

TALLINNA TEHNIKAÜLIKOOL

TARKVARATEHNIKA PROJEKT

**Komponenttehnoloogia kasutamine
tarkvaraprojektides.**

Üliõpilane: Ivari Horm
990673 LAS

Juhendaja: K. Grigorjeva

2003

Sisukord

SISUKORD	2
SISSEJUHATUS	3
PROJEKTI EESMÄRK	3
PROJEKTI ÜLEVAADE	4
KALKULAATOR	5
KARTOTEEK	6
KOKKUVÕTE	7
LISA1: KALKULAATORIKOMPONENDI PROGRAMMIKOOD	8
LISA2: KARTOTEEGIKOMPONENDI PROGRAMMIKOOD	14

Sissejuhatus

Viimasel ajal on tarkvara loomise protsessis järjest suuremat populaarsust võitmas komponenttehnoloogia. Viimane kujutab endast varem valmisprogrammeeritud taaskasutatavate komponentide rakendamist konkreetse tarkvaraprojekti realiseerimisel. Komponent ise kujutab endast mingit kindlat ülesannet täitvat programmi, mis omab kindlat arvu sisendeid ja väljundeid. Microsofti vastav tehnoloogia kannab nimetust ActiveX (varasema nimetusega OLE), Sun'i poolt arendatavas Javas on selleks kasutusel nn. Java Bean'id.

Projekti eesmärk

Antud projekti eesmärk on tutvustada komponenttehnoloogia võimalusi, realiseerides näiteülesandena tarkvaraprojekti, mis kasutab endas kahte ActiveX komponenti.

Projekti ülevaade

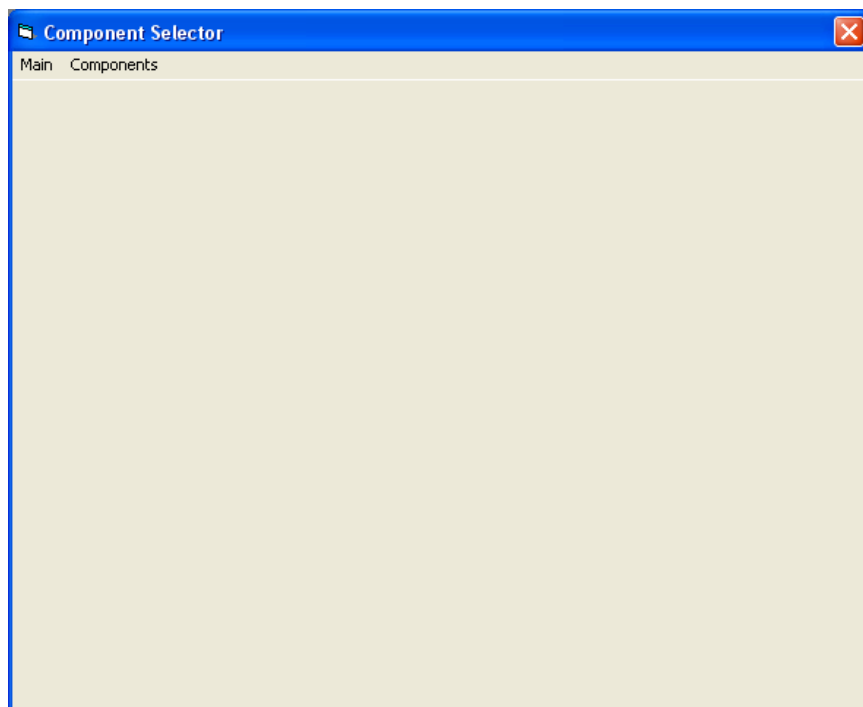
Loodav tarkvara koosneb kahest ActiveX komponendist:

- Arvutusi hõlbustav kalkulaator.
- Andmete salvestamiseks mõeldud kartoteek.

Mõlemad komponendid on seotud kokku üheks käivitatavaks programmiks. Samuti on kalkulaatori komponent paigutatud veebi, mis tagab selle kättesaamise suvalisest Interentti ühendatud arvutist.

Realiseerimisel on kasutatud Microsofti programmeerimiskeskonda Visual Basic 6, seetõttu on antud tarkvara kasutatav ilma abiprogrammideta vaid Windowsi platvormil.

Programmi käivitamisel ilmub dialoogikast, kust on võimalik valida, millise komponendiga tööd jätkata:

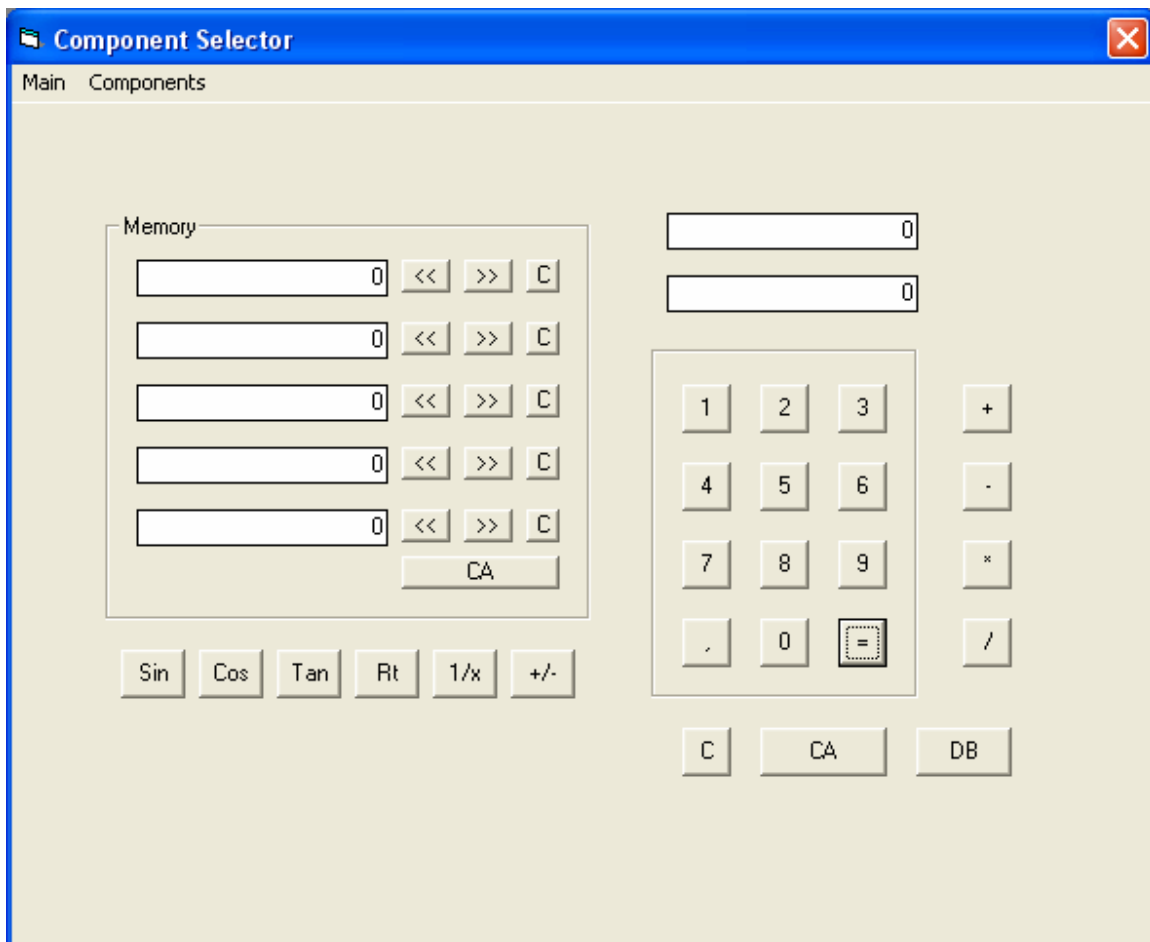


Kalkulaator

Antud komponent võimaldab teha elementaarseid arvutustehteid. Lisaks on võimalik arvutada ka peamiste trigonomeetriliste funktsioonide (siinus, koosinus, tangens) väärtusi.

Kalkulaatori displei koosneb kahest väljast (nn. X ja Y registrist), mis binaarsete operatsioonide puhul tagab mõlema operandi nägemise.

Võimalik on kasutada ka viite mäluelementi. Mäluelementides olevate arvudega saab teha samamoodi tehteid nagu registrites olevate väärtustegagi, s.t. mälus olevale arvule ja on võimalik Y-registris olev arv juurde liita, samuti on võimalik teostada ka vastupidine operatsioon. Võimalikke operatsioone endid on kokku neli: liitmine, lahutamine, korrutamine ja jagamine.



Lisaks on võimalik nupuga DB salvestada mõlema registri sisu andmebaasi, mida saab hiljem kuvada kartoteegimooduliga.

Kalkulaatori komponenti on võimalik kasutada ka Interneti vahendusel aadressil <http://study.deepdust.com/soft/kalkulaator.php>. Kuna hetkel on komponent veel signeerimata, tuleb selle kasutamiseks muuta arvuti Internet Exploreri seadeid:

- Valida menüüst „Tools“ käsk „Internet Options“
- Ilmunud aknast tuleb kuvada lipik (Tab) „Security“
- Klõpsata nupul „Custom Level...“
- Valik „Download unsigned ActiveX components“ valida „Disable“ asemel „Prompt“.

Hiljem on mõistlik antud muudatused tühistada ja varem kehtinud seaded taastada.

Kalkulaatorikomponendi programmikood on toodud lisas nr. 1

Kartoteek

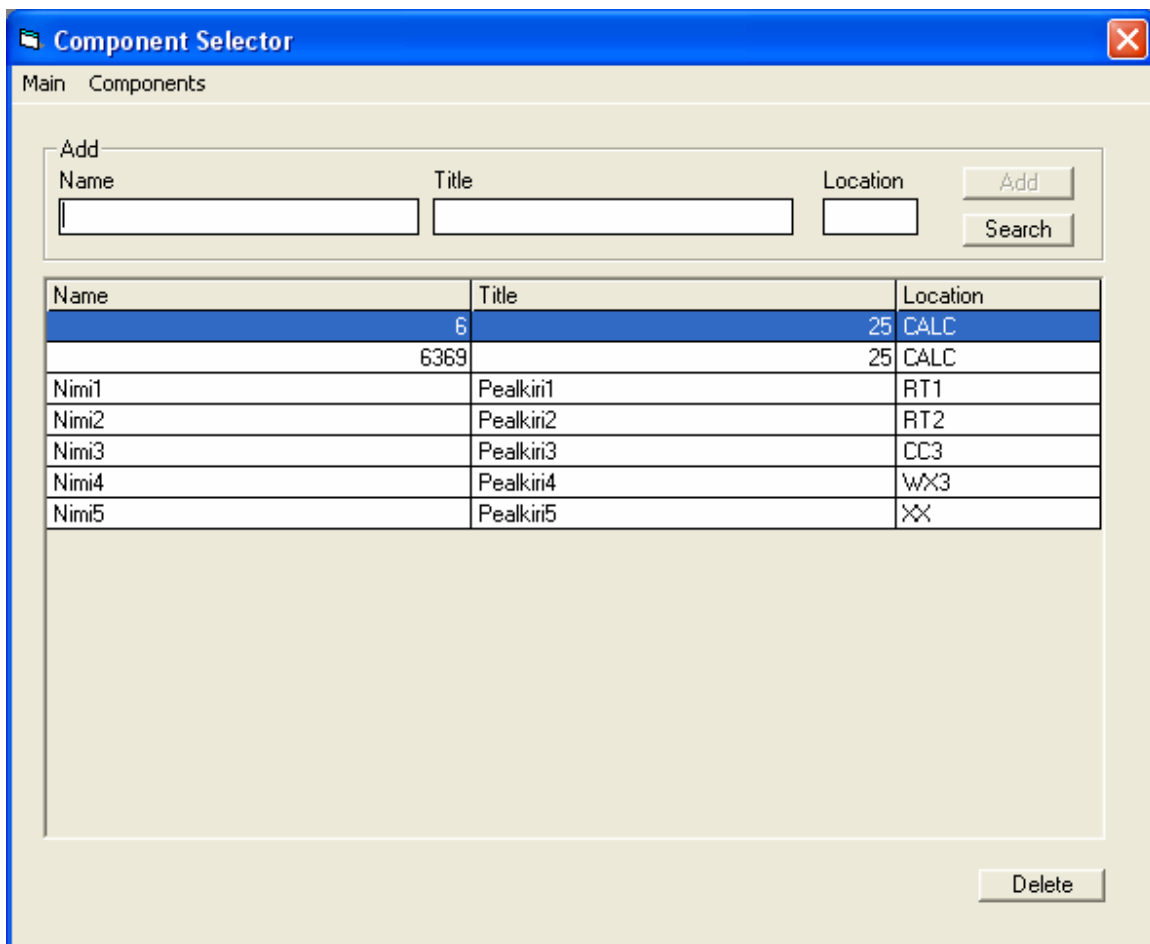
Komponent kujutab endast kolme andmeväljaga andmebaasi:

- Nimi
- Pealkiri
- Asukoht

Kasutusel on MS Accessi failiformaat, mis tagab ühilduvuse teiste sama firma toodetega.

Komponent võimaldab:

- Andmebaasi uute kirjade lisamist
Andmete salvestamisel kalkulaatori komponendist paigutatakse andmeväljale „asukoht“ kirje „CALC“.
- Olemasolevate kirjade seast otsimist mistahes andmevälja või nende kombinatsioonide alusel
- Andmebaasis olevate kirjade kustutamist



Kartoteegikomponendi programmikood on toodud lisis nr. 2.

Kokkuvõte

Komponenttehnoloogia on suurepärase võimaluse tarkvara loomisprotsessi lihtsustada, vähendades seeläbi tarkvaraprojektidele tehtavaid ajalisi ja rahalisi kulusi. Samas on valmiskomponentide laialdasem kasutamine siiski ebatõenäoline, kuna enamik loodavast tarkvarast on ülesandespetsiifiline, mistõttu ei pruugi leiduda valmiskomponente, mida vahetult kasutada saab. Sellegipoolest võib sõltuvalt ülesandest väikeste kompaktsete moodulite ringiprogrammeerimine osutuda otstarbekamaks kui suure terviklahenduse algusest lõpuni realiseerimine.

Niisiis kujutab komponenttehnoloogia endast suurepärasest võimalusest tüütute korduvate protseduuride igakordsest uuestiprogrammeerimisest loobumiseks.

LISA1: Kalkulaatorikomponendi programmikood

```
Private Sub db_Click()  
    With Adodc1  
        .ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=C:\Andmed.mdb;Persist Security Info=False"  
        .RecordSource = "select * from datatable order by Name, Title"  
    End With  
    Adodc1.Refresh  
    With Adodc1.Recordset  
        .AddNew  
        !Name = operx  
        !Title = opery  
        !location = "CALC"  
        .Update  
        .Requery  
    End With  
    Adodc1.Refresh  
End Sub
```

```
Private Sub UserControl_KeyPress(KeyCode As Integer)  
Select Case KeyCode  
    Case 49:    addnrx (1)  
    Case 50:    addnrx (2)  
    Case 51:    addnrx (3)  
    Case 52:    addnrx (4)  
    Case 53:    addnrx (5)  
    Case 54:    addnrx (6)  
    Case 55:    addnrx (7)  
    Case 56:    addnrx (8)  
    Case 57:    addnrx (9)  
    Case 48:    addnrx (0)  
    Case 44:    Call comma_Click  
    Case 42:    Call mul_Click  
    Case 47:    Call equ_Click  
    Case 45:    Call sub_Click  
    Case 43:    Call add_Click  
    Case 27:    Call clearall_Click  
    Case 8:     Call clear_Click  
    Case 13:    Call result_Click  
End Select
```

```
KeyCode = 0  
End Sub
```

```
Private Sub add_Click()  
addnry ("+")  
End Sub
```

```
Private Sub am1_Click()  
If (opery.Text <> 0) Then  
    If (m1 <> 0 And sign.Caption <> "" And wasunar = 0) Then  
        m1 = math.Eval(Replace(m1.Text, ",", ".")) + sign.Caption +  
Replace(opery.Text, ",", ".")  
    Else  
        m1 = opery.Text  
    End If  
Else  
    m1 = operx.Text  
End If  
End Sub
```

```
Private Sub am2_Click()  
If (opery.Text <> 0) Then  
    If (m2 <> 0 And sign.Caption <> "" And wasunar = 0) Then
```



```

        m2 = math.Eval(Replace(m2.Text, ",", ".") + sign.Caption +
Replace(opery.Text, ",", "."))
    Else
        m2 = opery.Text
    End If
Else
    m2 = operx.Text
End If
End Sub

Private Sub am3_Click()
If (opery.Text <> 0) Then
    If (m3 <> 0 And sign.Caption <> "" And wasunar = 0) Then
        m3 = math.Eval(Replace(m3.Text, ",", ".") + sign.Caption +
Replace(opery.Text, ",", "."))
    Else
        m3 = opery.Text
    End If
Else
    m3 = operx.Text
End If
End Sub

Private Sub am4_Click()
If (opery.Text <> 0) Then
    If (m4 <> 0 And sign.Caption <> "" And wasunar = 0) Then
        m4 = math.Eval(Replace(m4.Text, ",", ".") + sign.Caption +
Replace(opery.Text, ",", "."))
    Else
        m4 = opery.Text
    End If
Else
    m4 = operx.Text
End If
End Sub

Private Sub am5_Click()
If (opery.Text <> 0) Then
    If (m5 <> 0 And sign.Caption <> "" And wasunar = 0) Then
        m5 = math.Eval(Replace(m5.Text, ",", ".") + sign.Caption +
Replace(opery.Text, ",", "."))
    Else
        m5 = opery.Text
    End If
Else
    m5 = operx.Text
End If
End Sub

Private Sub chansign_Click()
sign = "+/-"
Call unarfun
End Sub

Private Sub clear_Click()
If (sign.Caption = "") Then
    opery = 0
Else
    If (operx.Text = 0) Then
        sign = ""
        wasunar = 0
    Else
        operx = 0
    End If
End If
End Sub

```

```

    End If
End If
End Sub

Private Sub clearall_Click()
operx = 0
sign = ""
opery = 0
wasunar = 0
End Sub

Private Sub cm1_Click()
m1 = 0
End Sub

Private Sub cm2_Click()
m2 = 0
End Sub

Private Sub cm3_Click()
m3 = 0
End Sub

Private Sub cm4_Click()
m4 = 0
End Sub

Private Sub cm5_Click()
m5 = 0
End Sub

Private Sub cma_Click()
m1 = 0
m2 = 0
m3 = 0
m4 = 0
m5 = 0
End Sub

Private Sub comma_Click()
If (Fix(operx.Text) = operx.Text) Then
iscomma = 1
End If
End Sub

Private Sub cosinus_Click()
sign = "Cos"
Call unarfun
End Sub

Private Sub equ_Click()
addnry ("/")
End Sub

Private Sub mul_Click()
addnry ("*")
End Sub

Private Sub nr1_Click()
addnrx (1)
End Sub

Private Sub addnrx(nr)

```

```

If (wasunar = 1) Then
Call clearall_Click
End If
If (operx.Text = 0) Then
operx = nr
Else
operx = operx.Text & nr
End If
If (iscomma = 1) Then
operx = operx.Text / 10
iscomma = 0
End If
End Sub
Private Sub addnry(sig)
If (wasunar = 1) Then
sign = sig
wasunar = 0
End If
If (operx.Text <> 0) Then
If (opery.Text <> 0) Then
Call result_Click
Else
opery = operx.Text
operx = 0
End If
End If
sign = sig
End Sub

Private Sub nr2_Click()
addnrx (2)
End Sub

Private Sub nr3_Click()
addnrx (3)
End Sub
Private Sub nr4_Click()
addnrx (4)
End Sub
Private Sub nr5_Click()
addnrx (5)
End Sub
Private Sub nr6_Click()
addnrx (6)
End Sub
Private Sub nr7_Click()
addnrx (7)
End Sub
Private Sub nr8_Click()
addnrx (8)
End Sub
Private Sub nr9_Click()
addnrx (9)
End Sub
Private Sub nr0_Click()
addnrx (0)
End Sub

Private Sub result_Click()
On Error GoTo error
If (sign.Caption <> "" And wasunar <> 1) Then
opery = math.Eval(Replace(opery.Text, ",", ".") + sign.Caption +
Replace(operx.Text, ",", "."))

```

```

Else
    opery = operx
End If
operx = 0
Exit Sub
error: Call errorhandler
End Sub

Private Sub revers_Click()
sign = "1/x"
Call unarfun
End Sub

Private Sub rm1_Click()
If (m1 <> 0 And sign.Caption <> "" And wasunar = 0) Then
    opery = math.Eval(Replace(opery.Text, ",", ".") + sign.Caption +
Replace(m1.Text, ",", "."))
Else
    opery = m1.Text
End If
End Sub

Private Sub rm2_Click()
If (m2 <> 0 And sign.Caption <> "" And wasunar = 0) Then
    opery = math.Eval(Replace(opery.Text, ",", ".") + sign.Caption +
Replace(m2.Text, ",", "."))
Else
    opery = m2.Text
End If
End Sub

Private Sub rm3_Click()
If (m3 <> 0 And sign.Caption <> "" And wasunar = 0) Then
    opery = math.Eval(Replace(opery.Text, ",", ".") + sign.Caption +
Replace(m3.Text, ",", "."))
Else
    opery = m3.Text
End If
End Sub

Private Sub rm4_Click()
If (m4 <> 0 And sign.Caption <> "" And wasunar = 0) Then
    opery = math.Eval(Replace(opery.Text, ",", ".") + sign.Caption +
Replace(m4.Text, ",", "."))
Else
    opery = m4.Text
End If
End Sub

Private Sub rm5_Click()
If (m5 <> 0 And sign.Caption <> "" And wasunar = 0) Then
    opery = math.Eval(Replace(opery.Text, ",", ".") + sign.Caption +
Replace(m5.Text, ",", "."))
Else
    opery = m5.Text
End If
End Sub

Private Sub sinus_Click()
sign = "Sin"
Call unarfun
End Sub

```

```

Private Sub sqrt_Click()
sign = "Rt"
Call unarfun
End Sub

Private Sub sub_Click()
addnry ("-")
End Sub

Private Sub unarfun()
On Error GoTo error
wasunar = 1
If (operx <> 0) Then
    If (sign.Caption = "Rt") Then
        operx = Sqr(operx.Text)
    ElseIf (sign.Caption = "1/x") Then
        operx = 1 / operx.Text
    ElseIf (sign.Caption = "+/-") Then
        operx = -operx.Text
    Else
        operx = math.Eval(sign.Caption + "(" + Replace(operx.Text, ",", ".") +
")")
    End If
Else
    If (sign.Caption = "Rt") Then
        opery = Sqr(operx.Text)
    ElseIf (sign.Caption = "1/x") Then
        opery = 1 / operx.Text
    ElseIf (sign.Caption = "+/-") Then
        opery = -operx.Text
    Else
        opery = math.Eval(sign.Caption + "(" + Replace(operx.Text, ",", ".") +
")")
    End If
End If
Exit Sub
error: Call errorhandler
End Sub

Private Sub tangens_Click()
sign = "Tan"
Call unarfun
End Sub

Private Sub errorhandler()
Select Case Err.Number
    Case 11: MsgBox ("You cannot divide by zero!")
    Case 5: MsgBox ("Cannot perform that operation!")
End Select
End Sub

```

LISA2: Kartoteegikomponendi programmikood

```
Option Explicit
Private Sub add_Click()
    With Adodcl.Recordset
        .AddNew
        !Name = nname
        !title = title
        !location = location
        .Update
        .Requery
    End With
    nname = ""
    title = ""
    location = ""
    Call Resize_Grid
End Sub

Private Sub delete_Click()
    With Adodcl.Recordset
        .Move (MSHFlexGrid1.Row - 1)
        .delete
        .Requery
    End With
    Call Resize_Grid
End Sub

Public Sub Loadit()
    With Adodcl
        .ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Andmed.mdb;Persist Security Info=False"
        .RecordSource = "select * from datatable order by Name, Title"
    End With
    Call Resize_Grid
End Sub

Private Sub Resize_Grid()
    Adodcl.Refresh
    Set MSHFlexGrid1.DataSource = Adodcl
    MSHFlexGrid1.FormatString = "Name | Title | Location"
    MSHFlexGrid1.ColWidth(0) = 0.4 * MSHFlexGrid1.Width
    MSHFlexGrid1.ColWidth(1) = MSHFlexGrid1.ColWidth(0)
    MSHFlexGrid1.ColWidth(2) = MSHFlexGrid1.Width - (MSHFlexGrid1.ColWidth(0) *
2) - 60
    nname.SetFocus
End Sub

Private Sub search_Click()
    If (search.Caption = "Search") Then
        'Adodcl.RecordSource = "select * from datatable where Name='" + nname.Text +
"' and Title='" + title.Text + "' and location='" + location.Text + "' order by
Name, Title"
        Adodcl.RecordSource = "select * from datatable where Name like '" +
nname.Text + "%' and Title like '" + title.Text + "%' and location like '" +
location.Text + "%' order by Name, Title"
        search.Caption = "View All"
    Else
        Adodcl.RecordSource = "select * from datatable order by Name, Title"
        search.Caption = "Search"
    End If
    Call Resize_Grid
End Sub
```

```
Private Sub nname_Change()  
    If nname.Text <> "" And title.Text <> "" And location.Text <> "" Then  
        add.Enabled = True  
    Else  
        add.Enabled = False  
    End If  
  
End Sub  
  
Private Sub title_Change()  
    Call nname_Change  
End Sub  
  
Private Sub location_Change()  
    Call nname_Change  
End Sub
```