



清松电脑系列丛书

Visual Foxpro 6.0

Visual Foxpro

Visual Foxpro 6.0

Visual Foxpro 6.0

张治文 何 磊
沈惠璋 吴继泽

开发实例

编著

清华大学出版社



Visual FoxPro 6.0 开发实例

张治文 何 磊 沈惠璋 吴继泽 编著

清 华 大 学 出 版 社

(京) 新登字158号

内 容 简 介

本书是一本介绍用 Visual FoxPro 6.0 中文版开发管理信息系统的书籍。全书通过对一个适合于中小型零售商业POS系统的全面分析,介绍了用 Visual FoxPro 开发信息系统的全过程。通过对本书的学习,读者既可以掌握有关 Visual FoxPro 6.0 编程的知识,也可以掌握如何用 Visual FoxPro 6.0 构造管理信息系统的框架。

书中介绍的实例是一个实用化系统的简化版本,读者可以直接扩充其功能,所有的源代码可以直接引用。本书不仅是广大 Visual FoxPro 6.0 开发人员的好助手,也是读者自学的好教材。

版权所有,翻印必究。本书封面贴有清华大学出版社激光防伪标签,无标
签者不得销售。

书 名: Visual FoxPro 6.0 开发实例

作 者: 张治文 何磊 沈惠璋 吴继泽 编著

出 版 者: 清华大学出版社(北京 清华大学校内,邮政编码: 100084)

因特网址: <http://www.tup.tsinghua.edu.cn>

责任编辑: 薛亚菲

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 17.25 字数: 409 千字

版 次: 1999 年 5 月第 1 版 2000 年 1 月第 3 次印刷

书 号: ISBN 7-302-03494-X/TP·1907

印 数: 13001—21200

定 价: 29.00 元

前 言

作为微机管理信息平台的 Visual FoxPro 又推出了6.0的中文版。该版本除了继承Visual FoxPro 以前版本的优点外,还增强了面向对象技术、ActiveX技术以及对Internet的全面支持。

本书通过一个小型商业POS系统的开发,讲述了如何在 Visual FoxPro 6.0 环境下构造管理信息系统。

通过本书介绍的应用程序表明了使用新的Visual FoxPro开发一个管理信息系统是如何简洁。首先建立一个数据库,之后建立所有表以及他们之间的关系。用表单建立数据的录入,用视图和报表建立输出,然后输入所有的菜单项。这样就可以得到一个完整的应用程序。

本书是为初学 Visual FoxPro 6.0 以及从过程转向对象编程的读者而写的。虽然没有介绍全部的编程内容,但学习本书后,读者会觉得用新方法开发各种管理信息系统是如此轻松和快捷。

作为初学者要从第 1 章开始学习,如果读者对Visual FoxPro 程序设计有一定的基础,可以从第 2 章开始学习。

如果读者希望就 Visual FoxPro 6.0 方面的问题同我们探讨,可以通过电子邮件联系。我的电子邮件地址为:

zxjzhang@public.east.cn.net

作 者

1999 年 3 月

目 录

第 1 章 Visual FoxPro 6.0 编程基础	(1)
1.1 Visual FoxPro 6.0 的新增功能	(1)
1.2 Visual FoxPro 6.0 的数据类型	(3)
1.3 常量与变量	(4)
1.3.1 常量	(4)
1.3.2 变量及变量赋值命令	(5)
1.4 表达式	(6)
1.4.1 运算符	(6)
1.4.2 表达式	(10)
1.5 函数	(11)
1.5.1 函数的基本用法	(11)
1.5.2 函数的嵌套	(12)
1.6 Visual FoxPro 命令	(12)
1.7 程序设计基础	(13)
1.7.1 顺序设计	(13)
1.7.2 分支设计	(14)
1.7.3 循环程序设计	(18)
1.8 过程调用	(24)
1.8.1 过程的概念和调用	(24)
1.8.2 内部过程	(26)
1.8.3 内存变量及过程中的参数传递	(28)
1.8.4 自定义函数	(29)
第 2 章 使用项目管理器	(31)
2.1 管理信息系统概述	(31)
2.2 用 Visual FoxPro 6.0 开发信息系统的基本步骤	(33)
2.3 范例信息系统的特点	(33)
2.4 项目管理器概述	(33)
2.5 建立新项目	(34)
2.6 “项目管理器”的使用	(36)
2.6.1 在“项目管理器”中找到需要的文件	(36)
2.6.2 “数据”选项卡	(37)

2.6.3	“文档”选项卡	(37)
2.6.4	其余的选项卡	(38)
2.6.5	文件的创建及修改	(38)
2.6.6	为文件添加说明	(39)
2.6.7	文件的添加及删除	(39)
2.6.8	浏览表文件中的内容	(40)
2.6.9	项目间的文件共享	(41)
2.6.10	“项目管理器”的折叠与展开	(41)
第3章	建立数据库	(43)
3.1	创建数据库	(43)
3.1.1	创建一个新数据库	(43)
3.1.2	在数据库中添加表	(44)
3.1.3	从数据库中移去表	(45)
3.2	创建表	(46)
3.2.1	建立数据库表	(46)
3.2.2	创建字段的默认值	(47)
3.2.3	设置输入掩码	(48)
3.2.4	设置字段的标题	(49)
3.3	表的索引	(50)
3.3.1	创建索引	(50)
3.3.2	删除索引	(51)
3.4	建立表之间的关系	(52)
3.4.1	创建永久关系	(52)
3.4.2	删除表间的永久关系	(53)
3.4.3	建立临时关系	(53)
3.5	设计实例	(53)
3.5.1	员工表	(54)
3.5.2	商品表	(54)
3.5.3	客户表	(55)
3.5.4	供应商表	(55)
3.5.5	销售表	(56)
3.5.6	库存表	(57)
第4章	用表单输入数据	(58)
4.1	面向对象程序设计的语法	(58)
4.2	设计表单	(59)
4.3	创建表单	(60)
4.3.1	新建一个表单	(60)

4.3.2	设置表单的数据环境	(61)
4.3.3	设置表单的属性	(62)
4.3.4	编辑表单的事件代码和方法程序代码	(64)
4.4	在表单中添加对象	(64)
4.5	设计实例	(75)
4.5.1	销售表单	(75)
4.5.2	大宗客户销售表单	(92)
4.5.3	商品管理表单	(111)
4.5.4	客户管理表单	(124)
4.5.5	供应商管理表单	(136)
4.5.6	进货表单	(151)
4.5.7	操作员管理	(164)
4.5.8	登录表单	(174)
4.5.9	按金额分类统计表单	(179)
4.5.10	按数量分类统计表单	(189)
4.5.11	日统计表单	(198)
4.5.12	月统计表单	(201)
4.5.13	库存统计表单	(203)
第 5 章	用视图查询数据	(207)
5.1	创建视图	(207)
5.1.1	视图设计	(207)
5.1.2	创建视图	(208)
5.2	使用视图	(211)
5.3	设计实例	(212)
5.3.1	pos_jefl 视图	(212)
5.3.2	pos_kctjl 视图	(213)
5.3.3	pos_rtjl 视图	(213)
5.3.4	pos_ytjl 视图	(214)
5.3.5	xsdyl 视图	(215)
第 6 章	设计报表	(216)
6.1	设计报表	(216)
6.1.1	确定报表布局	(216)
6.2	创建报表	(216)
6.3	修改报表	(219)
6.4	预览和打印报表	(222)
6.4.1	预览结果	(222)
6.4.2	打印报表	(223)

6.5	实例分析	(224)
6.5.1	按金额分类统计报表	(224)
6.5.2	库存统计报表	(224)
6.5.3	销售日统计报表	(225)
6.5.4	按数量分类统计报表	(226)
6.5.5	销售单报表	(226)
6.5.6	月销售统计报表	(227)
第7章	设计菜单	(228)
7.1	创建菜单系统	(228)
7.2	规划菜单系统	(229)
7.3	创建菜单、菜单项和子菜单	(229)
7.3.1	创建菜单	(230)
7.3.2	创建菜单项	(231)
7.3.3	创建子菜单	(231)
7.3.4	菜单项分组	(231)
7.3.5	在应用程序中包含菜单	(232)
7.4	为菜单系统指定任务	(232)
7.4.1	指定访问键	(232)
7.4.2	指定键盘快捷键	(233)
7.4.3	启用和禁用菜单项	(234)
7.4.4	为菜单或菜单项指定任务	(235)
7.4.5	向菜单系统添加初始化代码	(236)
7.4.6	向菜单系统添加清理代码	(237)
7.4.7	在运行时控制菜单	(237)
7.5	测试与调试菜单系统	(240)
7.6	定制菜单系统	(241)
7.6.1	显示状态栏信息	(241)
7.6.2	定义菜单标题的位置	(241)
7.6.3	保存与还原菜单	(242)
7.6.4	设置菜单系统	(242)
7.7	实例分析	(242)
7.7.1	销售菜单	(243)
7.7.2	进货菜单	(244)
7.7.3	统计与分析菜单	(244)
7.7.4	维护菜单	(245)
7.7.5	打印菜单	(246)
7.7.6	退出菜单	(247)

第 8 章 设计主控程序	(249)
8.1 设置起始点	(249)
8.2 建立主控程序	(250)
8.2.1 初始化环境	(250)
8.2.2 显示初始的用户界面	(250)
8.2.3 控制事件循环	(251)
8.2.4 恢复初始的开发环境	(251)
8.3 设计实例	(252)
第 9 章 编译与发程序	(253)
9.1 连编项目文件	(253)
9.2 连编可执行程序	(254)
9.3 “连编选项”对话框中选项的作用	(256)
9.4 设置可执行程序的版本信息	(256)
9.5 可执行程序独立运行所需的支持文件	(258)
9.6 创建发布磁盘	(258)

第 1 章 Visual FoxPro 6.0 编程基础

Visual FoxPro 6.0 中文版是一套微机上的关系数据库管理系统，由于它具有强大的功能和可视的开发环境，因此受到广大开发人员的喜爱，成为国内应用群最多的数据库管理系统。Visual FoxPro 的命令格式类似于英语的日常用语，可读性好，它的每一基本指令又可派生出多条命令。整个命令系统提供了处理大型、复杂数据库系统的能力，利用这些命令可以开发出大型的管理信息系统。

本章是为没有 Visual FoxPro 编程基础的用户而写的，如果读者已有用 Visual FoxPro 进行编程的经验，可以从第 2 章开始阅读。

1.1 Visual FoxPro 6.0 的新增功能

在 Visual FoxPro 6.0 中文版中除了继承以前版本的性能外，还增添了许多新的功能，利用这些功能可以轻松地设计出各种基于网络应用的各种管理程序。这些新增或改进的功能如下：

Access 和 Assign 方法程序

利用这两种用户自定义的方法程序可在查询或试图更改属性的值时执行所需的代码，这样就能够为类或对象创建公共接口。将其界面与执行的操作分离开来，不仅能够方便地检查属性值的有效性，还可以轻而易举地保护 ActiveX 控件子类的属性。

Active Document

Active Document 是一种基于 Windows 的、嵌入在浏览器中的非 HTML 应用程序，它提供了在浏览器界面中访问该应用程序功能的途径。Visual FoxPro 的 Active Document 应用程序与其他的 Visual FoxPro 应用程序一样，都可以运行表单、报表和标签，将类实例化，运行代码及操作数据，而且 Active Document 应用程序还具有自身的优势，即能包容在像 Internet Explorer 这样的 Active Document 容器中。

组件管理库

“组件管理库”是一个新工具，可将类库、表单、按钮等对象进行分组并组成对象、项目、应用程序或其他分组。这些可视的分组是可动态定制的，这样就能在组件管理库中的多个分类之间使用、复制或重新组织组件。“组件管理库”也包含新的 Visual FoxPro 的基本类，这些基本类提供了“即时应用”功能。

代码范围分析器应用程序

代码范围应用程序可记录文件中的哪些代码行被执行了。而分析应用程序则提供了哪些行被真正执行、一行代码被执行的次数及所耗费的时间等信息。代码范围模式和分析模式使开发者能找出应用程序中存在问题的区域，特别是能够找到被跳过的代码以及性能瓶颈。Visual FoxPro 的“代码范围分析器”提供了可以使用并进行定制的代码范围引擎和一个多窗口应用程序，用于分析程序和项目。

对 GIF 和 JPEG 图形的支持

经过改进的 Visual FoxPro 能够支持 GIF (Graphics Interchange Format) 和 JPEG (Joint Photographic Electronic Group) 图形文件格式，进一步加强了对 Internet 互操作性的支持。

HTML 格式的帮助

HTML 格式的帮助是 Microsoft 所提供的用于创建适应 Internet 时代要求的帮助文件的解决方案。Visual Studio 中带有 HTML Help Workshop 工具，可帮助用户为自己的 Visual FoxPro 应用程序创建和发布基于 HTML 格式的帮助文件。

新的和改进的语言元素

将 Visual FoxPro API 库 Foxtools.fll 中的许多有关文件名操作的函数也添加到 Visual FoxPro 6.0 中，简化了编程工作。

OLE 拖放

Visual FoxPro 现在支持 OLE 拖放，这是一个强大而有用的工具，允许用户在其他支持 OLE 拖放的应用程序（例如 Visual Basic, Windows Explorer, Microsoft Excel 和 Word）之间移动数据。在一个由用户开发并发布的 Visual FoxPro 应用程序中，也可以在该应用程序的控件之间移动数据，或者在控件和其他支持 OLE 拖放的 Windows 应用程序之间移动数据。

项目管理器挂接程序

在以前版本的 Visual FoxPro 中，访问项目的唯一途径是像处理 .dbf 文件一样，处理 .pjx 文件或窗口标题为基础的“项目管理器”窗口。为了能以编程方式访问“项目管理器”，我们采用了 Project 对象关联模型。该模型由项目、Project 对象和相应的 ProjectHook 对象组成。当通过执行 CREATE PROJECT, MODIFY PROJECT 和 BUILD APP 等命令打开一个项目时，都将实例化相应的 Project 对象。Project 对象是打开项目和 Visual FoxPro IDE 之间的媒介，并且允许开发者直接与项目交互。Project 对象也将实例化一个 ProjectHook 对象，用于通知发生的事件。

自动服务程序 (Automation Server) 的改进

Visual FoxPro 提供了经过改进而更可靠的自动服务程序, 这些自动服务程序能更好地与 Microsoft Transaction Server, Microsoft Visual Basic 和 Active Server Pages 这样的产品和技术协同工作。

新增和改进了的向导和生成器

除了对已有的几个向导进行了改进之外, Visual FoxPro 6.0 还带有新的向导和生成器, 帮助用户生成应用程序、创建数据库、在 Web 上发布数据、建立对象模型, 以及创建自己的向导等工作。

对 2000 年日期的支持

在 Visual FoxPro 6.0 中改进了与日期操作有关的命令和函数, 这样在用户的应用程序代码中将支持 2000 年日期。

1.2 Visual FoxPro 6.0 的数据类型

在数据库系统中记载信息的符号组合称为数据, 人们用数据来描述实体的对象及其属性。例如, 对于一个中学生来讲, 可以用学生编号、各学科的成绩来进行描述, 这里的学生就是对象, 而学生编号、各学科的成绩就是描述对象的属性。

数据类型是简单数据的基本属性, 是一个重要的概念, 因为只有相同类型的数据之间才能直接运算, 否则就会发生数据类型不匹配的错误。

Visual FoxPro 是一种关系型数据库管理软件, 在关系型数据库中把描述每一实体集合的数据表示成一张二维表。例如, 表示商品信息的一张二维表如表1.1所示。

表 1.1 商品信息

记录号	商品编码	商品名称	供应商编码	进货单价	售货单价
1	0128	笔座	100001	9.00	11.00
2	0100	小号胶棒	200012	8.00	10.00
3	0122	圆珠笔	100041	4.10	5.20
4	2222	自动铅笔	100022	2.30	3.50
5	3333	抗菌圆珠笔	300022	3.60	4.80
6	4444	杂志盒	100321	67.20	84.00
7	5555	涂色小白兔	100341	1.12	1.40

这样, 我们就可以从表1-1中看出商品信息中共有 7 种商品的数据, 每一横行共有5个字段组成, 分别描述了记录的结构, 也就是指实体的属性。包括: 商品编码、商品名称、供应商编码、进货单价、售货单价等信息。

在Visual FoxPro 6.0中对于表结构的字段提供了13种类型, 对他们的具体描述如表1.2所示。

表 1.2 Visual FoxPro 6.0的字段类型

数 据 类 型	说 明
字符型	字母、数字型文本
货币型	货币单位
数值型	整数或小数
浮动型	整数或小数
日期型	年, 月, 日
日期时间型	年, 月, 日, 时, 分, 秒
双精度型	双精度数值
整型	不带小数点的数值
逻辑型	真或假
备注型	不定长的字母、数字文本
通用型	OLE (对象链接与嵌入)
字符型 (二进制)	字母数字型文本, 但是当代码页更改时字符值不变
备注型 (二进制)	不定长的字母数字文本, 但是当代码页更改时备注并不改变

虽然在Visual FoxPro 6.0 中定义了13 种字段类型, 但是数据类型只有5种。分别是: 字符型数据、数字和浮点数字型数据、日期型数据、逻辑型数据和通用型数据。字段属数据库文件所特有, 而数据既可作为数据库文件中的字段内容, 也可以作为内存变量内容或作常量使用。

1.3 常量与变量

常量是在程序执行过程中不改变其值的数据, 而变量是在程序执行过程中其内容可能不断变化的数据。FoxPro定义了常量、内存变量、系统变量和字段变量。

1.3.1 常量

FoxPro定义了4种类型的常量, 分别是: 数值型常量、字符型常量、逻辑型常量和日期型常量。

数值型常量: 可以是整数或实数, 例如: 89 和 34.56 都是数值型常量。

字符型常量: 是用定界符括起来的由字符、空格和数字所组成的字串。定界符可用单引号、双引号或方括号。当某一种定界符本身是字符型常量的组成部分时, 就应选用另一种定界符。例如: "小白兔牙膏"、" "、"999.99"等都是写法正确的字符型常量, 而MSD、'abcde'fg'hi' 都是写法不正确的字符型常量。

逻辑型常量：该常量只有“真”与“假”两个值。用.T.、.t.表示“真”，用.F.、.f.表示“假”。

注意：两边的小圆点不能丢掉。

日期型常量：日期型常量必须用大括号括起来，例如：{^1998.10.12}和{^1998/06/12}都是日期型常量的正确写法。

1.3.2 变量及变量赋值命令

FoxPro定义了用户内存变量和系统内存变量。用户内存变量简称为内存变量，其名称用字母、下划线和数字来表示，但必须以字母打头；而系统内存变量名称由系统规定。

内存变量

内存变量是数据库结构之外独立存在于内存中的变量，一般随程序运行结束或退出FoxPro而释放。内存变量常用于存储常量、程序运行的中间结果，也用于存储控制程序执行的各种参数。

在一般情况下，内存变量的类型取决于所存储的数据类型。若存储的数据类型发生了变化，它的类型也随之改变。FoxPro定义4种类型内存变量，即字符型内存变量、数字型内存变量、逻辑型内存变量和日期型内存变量。最多允许定义65 000个内存变量。

此外，FoxPro还允许用户定义一维和二维内存变量数组，变量数组的元素类型同样由所赋数据的类型确定，如所赋数据的类型变化，其数组元素类型也随之改变。

系统内存变量

系统内存变量是Visual FoxPro自动生成和维护的变量，用于控制Visual FoxPro的输出和显示信息的格式。为与一般内存变量相区别，在系统内存变量名前加一下划线“_”。例如，系统内存变量“_PEJECT”用于设置打印输出时的走纸方式，系统缺省值是“BEFORE”，即打印前走纸换页，用户可将其设置为“NON”，即打印前不换页走纸。

为变量赋值

Visual FoxPro提供多种命令来定义内存变量和给内存变量赋值，常用的赋值命令格式如下：

STORE <表达式> TO <内存变量表> | <变量数组>

和

<内存变量> | <变量数组> = <表达式>

这两个命令都用于定义内存变量并给变量赋值，不同之处是前一个命令可定义多个内存变量并赋以同一数据，而后一个命令却只能定义单个内存变量。对于内存变量数组二者是完全相同的。

例如，在Visual FoxPro的命令窗口中键入如下内容来为相应的变量赋值：

```
a=12.34
b="小白兔牙膏"
c=.t.
d={^1998/10/21}
```

实际上就是定义了4个不同类型的内存变量，分别是：a，b，c和d，并且分别赋予了数值、字符、逻辑和日期4种类型的数据。然后在命令窗口中通过键入输出命令就可以看到变量的内容，例如键入“? a”，可以看到如图1.1所示的结果。

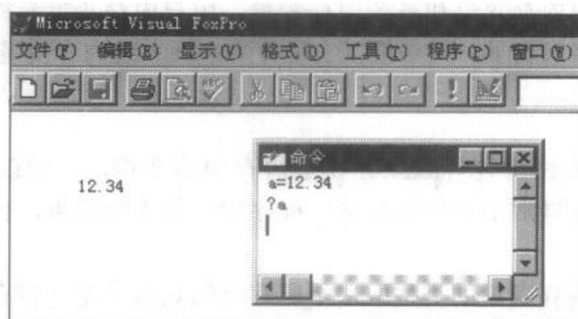


图 1.1

如果要为几个内存变量赋予相同的值，可以使用 store 命令，举例如下：

```
store 2 to a, b, c
```

则表示将2同时赋予a，b，c三个内存变量。

1.4 表达式

表达式是 FoxPro 命令的重要组成部分，它是由常量、变量、函数及运算符连接而成的有意义的式子。

1.4.1 运算符

根据运算符的作用不同划分为 4 类：算术运算符、关系运算符、逻辑运算符和字符串连接运算符，下面做详细介绍。

算术运算符

算术运算符主要用于数字型数据间的算术运算，其运算结果也是数字型数据。运算

符的优先级排序为：括号、单目运算、乘幂、乘除、求余、加减，其中括号优先级最高，加减运算符优先级最低。算术运算符包括以下几种，如表 1.3 所示。

表 1.3 算术运算符

符 号	说 明
()	分组优先运算符
**、^	乘幂运算符
*/	乘除运算符
%	余数运算符
+、-	加减运算符或单目运算符

例如，用户在命令窗口中键入两条显示运算结果的命令：

? 4^2+5
? 4*(2*3)+5

其显示在屏幕上的结果值分别为：21 和 29。

关系运算符

关系运算符用于比较运算，即比较同类型表达式的值，如果其值使关系成立，则比较运算的结果取逻辑值“真”，否则取逻辑值“假”。关系运算符包括以下几种，如表 1.4 所示。

表 1.4 关系运算符

符 号	说 明
<	小于
>	大于
=	相等比较
==	字符串全等比较
<=	小于或等于
>=	大于或等于
<>、#、!=	不等于
\$	包含运算

下面分别为变量赋值并执行相应的比较运算，从表 1.5 可以看到命令及执行的结果。

表 1.5 关系运算举例

命 令	说 明	执行结果	说 明
a=23.34	定义数字型变量		
b=27.45	定义数字型变量		
? a<b	数值比较	.T.	关系成立, 结果为真
? a>b	数值比较	.F.	关系不成立, 结果为假
x="a"	定义字符型变量		
y="b"	定义字符型变量		
? x>y	比较字符 a 和 b 的 ASCII 码	.F.	a 的 ASCII 码小于 b, 关系不成立, 结果为假
? x<y	比较字符 a 和 b 的 ASCII 码	.T.	a 的 ASCII 码小于 b, 关系成立, 结果为真
? "bc"\$"abcd"	比较\$左边字串是否是其右边字串的子串	.T.	是其子串, 结果为真
? "ac"\$"abcd"	比较\$左边字串是否是其右边字串的子串	.F.	不是其子串, 结果为假
? "文件"\$"新建文件"	比较\$左边字串是否是其右边字串的子串	.T.	是其子串, 结果为真
? "abcd"="abc"	比较等号右边字串是否是其左边字串的子串	.T.	是其子串, 结果为真
? "abc"="abcd"	比较等号右边字串是否是其左边字串的子串	.F.	不是其子串, 结果为假
? "abcd"=="abc"	字串等长并完全相同比较	.F.	不等长, 结果为假

逻辑运算符

逻辑运算符用作表达式之间的逻辑运算, 参加运算的表达式应有逻辑值, 其运算结果也是逻辑量。逻辑运算符的优先级从高到低的次序为: 逻辑非、逻辑与、逻辑或。逻辑运算符有如下 3 种, 如表 1.6 所示。

表 1.6 逻辑运算符

运 算 符 号	说 明
. NOT.	逻辑非
. AND.	逻辑与
. OR	逻辑或

逻辑运算符的意义是:

A.AND.B 只当 A 和 B 都为真时, 逻辑运算的结果才取真值, 否则取假值。

A.OR.B 当 A 和 B 中至少有一个取真值时, 逻辑运算结果取真值。

.NOT.A 当 A 取真值时, 结果取假值, 反之结果取真值。

例如, 我们分别为变量 a、b、c 赋予下列的值:

a=.f.

b=4