

经济数学

程序设计引论

PROGRAMMING
DESIGN

瓦龙德 著

economy

Mathematics

贵州教育出版社

图书在版编目(CIP)数据

经济数学程序设计引论/瓦龙德著. - 贵阳:
贵州教育出版社, 2001.6
ISBN 7-80650-172-X

I. 经… II. 瓦… III. ①经济数学-计算机
辅助计算-程序设计②BASIC 语言-程序设计
IV. F224-39

中国版本图书馆 CIP 数据核字(2001)第 031485 号

经济数学程序设计引论

瓦龙德 著

出版发行 贵州教育出版社
社 址 贵阳市中华北路 289 号 (邮编 550001)
印 刷 贵阳云岩通达彩印厂
开 本 787×1092 毫米 1/16
印张字数 11 印张 267 千字
印 数 1—1180 册
版次印次 2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷

书 号: ISBN 7-80650-172-X/F·7

定 价: 17.00 元

如发现印、装质量问题, 影响阅读, 请与印刷厂联系调换。

厂址: 贵阳市百花山路 22 号 电话: 6763625 邮编: 550003

序 言

经济学中的数学问题,主要是数值计算问题,往往涉及复杂的运算。用手工处理速度慢、效率低,甚至难以完成;用计算机处理不仅可以大大缩短运算时间,而且对于没有公式解的问题,也可以先去寻找一个收敛于所求答案的无限过程,然后按照这个过程执行一定的步数,求出近似解。在实际工作中,由于所有计算都要用有限十进小数来表示,所以从原则上说,计算结果大都带有近似的性质,近似解并不意味着“次等解”。

作者撰写本书的目的在于探索一条如何运用经济数学方法、改革经济数学教学的路子。当前国内有关经济数学和程序设计的书籍大都各行其道,有的专门研究数学问题,有的专门讨论程序设计;本书将二者结合起来,针对经济学中的数值问题,用计算机程序设计方法加以解决。

用计算机去解决经济数学问题,必须寻找便于机器计算的算法。本书吸取了传统计算数学中的优秀成果,选取了迭代法、追赶法和消元法等看似平常,实际上既通用而又有效的方法,并在原来的基础上加以革新与改进,如对高斯消去法计算行列式,增加了误差调整;对迭代法解线性方程组,在迭代过程中判别收敛性。

作者在本书中提出了一些新的观点。如函数作图问题,本来是研究函数性状的工具,但是在微积分的专著和教程中,为了作好函数图形,往往要先用微分学的方法求出函数的单调区间、极值点、拐点……,然后再用描点连线的方法去作图,使之成为一件十分复杂、费时的任务。本书提出:计算机具有运算速度快、计算精度高的特点,不必用微分学的方法帮助函数作图,可以编写程序直接计算出足够多点的坐标,连接出相当精确的函数曲线,借助图象再去研究函数曲线的单调区间、极值点、拐点……,从而使函数作图能够真正有助于函数性状的研究,把颠倒的过程重新颠倒回来。

本书选用的开发平台为 Visual BASIC 6.0,这是一种面向对象的、结构化的现代程序设计语言,该语言功能强大、简单易用。本书全部程序均已调试通过,其中大多数程序都可以方便地改变参数,使一个程序不仅能解决一个问题,而且能解决一大类问题,当参数改变时无须另行编写新的程序。例如运行程序 1.3,可以求出 20 类带有任意参数的常用经济函数的值;运行程序 6.5,可以判定任意线性方程组解的情况,并求出其解。

本书包括有关函数、极限、级数、数值导数、数值积分、行列式、矩阵、 n 维向量、线性方程组等经济数学问题的程序设计方法,每个程序都包含功能、算法说明、对象及属性、窗体界面表、程序说明、程序代码(含详细注释)、运行效果等内容。在程序设计中突出解决数值问题的基本方法,界面力求简朴,不过多涉及枝节问题和专门技巧。

限于水平,书中多有不足之处,热诚欢迎专家、读者不吝指正。

作者 2001 年 3 月

Email:walongde@21cn.com

录

序 言

第一章 函数	(1)
第一节 函数值的计算	(1)
第二节 函数作图	(24)
第二章 极限	(45)
第一节 数列极限与函数极限	(45)
第二节 级数	(53)
第三节 数值导数	(59)
第三章 数值积分	(66)
第一节 变步长梯形法	(66)
第二节 变步长抛物线法	(73)
第四章 行列式	(77)
第一节 高斯主元素消去法	(77)
第二节 改进消元法	(87)
第五章 矩阵与 n 维向量	(95)
第一节 矩阵的线性运算	(95)
第二节 矩阵乘法	(103)
第三节 逆矩阵	(110)
第四节 矩阵的秩	(117)
第五节 n 维向量组的秩与线性相关性	(123)
第六章 线性方程组	(130)
第一节 用主元素消去法解线性方程组	(130)
第二节 用迭代法解线性方程组	(144)
第三节 齐次线性方程组的基础解系	(151)
第四节 一般线性方程组	(161)

第一章 函 数

第一节 函数值的计算

经济数学要研究各式各样的函数。在 Visual BASIC 程序设计语言中,对于简单的问题可以直接用赋值语句算出函数值;对于比较复杂的问题,可以编写一个程序段或者通过自定义函数去求函数值。

程序 1.1 收益函数

一 功能

已知生产 x 件产品的总收入 $R(x)$ 为

$$R(x) = 200x - 0.01x^2$$

求当 x 分别为 10、100、500 … 时, $R(x)$ 的值。

二 算法说明

1. 本程序的问题比较简单,可以用赋值语句求出函数值。
2. 如果直接按照所给的表达式进行计算,需要做 2 次乘法、1 次乘方。现在改用公式:

$$R(x) = (200 - 0.01x)x$$

只需做 2 次乘法,不必做乘方。变化虽然很简单,然而提高了运算速度,具有普遍意义。

三 对象及属性表

对象名称	属性	值
窗体 Form1	Caption	收益函数求值
标签 Label1	Caption	函数表达式
Label2	Caption	请输入自变量的值: $x =$
lblJg	Caption	(清空)
文本框 txtZbl	Text	(清空)
按钮 cmdRun	Caption	运算
cmdExit	Caption	退出
OLE 控件 OLE1	Class	Equation 3(已知函数的表达式)

四 窗体界面

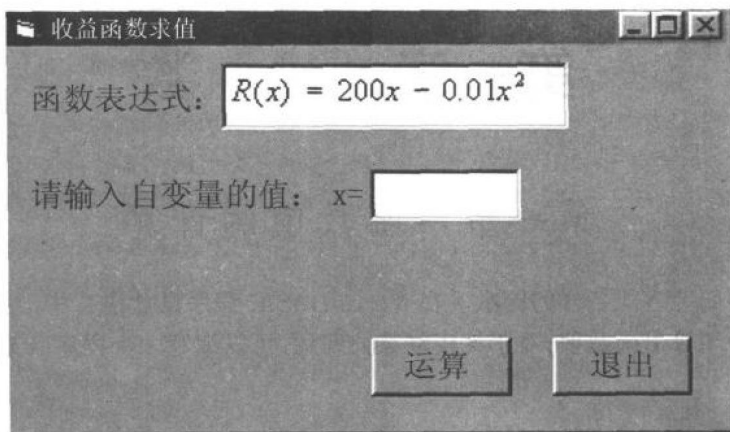


图 1-1

五 程序说明

1. 变量 x 表示产品的数量, r 表示总收入。
2. 运行程序时, 在文本框中输入的数据属于“字符型”, 使用函数 `Val` 后, 转换为“数值型”, 才可以参加算术运算求出 r 的值。

六 程序代码

```
Option Explicit  
Dim x As Single  
Dim r As Single
```

```
Private Sub cmdRun_Click()
```

```
    ' 运算
```

```
        x = Val(txtZbl.Text)
```

```
        r = (200 - 0.01 * x) * x
```

```
        lblJg.Caption = "运算结果:R(" + Str(x) + ")=" + Str(r)
```

```
End Sub
```

```
Private Sub cmdExit_Click()
```

```
    ' 退出
```

```
        End
```

```
End Sub
```

七 运行效果

本程序运行后屏幕显示如图 1-1, 焦点在“请输入自变量的值: x =”右侧的文本框(txtZbl), 此时输入自变量的数值(如: 10), 然后点击“运算”按钮, 屏幕就显示“运算结果: R(10) = 1999”, 重新输入自变量的其它数值, 再点击“运算”按钮, 屏幕又显示新的运算结果……。

程序 1.2 n 次多项式

一 功能

求 n 次多项式

$$P(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n$$

的值。

二 算法说明

1. 本程序中, 多项式的次数和各项的系数都可以自由选定。

2. 为了提高运算速度, 先将多项式表示成嵌套形式:

$$P(x) = (\cdots((a_0x + a_1)x + a_2)x + \cdots + a_{n-1})x + a_n$$

然后从里往外一层一层地计算。递推公式如下:

$$u_0 = a_0$$

$$u_k = u_{k-1}x + a_k \quad k = 1, 2, \cdots n$$

最后得到的 u_n 便是多项式的值 $P(x)$ 。

三 对象及属性表

对象名称	属性	值
窗体 Form1	Caption	n 次多项式求值
标签 Label1	Caption	函数表达式:
Label2	Caption	请输入多项式的次数: n =
Label3	Caption	请输入系数:
Label4	Caption	请输入自变量的值: x =
lblXs(0)	Caption	a0 =
lblXs(1)	Caption	a1 =
lblXs(2)	Caption	a2 =
lblJg	Caption	(清空)
文本框 txtCs	Text	2
txtZbl	Text	(清空)
txtXs(0)	Text	(清空)
txtXs(1)	Text	(清空)

对象名称	属性	值
txtXs(2)	Text	(清空)
按钮 cmdRun	Caption	运算
cmdExit	Caption	退出
OLE 控件 OLE1	Class	Equation 3(已知多项式的表达式)

四 窗体界面

图 1-2

五 程序说明

1. x 表示自变量, n 存放多项式的次数, 数组 a 存放多项式的各个系数, u 寄存函数值, i 为循环变量。

2. 本程序借助标签控件数组 $lblXs$ 和文本框控件数组 $txtXs$ 输入多项式系数。

3. 窗体设计时将多项式的次数预置为 2, 但在运行时可以自由改变, 数组 a 及控件数组 $lblXs$ 、 $txtXs$ 的长度没有预先确定。在程序中将它们分别设为动态数组和动态控件数组。

4. 函数值计算采用如下递推公式:

$$u_0 = a_0$$

$$u_k = u_{k-1}x + a_k \quad k = 1, 2, \dots, n$$

最后得到的 u_n 便是所求的函数值。

5. 多项式各系数赋值及函数值计算的递推过程用 For - Next 循环语句实现。

6. 本程序需要输入的参数较多, 为了便于运行, 编写了焦点转移过程。

六 程序代码

```
Option Explicit
Dim a() As Single
Dim x As Single
Dim u As Single
Dim n As Integer
Dim i As Integer

Private Sub cmdRun_Click()
' 运算
    For i = 0 To n
        a(i) = Val(txtXs(i).Text)
    Next i
    x = Val(txtZbl.Text)
    u = a(0)
    For i = 1 To n
        u = u * x + a(i)
    Next i
    lblJg.Caption = "运算结果:P(" + Str(x) + ")=" + Str(u)
    cmdExit.SetFocus
End Sub

Private Sub cmdExit_Click()
' 退出
    End
End Sub

Private Sub txtXs_KeyPress(Index As Integer, KeyAscii As Integer)
' 按回车转移焦点
    If KeyAscii = 13 Then
        If Index < n Then
            txtXs(Index + 1).SetFocus
        Else
            txtZbl.SetFocus
        End If
    End If
End Sub
```

```

        End If
        KeyAscii = 0
    End If
End Sub

Private Sub TxtCs_KeyPress(KeyAscii As Integer)
' 按回车转移焦点
    If KeyAscii = 13 Then
        txtXs(0).SetFocus
        KeyAscii = 0
    End If
End Sub

Private Sub TxtCs_LostFocus()
' 定义多项式的次数、系数数组大小以及输入系数的有关控件(标签、文本框)
' 采用动态数组、动态控件数组等技术
    n = Val(txtCs.Text)
    ReDim a(n)
    If n > 2 Then
        For i = 3 To n
            Load lblXs(i)
            Load txtXs(i)
            If i / 15 = Int(i / 15) Then
                lblXs(i).Left = lblXs(i - 15).Left + 1850
                txtXs(i).Left = txtXs(i - 15).Left + 1850
                lblXs(i).Top = lblXs(i - 15).Top
                txtXs(i).Top = txtXs(i - 15).Top
            Else
                lblXs(i).Left = lblXs(i - 1).Left
                txtXs(i).Left = txtXs(i - 1).Left
                lblXs(i).Top = lblXs(i - 1).Top + 480
                txtXs(i).Top = txtXs(i - 1).Top + 480
            End If
            lblXs(i).Visible = True
            txtXs(i).Visible = True
            lblXs(i).Caption = ("a" & i) + "="
        Next i
    End If

```

```
txtXs(0).SetFocus  
End Sub  
  
Private Sub txtZbl_KeyPress(KeyAscii As Integer)  
    ' 按回车转移焦点  
    If KeyAscii = 13 Then  
        cmdRun.SetFocus  
        KeyAscii = 0  
    End If  
End Sub
```

七 运行效果

本程序运行后屏幕显示如图 1-2, 焦点在“请输入多项式的次数: n=”右侧的文本框 (txtCs), 将预置的数值(2)改为需要的数值(例如求三次成本函数的函数值时, 可以输入 3), 然后回车(或用其它方法转移焦点), 便生成动态控件数组, 如图 1-3, 再按照提示操作即可。

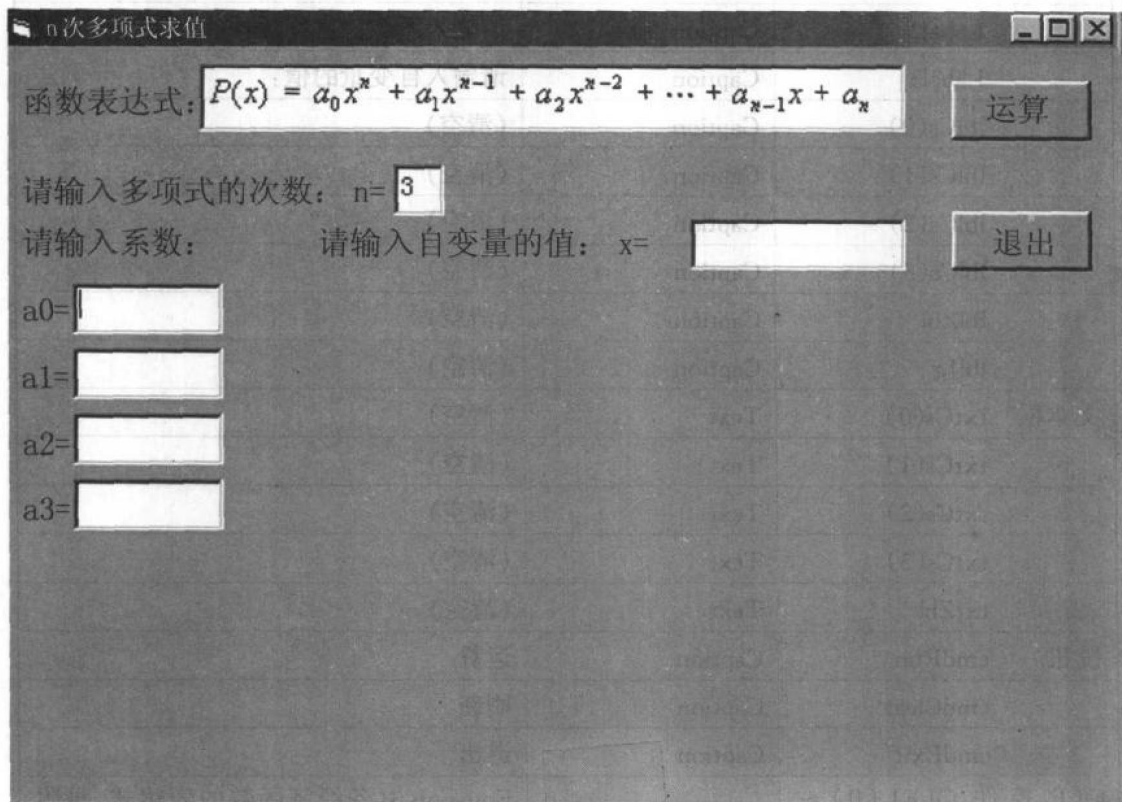


图 1-3

程序 1.3 常用经济函数

一 功能

计算需求函数、Engle 函数、供给函数、成本函数、平均成本函数、复利函数和增长函数等 7 类 20 种经济函数(参见附表 1)的函数值。

二 算法说明

1. 本程序中的各类经济函数都是初等函数,可用赋值语句直接调用内部函数求出函数值。
2. 为了提高运算速度,应尽量少做乘方和乘、除法。

三 对象及属性表

对象名称	属性	值
窗体 Form1	Caption	常用经济函数求值
标签 Label1	Caption	函数表达式:
Label2	Caption	请输入系数:
Label3	Caption	请输入自变量的值:
lblCs(0)	Caption	(清空)
lblCs(1)	Caption	(清空)
lblCs(2)	Caption	(清空)
lblCs(3)	Caption	(清空)
lblZbl	Caption	(清空)
lblJg	Caption	(清空)
文本框 txtCs(0)	Text	(清空)
txtCs(1)	Text	(清空)
txtCs(2)	Text	(清空)
txtCs(3)	Text	(清空)
txtZbl	Text	(清空)
按钮 cmdRun	Caption	运算
cmdClear	Caption	刷新
cmdExit	Caption	退出
OLE 控件 OLE1(0) ~ OLE1(19)	Class	Equation 3(各经济函数的表达式,见附表 1)
菜单		(见附表 2)

附表 1

控件名称	函数表达式
OLE1(0)	$Q = a - bp \quad (a > 0, b > 0)$
OLE1(1)	$Q = \frac{a}{p+c} - b \quad (a > 0)$
OLE1(2)	$Q = a - bp - cp^2 \quad (a > 0, b \geq 0, c > 0)$
OLE1(3)	$Q = Ae^{-bp} \quad (A > 0, b > 0)$
OLE1(4)	$Q = Qp^{-r} \quad (A > 0, r > 0)$
OLE1(5)	$y = ax^b \quad (a > 0)$
OLE1(6)	$y = \frac{ax}{x+b} \quad (a > 0, b > 0)$
OLE1(7)	$y = \frac{a(x-c)}{x+b} \quad (a > 0, b \geq 0, c \geq 0)$
OLE1(8)	$y = \frac{ax(x-c)}{x+b} \quad (a > 0, b > 0, c > 0)$
OLE1(9)	$Q = ap - b \quad (a > 0, b > 0)$
OLE1(10)	$Q = ap^r - b \quad (a > 0, b > 0, r > 0)$
OLE1(11)	$C = aQ + b \quad (a > 0, b \geq 0)$
OLE1(12)	$C = aQ^2 + bQ + c \quad b \geq 0, c \geq 0)$
OLE1(13)	$C = aQ^3 + bQ^2 + cQ + d \quad (c > 0, d > 0)$
OLE1(14)	$C_A = \frac{aQ+b}{Q} \quad (a > 0, b \geq 0)$
OLE1(15)	$C_A = \frac{aQ^2+bQ+C}{Q} \quad (b \geq 0, c \geq 0)$
OLE1(16)	$C_A = \frac{aQ^3+bQ^2+cQ+d}{Q} \quad (c > 0, d > 0)$

OLE1(17)	$A_t = A_0(1+r)^t$
OLE1(18)	$A_t = A_t(1+r)^{-t}$
OLE1(19)	$A_t = A_0 e^{rt}$

附表 2

菜单级别	菜单标题	名称
1	需求函数	xqhs
2	线性需求函数	xxxqhs
2	双曲线需求函数	sqxxqhs
2	抛物线需求函数	pwxxqhs
2	指数式需求函数	zssxqhs
2	幂函数式需求函数	mhssxqhs
1	Engle 函数	Engle
2	I 型 $y = a * x^b$	Engle _ 1
2	II 型 $y = a * x / (x + b)$	Engle _ 2
2	III 型 $y = a * (x - c) / (x + b)$	Engle _ 3
2	IV 型 $y = a * x * (x - c) / (x + b)$	Engle _ 4
1	供给函数	gjhs
2	I 型 $Q = a * p - b$	gjhs _ 1
2	II 型 $Q = a * p^r - b$	gjhs _ 2
1	成本函数	cbhs
2	线性成本函数	xxcbhs
2	二次成本函数	eccbhs
2	三次成本函数	sccbhs
1	平均成本函数	pjcbhs
2	成本为线性函数	xxcbhs
2	成本为二次函数	eccbhs
2	成本为三次函数	sccbhs
1	复利函数	flhs
2	复利问题	flwt
2	贴现问题	txwt
1	增长函数	zzhs

四 窗体界面

在窗体 Form1 中,对象 OLE1(0)~OLE1(19)的位置是重迭在一起的,它们初始时刻的 Visible 属性均为 false。

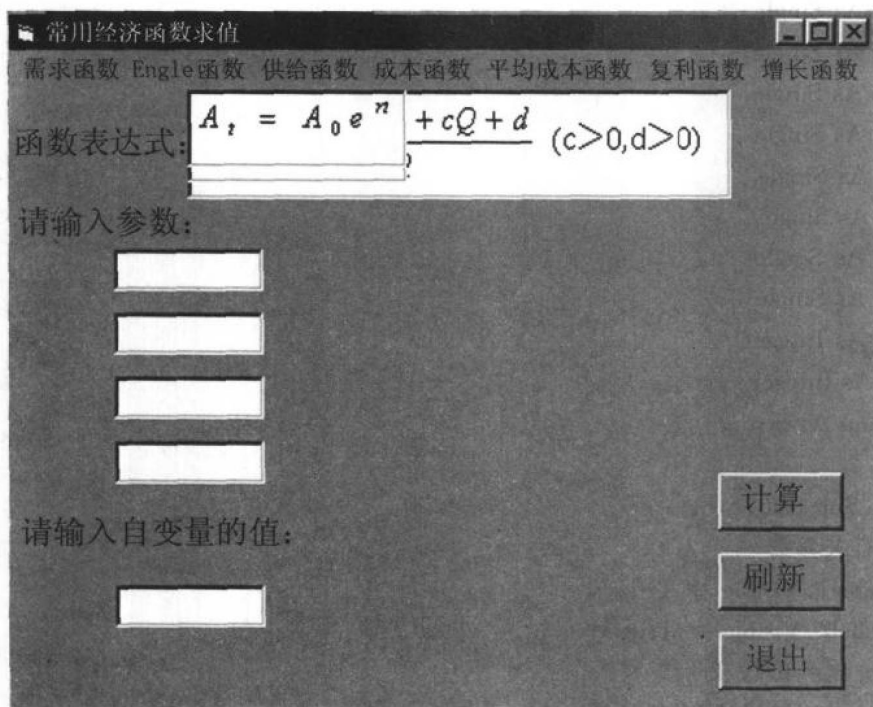


图 1-4

五 程序说明

1. 常用经济函数的自变量按照习惯分别用 p 、 x 、 q 、 t 等表示,函数值则用 q 、 y 等表示, a 、 b 、 c 、 d 、 r 存放函数的各个参数, k 表示参数的个数(比实际参数个数少 1), i 为循环变量, mnu 寄存菜单代号。

2. 本程序借助标签控件数组 $lblXs$ 和文本框控件数组 $txtXs$ 输入函数的各个参数。控件数组 $lblXs$ 、 $txtXs$ 的长度按各函数中参数个数最多者定义,对于参数较少的函数,运行时,使多余的控件不可见。

3. 各菜单项对应的 Click 事件过程主要完成的任务是:设置菜单代号、显示相应的函数表达式及输入参数的文本框,准备输入各参数。其共同处理的任务专门编写一个子过程 res 。

4. 点击“运算”按钮对应的事件过程 $cmdRun_click$ 使用了两级 select case 语句,第一级对应一级菜单,第二级对应二级菜单。

六 程序代码

```
Option Explicit
Dim a As Single
Dim b As Single
Dim c As Single
Dim d As Single
Dim p As Single
Dim q As Single
Dim r As Single
Dim t As Single
Dim x As Single
Dim y As Single
Dim k As Integer
Dim i As Integer
Dim mnu As String
```

```
Private Sub res()
```

```
    ' 显示有关参数
```

```
        Label1.Visible = True
```

```
        Label2.Visible = True
```

```
        For i = 0 To k
```

```
            txtCs(i).Visible = True
```

```
        Next i
```

```
        txtCs(0).SetFocus
```

```
End Sub
```

```
Private Sub cmdRun _ Click()
```

```
    ' 运算
```

```
        Select Case Left(mnu, 1)
```

```
            Case "1"
```

```
                a = Val(txtCs(0))
```

```
                b = Val(txtCs(1))
```

```
                p = Val(txtZbl)
```

```
                Select Case mnu
```

```
                    Case "11"
```

```
                        q = a - b * p
```

```
                    Case "12"
```

```
c = Val(txtCs(2))
q = a / (p + c) - b
Case "13"
c = Val(txtCs(2))
q = a - (b + c * p) * p
Case "14"
q = a * Exp(-b * p)
Case "15"
q = a * p ^ (-b)
End Select
lblJg.Caption = "运算结果: Q=" + Str(q)
Case "2"
a = Val(txtCs(0))
b = Val(txtCs(1))
x = Val(txtZbl)
Select Case mnu
Case "21"
y = a * x ^ b
Case "22"
y = a * x / (x + b)
Case "23"
c = Val(txtCs(2))
y = a * (x - c) / (x + b)
Case "24"
c = Val(txtCs(2))
y = a * x * (x - c) / (x + b)
End Select
lblJg.Caption = "运算结果: Y=" + Str(y)
Case "3"
a = Val(txtCs(0))
b = Val(txtCs(1))
p = Val(txtZbl)
Select Case mnu
Case "31"
q = a * p - b
Case "32"
r = Val(txtCs(2))
q = a * p ^ r - b
```