窗体,在保存时需要同时保存各个窗体文件(.frm)和一个工程文件(.vbp)。

8. Visual Basic 应用程序运行方式

(1) 解释运行模式可以通过单击“运行”菜单下的“启动”命令,或按 F5 功能键,或单击工具栏上的“启动”按钮来实现,系统读取触发事件的那段事件过程代码,将其转换为机器代码,然后执行该机器代码。此时转换后的机器代码不保存,如需要再次运行该程序,必须再执行一次。解释运行模式一般用于开发阶段。

(2) 编译运行模式通过单击“文件”菜单下的“生成 *.exe...”命令实现,系统读取程序中的全部代码,转换为机器代码,并保存在扩展名为.exe的可执行文件中,此文件可脱离 Visual Basic 环境运行。对调试无误的应用程序,一般使用编译运行模式。

9. 面向对象程序设计概论

(1) 对象:是指 Visual Basic 中可访问的实体。如窗体、控件、外部文件、变量等,整个应用程序也是对象。它包含有一定的属性、方法,并能对外界的事件进行响应。

(2) 类:类是同一种对象的统称,是一个抽象的整体概念。属于同一类的所有对象具有同一组属性、方法和事件,只是其属性值不同,对事件的响应不同。

(3) 属性:指对象所具有的性质、特征。如对象的名字、大小、位置和颜色等特性,它用来描述对象的状态。在设计应用程序时,通过改变对象的属性值来改变对象的外观和行为。

设置属性时格式为:对象.属性名 = 属性值

调用属性时格式为:对象.属性

(4) 方法:指对象所具有的动作和行为,是 Visual Basic 提供的用来完成特定操作的特殊子程序,是系统事先设定好的,用户只能使用,不能修改。如 Print、Hide、Show、Move 等,用户只需通过调用它们来实现一些功能。由于方法不是一个独立的实体,它的功能要有对象来体现,所以调用方法应指明是针对哪个对象的。

调用方法的一般格式为:<对象名>.方法[参数]

(5) 事件:是由 Visual Basic 预先定义的,能被对象识别和响应的动作。每个控件都可以对一个或多个事件进行识别和响应。在 Visual Basic 中对象与应用程序的往来是通过事件来进行的。例如,窗体装载时的 Load 事件、鼠标单击(Click)事件、鼠标双击(DblClick)事件等。

(6) 事件过程:对每个对象,可能会有多个事件产生,当一个对象识别到某一事件发生时,就去响应事件的程序代码段,该代码段称为事件过程。

1.2 实 验

【实验目的】

- (1) 掌握启动和退出 Visual Basic 的方法。
- (2) 熟悉 Visual Basic 集成开发环境中各窗口的作用和使用方法。
- (3) 掌握在窗体上添加控件的方法,调整控件大小和位置方法。
- (4) 掌握建立、编辑和运行一个简单 Visual Basic 程序的全过程。

【实验内容】

- (1) 启动 Visual Basic 6.0。

方法 1 操作步骤:依次单击“开始”→“程序”→“Microsoft Visual Basic 6.0 中文版”。

方法 2 操作步骤:使用“资源管理器”或“我的电脑”找到 Visual Basic 6.0 可执行文件所在文件夹,在文件夹中找到可执行文件 VB6.exe,双击则可启动 Visual Basic 6.0。

方法3 操作步骤:用鼠标双击桌面上的 Microsoft Visual Basic 6.0 快捷图标(如果桌面上有)。

(2) 退出 Visual Basic 6.0。

方法1 操作步骤:单击 Visual Basic 集成开发环境窗口标题栏中的“关闭”按钮。

方法2 操作步骤:单击 Visual Basic 集成开发环境窗口“文件”菜单下的“退出”命令。

方法3 操作步骤:使用组合键“Alt+F4”。

(3) 建立一个简易计算器的用户界面,如图 1-1 所示。

设计步骤如下:

① 启动 Visual Basic 6.0 应用程序,在弹出的“新建工程”对话框,选择“新建”选项卡下的“标准. exe”图标,单击“打开”按钮,新建一个工程。

② 单击工具箱中的命令按钮图标,该图标反相显示,表示已选该控件。将鼠标指针移到要添加控件的窗体上,鼠标的指针变为“+”号(“+”号的中心确定控件左上角的位置),将“+”号移动到窗体适当的位置,按下鼠标的左键不放,向右下拖动,直到大小合适时松开,就可在窗体上添加一个命令按钮。利用同样的方法,在窗体上再添加另外的 15 个命令按钮。

③ 单击工具箱中文本框图标,将鼠标指针移窗体上,鼠标的指针变为“+”号,将鼠标移动到适当的位置,按住鼠标的左键不放,向右下拖动,直到大小合适时松开,就会在窗体上添加一个文本框控件,如图 1-2 所示。



图 1-1 简易计算器用户界面

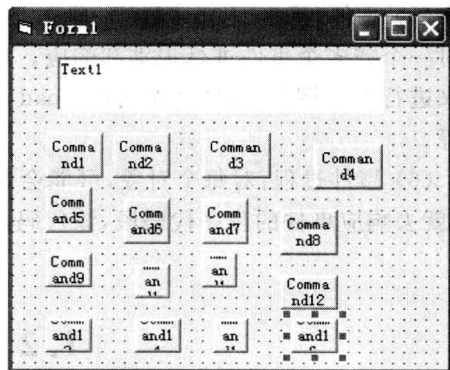


图 1-2 添加控件后的窗体

④ 按住“Shift”键不放,分别单击各个命令按钮,最后一个单击“Command1”命令按钮,使其成为“基准控件”,然后单击“格式”菜单下的“统一尺寸”菜单下的“两者都相同”菜单项。

⑤ 按住“Shift”键不放,选中 Command1、Command5、Command9、Command13 控件,最后选中 Text1 控件,并使其成为“基准控件”,然后单击“格式”菜单下的“对齐”菜单下的“左对齐”菜单项。

⑥ 按住“Shift”键不放,选中 Command4、Command4、Command12、Command16 控件,最后选中 Text1 控件,并使其成为“基准控件”,然后单击“格式”菜单下的“对齐”菜单下的“右对齐”菜单项。

⑦ 按住“Shift”键不放,选中 Command1、Command5、Command9、Command13 控件,然后单击“格式”菜单下的“垂直间距”菜单下的“相同间距”菜单项。

- ⑧ 按住“Shift”键不放,选中 Command2、Command3、Command4 控件,最后选中 Command1 控件,并使其成为“基准控件”,然后单击“格式”菜单下的“对齐”菜单下的“顶端对齐”菜单项,最后单击“格式”菜单下的“水平间距”菜单下的“相同间距”菜单项。
- ⑨ 按住“Shift”键不放,选中 Command6、Command7、Command8 控件,最后选中 Command5 控件,并使其成为“基准控件”,然后单击“格式”菜单下的“对齐”菜单下的“顶端对齐”菜单项,最后单击“格式”菜单下的“水平间距”菜单下的“相同间距”菜单项。
- ⑩ 按住“Shift”键不放,选中 Command10、Command11、Command12 控件,最后选中 Command9 控件,并使其成为“基准控件”,然后单击“格式”菜单下的“对齐”菜单下的“顶端对齐”菜单项,最后单击“格式”菜单下的“水平间距”菜单下的“相同间距”菜单项。
- ⑪ 按住“Shift”键不放,选中 Command14、Command15、Command16 控件,最后选中 Command13 控件,并使其成为“基准控件”,然后单击“格式”菜单下的“对齐”菜单下的“顶端对齐”菜单项,最后单击“格式”菜单下的“水平间距”菜单下的“相同间距”菜单项。调整好的界面如图 1-3 所示。

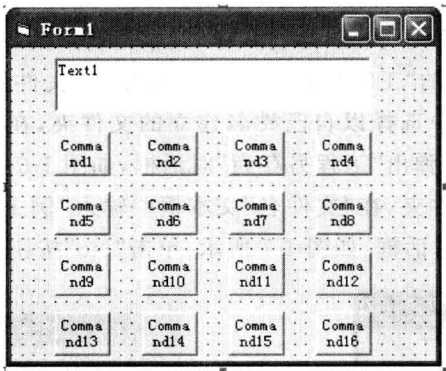


图 1-3 调整控件大小和位置后的窗体

- ⑫ 选中窗体对象,在属性窗口中找到“Caption”属性,删除默认值“Form1”,并将该属性设置为“简易计算器”。
- ⑬ 选中 Text1 对象,在属性窗口中找到“Text”属性,删除默认值“Text1”,使其属性值为空。
- ⑭ 选中 Command1 对象,在属性窗口中找到“Caption”属性,删除默认值“Command1”,并将该属性设置为“7”。
- ⑮ 按上述方法对其他 15 个命名按钮的“Caption”属性,按表 1-1 所示分别进行设置。

表 1-1 各命令按钮控件属性

对象	属性	属性值	说明
Command2	Caption	8	按钮的标题
Command3	Caption	9	按钮的标题
Command4	Caption	+	按钮的标题
Command5	Caption	4	按钮的标题
Command6	Caption	5	按钮的标题
Command7	Caption	6	按钮的标题

续表

对象	属性	属性值	说明
Command8	Caption	—	按钮的标题
Command9	Caption	1	按钮的标题
Command10	Caption	2	按钮的标题
Command11	Caption	3	按钮的标题
Command12	Caption	*	按钮的标题
Command13	Caption	0	按钮的标题
Command14	Caption	=	按钮的标题
Command15	Caption	清除	按钮的标题
Command16	Caption	/	按钮的标题

各个控件属性设置后的界面,如图 1-4 所示。

⑩ 单击工具栏上的“启动”按钮,运行程序,观察运行后的界面是否与图 1-1 相同。

⑪ 单击“文件”菜单下的“工程另存为”菜单项,弹出“文件另存为”对话框,如图 1-5 所示。在“保存在”组合框中选择以自己姓名建立的文件夹,在“文件名”文本框中输入“例 1. frm”。单击“保存”按钮,弹出“工程另存为”对话框,如图 1-6 所示。在“保存在”组合框中选择以自己姓名建立的文件夹,在“文件名”文本框中输入“例 1. vbp”。单击“保存”按钮,弹出“Source Code Control”对话框,如图 1-7 所示,单击“No”按钮,完成程序的保存。



图 1-4 设置控件属性后的窗体

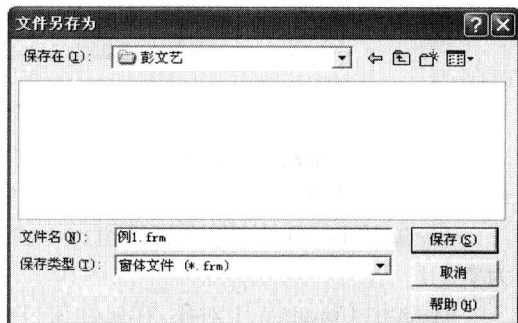


图 1-5 “文件另存为”对话框

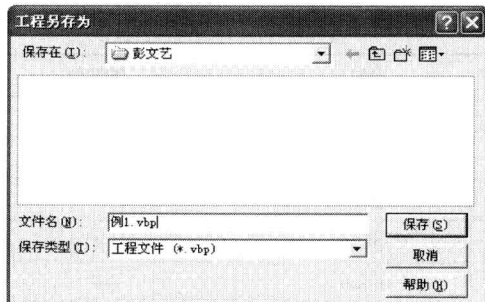


图 1-6 “工程另存为”对话框

(4) 建立一个应用程序,如图 1-8 所示,单击“显示”命令按钮,在窗体上显示一行文字“欢迎使用 Visual Basic”,单击“清除”按钮,删除窗体上显示的文字,单击“退出”按钮,结束应用程序。

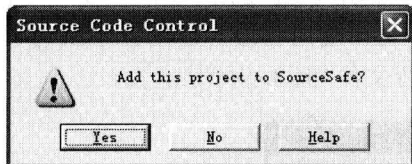


图 1-7 “Source Code Control”对话框

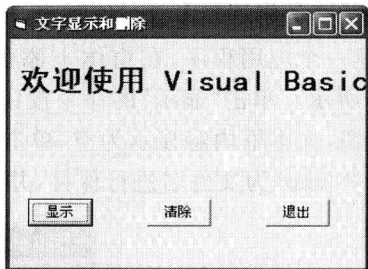


图 1-8 文字显示和删除用户界面

设计步骤如下:

- ① 建立用户界面。新建一个工程,进入窗体设计器,添加 3 个命令按钮 Command1、Command2、Command3,1 个标签控件 Label1,调整好各个控件的大小和位置。
- ② 设置控件属性。按照表 1-2 所示分别设置各个控件的属性。

表 1-2 各控件的属性设置

对象	属性	属性值	说明
Form1	Caption	文字显示和删除	窗体标题
Command1	Caption	显示	按钮标题
Command2	Caption	清除	按钮标题
Command3	Caption	退出	按钮标题
Label1	Caption	" "	标签的内容为空
	AutoSize	True	标签自动调整大小
	Font	黑体、加粗、20 号	标签中文字的字体、字形、字号

- ③ 双击“显示”命令按钮,打开代码窗口,编写该按钮单击事件,程序代码如下:

```
Private Sub Command1_Click()
    Label1.Caption = "欢迎使用 Visual Basic"
End Sub
```

- ④ 双击“清除”命令按钮,打开代码窗口,编写该按钮单击事件,程序代码如下:

```
Private Sub Command2_Click()
    Label1.Caption = ""
End Sub
```

- ⑤ 双击“退出”命令按钮,打开代码窗口,编写该按钮单击事件,程序代码如下:

```
Private Sub Command3_Click()
    End
End Sub
```

- ⑥ 单击工具栏上的“启动”按钮,运行程序。单击窗体各个命令按钮,测试是否能够实

现各个程序功能。

⑦ 单击“文件”菜单下的“工程另存为”菜单项,以“例 2. frm”和“例 2. vbp”为文件名分别保存窗体和工程。

【实验扩展内容】

建一个应用程序,在窗体上添加 3 个命令按钮和 1 个文本框,程序运行后的界面如图 1-9 所示。单击“显示”的命令按钮时,在文本框中显示“欢迎使用 Visual Basic”,单击“清除”按钮,文本框内容显示为空,单击“结束”按钮时,退出应用程序。应用程序以“例 3. frm”和“例 3. vbp”为文件名进行保存,并编译生成可执行文件“例 3. exe”。

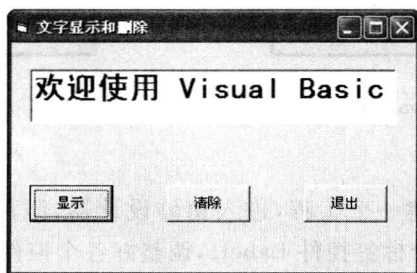


图 1-9 文字显示和删除用户界面

1.3 例题解析

选择题

以下关于 Visual Basic 特点的叙述中,错误的是()。

- A) Visual Basic 是采用事件驱动编程机制的语言
- B) Visual Basic 程序既可以编译运行,也可以解释运行
- C) 构成 Visual Basic 程序的多个过程没有固定的执行顺序
- D) Visual Basic 程序不是结构化程序,不具备结构化程序的三种基本结构

【答案】D

【解析】Visual Basic 是在 BASIC 和 Quick BASIC 语言中发展起来的,因此具有高级程序设计语言的语句结构,Visual Basic 是结构化程序。

1.4 课后习题答案

选择题

1. C 2. A 3. B 4. C 5. C 6. A 7. A 8. D 9. D 10. D

2.1 学习指导

1. 基本数据类型

Visual Basic 中的数据类型有以下几种。

(1) 整型(Integer)和长整型(Long):指不带小数点和指数符号的数。整型数据表示形式为: $\pm n[\%]$,长整型数据的表示形式为: $\pm n\&$ 。

(2) 单精度(Single)和双精度(Double):指带有小数部分的数(也称浮点数或实数)。

单精度数据有两种表示形式:① $\pm n.n[!]$ 。② $\pm n.n E \pm m[!]$ 。

双精度数据有两种表示形式:① $\pm n.n\#$ 。② $\pm n.n D \pm m\#$ 。

(3) 货币型(Currency):表示不带小数部分的数,也可以表示带有小数部分的数,表示带有小数部分的数时,小数点后最多有4位数字,当超过4位时进行四舍五入保留4位,主要用于货币计算,数据表示形式为: $\pm n[.n]@$ 。

(4) 字节型(Byte):指不带符号整数,表示的范围在0~255的整数,字节型数据在存储二进制数据时很有用,当进行ANSI码和Unicode码之间格式转换时,数据不会丢失。

(5) 逻辑型(Boolean):只有两个值 True 和 False,用来表示只有两种状态的数据。当逻辑型数据要和其他的数值型数据运算时,要将逻辑型的数据转换为其他类型的数据,其规则为:False 转换为0,而 True 转换为-1。同样也可以将其他数值型的数据转换为逻辑型数据,其规则为:0 转换为 False,而其他非0值转换为 True。

(6) 日期型(Date):用来表示日期和时间,日期型数据表示形式:用“#”将日期和时间信息括起来,例如,#2013-10-1 14:30:10#。

(7) 字符串(String):用来表示字符串,字符串可以包含字母、汉字、数字、空格、标点符号等一切可打印的西文字符和中文汉字的序列。字符串数据表示形式:用" "将字符括起来,例如,"Visual Basic 程序设计"。

(8) 变体型(Variant):是一种特殊的数据类型,也叫可变数据类型。变体型数据可以用于存储数值型、字符串型、日期型等任何类型的数据。

(9) 对象型(Object):用来表示对象,例如命令按钮控件等。

2. 常量

常量指在程序运行过程中代表的固定不变的数据,Visual Basic 的常量分两种,一种是

文字常量(也叫字面常量),另一种是符号常量。

(1) 文字常量:文字常量的类型和值由它本身的表示形式决定,包括数值常量、字符串型常量、日期型常量、逻辑型常量共 4 种。为了显示地指明数据的类型,可以在常量数据的后面加上类型说明符,这些说明符分别为:

%	整型
&	长整型
!	单精度型
#	双精度型
@	货币型
\$	字符串型

(2) 符号常量:在程序中使用常量名代表一个常量称为符号常量,使用符号常量可以增加程序的可读性和可维护性,因为在程序中看到的是有具体含义的名字,而不是具体的数值。符号常量分为两类,一类是系统内部定义的符号常量,另一类符号常量是用户自己定义的符号常量,使用前需用 Const 语句声明,在声明一个符号常量后就可以用常量名来引用其代表的常数。

声明符号常量语法为:

```
[Public|Private] Const 符号常量名[As 类型名] = 常量
```

3. 变量

变量是在程序运行过程代表的可以改变的数据,用来暂时存储程序中的数据。变量实际上是内存中的存储单元。用一个标识符来指明某个数据存放在内存中的具体位置,这种标识符称为变量名,即用变量名来标识变量。变量代表的内存单元中的实际内容就是变量值。使用变量之前,应先定义所用到的变量(包括变量名和类型),使系统分配相应的内存空间,并确定该空间可存储的数据类型,所有的变量都有名字和类型。

(1) 用类型说明语句定义变量。

用类型说明语句定义变量一般格式如下:

```
Dim|Public|Private|Static <变量名>[As <数据类型>],[<变量名>[As <数据类型>]]...
```

(2) 用类型说明符定义变量类型。

除了可以使用 As 类型子句说明变量的类型外,还可以使用类型说明符放在变量名的尾部,来说明不同类型变量。

(3) 变量的隐式声明。

在 Visual Basic 中,允许对变量进行隐式声明,即对不变量进行声明而直接引用,此变量将被默认为变体型变量。

4. 运算符和表达式

运算就是对数据进行加工和处理。基本的运算可以用简洁的符号来表示,复杂的运算一般使用函数或过程来实现,表示某种运算的符号称为运算符(Operator),被运算的对象称为运算量或操作数。表达式(Expression)是由若干个运算符和运算对象(或操作数)组成的有意义的式子,用来描述对什么数据、按什么顺序进行运行。

(1) 加、减、乘、除运算

加、减、乘、除运算符的含义与数学中表示的意义相同。但要注意表示乘法运算时，“*”不能省略。

(2) 指数运算

指数运算用来计算成方和方根，其运算符为“^”。若左操作数为正数，则右操作数可以为任何数值，如若左操作数为负数，则右操作数必须为整数。

(3) 整除运算

整除运算用来计算两个整数相除，其运算符为“\”。两个操作数必须为整数，如果不是整数，Visual Basic 首先对操作数进行四舍五入为整型数，然后进行整除运算，运算的结果也只简单地截取整数部分，小数部分直接舍掉，不作四舍五入处理。

(4) 求余运算

求余运算用来计算两个整型数相除所得的余数，其运算符为“Mod”。两个操作数必须为整型数，如果不是整型数，Visual Basic 首先对操作数进行四舍五入为整型数，然后进行求余运算，运算结果的符号取决于左操作数的符号，即左操作数为正数，所求余数的结果为正数；若左操作数为负数，所求余数的结果为负数。

(5) 算术运算符优先级

在一个表达式中需要进行运算时，每一步都按预先确定的顺序进行计算求解，这个顺序由运算符的优先级决定。

5. 字符运算符和字符表达式

连接运算符是用来连接字符串数据的运算符，Visual Basic 中提供了两个连接运算符：“&”和“+”，它们都可以用于字符串数据的连接。当运算符的操作数都是字符串数据时，它们的作用一样。连接运算符“&”与“+”的区别和使用如下。

“&”强制两个操作数作字符串连接，如果操作数不是字符串数据，则先将操作数转换字符串数据再作字符连接运算，操作数与运算符“&”间至少要加一个空格。当“+”运算符的两边的操作数都是字符串数据时，则作字符连接运算；当“+”运算符的两边的操作数都是数值型数据时，则作加法运算。如果一个操作数为字符串数据，另一个为数值型数据，则出错。但有特殊情况，如果这个字符串数据为一个纯数字的字符串数据，另一个为数值型数据，则也作加法运算。

6. 关系运算符和关系表达式

关系运算符用来对数值型、字符串、日期型数据进行比较的运算，主要有等于(=)、不等于(<>)、小于(<)、大于(>)、小于等于(<=)和大于等于(>=)运算符，所有的关系运算符优先级相同。

关系表达式是用关系运算符将常量、变量连接起来的表达式，关系表达式的结果为逻辑型数据，如果比较关系成立，结果为 True；如果比较关系不成立，结果为 False。

当进行比较关系运算时，当运算符两边的操作数据类型一致时，可直接进行比较；如果两边的操作数据类型不一致时，按一定的规则转换成同一种数据类型，再进行比较运算；如果不能转换成同一种数据类型，则出错。