

## Code des Objektes „frmDatenVisualisierung“

Option Explicit	'Alle Variablen müssen deklariert sein
Dim Zeit As String	'Für die Umwandlung der Datei Erstellungszeit
Dim Dateiname As String	'Name der zu speichernden Datei
Dim Vorschlag As String	'Übernimmt das Dateidatum und den File Name Vorschlag

---

'Private :Auf die Sub-Prozedur kann nur durch andere Prozeduren aus dem Modul zugegriffen werden,  
'in dem sie deklariert wurde.

Private Sub cboDiagramme\_Click()

AktiviereDiagramm 'Die Prozedur AktiviereDiagramm wird aufgerufen

Select Case cboDiagramme.Value 'Wenn der folgende Punkt im Kombinationsfeld gewählt wird

Case 0 'XY Linie mit Datenpunkten

imgDiagramme.Picture = LoadPicture("c:\messdaten\pix\xypulin.bmp") 'Laden des entspr. Bildes

ActiveChart.ChartArea.Select 'Diagrammfläche wird gewählt

ActiveChart.ChartType = xlXYScatterLines 'Diagrammtyp wird der aktiven Fläche zugewiesen

If chkDatenanzeigen.Enabled = False Then 'Einschalten der Datenbeschriftungsoption

chkDatenanzeigen.Enabled = True

End If

Case 1 'XY interpolierte Linie ohne Datenpunkte

imgDiagramme.Picture = LoadPicture("c:\messdaten\pix\xyiplin.bmp")

ActiveChart.ChartArea.Select

ActiveChart.ChartType = xlXYScatterSmoothNoMarkers

If chkDatenanzeigen.Enabled = False Then

chkDatenanzeigen.Enabled = True

End If

Case 2 'XY nur Datenpunkte

imgDiagramme.Picture = LoadPicture("c:\messdaten\pix\xynurpu.bmp")

ActiveChart.ChartArea.Select

ActiveChart.ChartType = xlXYScatter

If chkDatenanzeigen.Enabled = False Then

chkDatenanzeigen.Enabled = True

End If

Case 3 'Säulen gruppiert

datenaus 'Prozedur datenaus löscht die Datenbeschriftung

imgDiagramme.Picture = LoadPicture("c:\messdaten\pix\saeulgr.bmp")

ActiveChart.ChartArea.Select

ActiveChart.ChartType = xlColumnClustered

chkDatenanzeigen.Enabled = False 'ausschalten, da Beschriftung unübersichtlich

chkDatenanzeigen.Value = False 'abwählen der Datenbeschriftungsoption

Case 4 'Kegelsäulen gruppiert

datenaus

imgDiagramme.Picture = LoadPicture("c:\messdaten\pix\kegsgr.bmp")

ActiveChart.ChartArea.Select

ActiveChart.ChartType = xlConeColClustered

chkDatenanzeigen.Enabled = False

chkDatenanzeigen.Value = False

End Select

End Sub

```

Private Sub chkDatenanzeigen_Change()
If chkDatenanzeigen.Value = True Then
    ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowValue,
    LegendKey:=False
    ActiveChart.SeriesCollection(3).DataLabels.Select
    Selection.AutoScaleFont = True
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Standard"
        .Size = 8
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
        .Background = xlAutomatic
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .Position = xlLabelPositionBelow
        .Orientation = xlHorizontal
    End With
    ActiveChart.SeriesCollection(1).DataLabels.Select
    Selection.AutoScaleFont = True
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Standard"
        .Size = 8
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
        .Background = xlAutomatic
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .Position = xlLabelPositionAbove
        .Orientation = xlHorizontal
    End With
    ActiveChart.SeriesCollection(2).DataLabels.Select
    Selection.AutoScaleFont = True
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Standard"
        .Size = 8
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False

```

'Bei Änderung des vorherigen Zustandes,Prozedur

'Wenn chkBox angewählt, dann

'Datenbeschriftung werden an den Datenreihen angezeigt

'Datenreihe 3 wird selektiert

'Formatierungen der Datenbeschriftungseigenschaften

'der aktiven Datenreihe

'Einstellung der Schrift, ihrer Größe, etc.

'Position und Ausrichtung der Datenbeschriftung ,

' bezogen auf die entsprechende Datenreihe

'Datenreihe 1 wird selektiert

'Datenreihe 2 wird selektiert

```

.Shadow = False
.Underline = xlUnderlineStyleNone
.ColorIndex = xlAutomatic
.Background = xlAutomatic
End With
With Selection
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
.Position = xlLabelPositionAbove
.Orientation = xlHorizontal
End With
With Selection
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
.Position = xlLabelPositionBelow
.Orientation = xlHorizontal
End With

Else                                     'Wenn die chkBox abgewählt wird
    datenaus                             'Prozedur zum deaktivieren der Datenbeschriftung wird aufgerufen
End If

End Sub

```

---

```

Private Sub chkDiagrammBack_Click()      'Ein- und Ausschalten des Diagramm-Hintergrundes
    AktiviereDiagramm                   'Diagramm aktivieren
If chkDiagrammback.Value = False Then   'Wenn chkBox nicht gewählt und dann angeklickt wird
    ActiveChart.ChartArea.Select
    ActiveChart.PlotArea.Select
    Selection.Interior.ColorIndex = xlNone 'wird der Hintergrund ausgeschaltet
    chkDiagrammback.ControlTipText = "Der Zeichenflächen Hintergrund ist ausgeschaltet"
    'Der Hilfetext der beim Überqueren mit der Maus angezeigt wird, wird geändert
Else
    ActiveChart.ChartArea.Select
    ActiveChart.PlotArea.Select
    Selection.Interior.ColorIndex = 15    'wird der Hintergrund auf Grau geschaltet
    chkDiagrammback.ControlTipText = "Der Zeichenflächen Hintergrund ist eingeschaltet"
End If
Range("A31").Select 'Eine Zelle ausserhalb des sichtbaren Bereiches wird gewählt (für die Optik)
End Sub

```

---

```

Private Sub cmdClear_Click()             'Löscht das Datenblatt und das Diagramm
Range("B3:D12").Select                   'Alle Tabellendaten ausser der Überschrift werden angewählt
    Selection.Value = 0                   'und mit "nichts" gefüllt
    Range("A1").Select                    'Dann wird das Feld A1 gewählt
cboDiagramme.Enabled = False            'Diagrammtyp Auswahl deaktiviert da kein Diagramm vorhanden
Zeit = ""                                 'Die Variable Zeit wird auf "Nichts" gesetzt
    AktiviereDiagramm                    'Das Diagramm wird aktiviert
Range("A31").Select
datenaus                                  'Daten löschen
chkDatenanzeigen.Enabled = False        'Datenanzeigen deaktivieren da keine Daten vorhanden

End Sub

```

**Private Sub** cmdSpeichern\_Click()

Vorschlag = FileDateTime("c:\messdaten\messdat\messdat.txt") 'Einlesen des Datei-Datums für den Speichern-Dialog

Vorschlag = Format(Vorschlag, "ddmmyy\_hm") 'Formatiert auf kurzes Datum Format

Vorschlag = Vorschlag + ".xls" 'hängt an das Datum die xls Endung an

DateiName = Application.GetSaveAsFilename(Vorschlag) 'und trägt den default Namen ein

ActiveWorkbook.SaveAs FileName:="messdat\DateiName", FileFormat:=xlNormal, Password:="",

WriteResPassword:="", \_

ReadOnlyRecommended:=False, CreateBackup:=False

**End Sub**

---

**Private Sub** cmdFormausblenden\_Click()

frmDatenVisualisierung.Hide 'Blendet das Formular aus, vorher kann nicht im Worksheet gearbeitet werden

**End Sub**

---

**Private Sub** cmdLaden\_Click() 'Prozedur, die die Messdaten aus der ASCII Datei in die Tabelle lädt

Range("A1").Select 'markieren von Feld A1, damit das Einfügen aus messdaten.txt funktioniert

cboDiagramme.Enabled = True 'nun besteht die Möglichkeit den Diagrammtyp zu verändern

Zeit = FileDateTime("c:\messdaten\messdat\messdat.txt") 'der Variablen Zeit wird die Datei-Zeit zugewiesen

Zeit = Format(Zeit, "dddd , d mmmm yyyy hh:mm:ss") 'diese wird entsprechend unseres

Datumsystemes formatiert

lblMessZeit.Caption = Zeit 'Zeigt die Erstellzeit der Datei auf dem Formular an

chkDatenanzeigen.Enabled = True 'checkBox Datenanzeigen aktivieren (nun kann gewählt werden)

'öffnen der Pascal Datei, markieren der Daten , einfügen in Messdaten.xls

Workbooks.OpenText FileName:="c:\messdaten\messdat\messdat.txt", Origin:= \_

xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= \_

xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, \_

Comma:=False, Space:=False, Other:=False, FieldInfo:=Array(Array(1, 1), \_

Array(2, 1), Array(3, 1), Array(4, 1))

Range("A1:D12").Select 'Der relevante Datenteil wird selektiert

Selection.Copy 'und in die Zwischenbalage kopiert

ActiveWindow.WindowState = xlMinimized 'Das Fenster der importierten Daten wird minimiert

ActiveSheet.Paste 'und die Daten in Messdaten.xls eingefügt

Windows("messdat.txt").Activate

ActiveWindow.WindowState = xlNormal

ActiveWindow.Close

'Die messdat.txt Datei wird wieder geschlossen

Range("A1:D1").Select

Selection.Font.Bold = True

'Fett formatieren der Überschriften

Selection.Borders(xlDiagonalDown).LineStyle = xlNone

'Formatiern der Rahmen

Selection.Borders(xlDiagonalUp).LineStyle = xlNone

**With** Selection.Borders(xlEdgeLeft)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

**End With**

**With** Selection.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.Weight = xlThin

.ColorIndex = xlAutomatic

**End With**

```

With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Range("A2:D12").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Range("A1:D12").Select
With Selection
    .HorizontalAlignment = xlCenter           'Mittige Ausrichtung der Messdaten
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .ShrinkToFit = False
    .MergeCells = False
End With

Columns("A:A").EntireColumn.AutoFit       'Automatische Ausrichtung der Spaltenbreite
Columns("B:B").EntireColumn.AutoFit       'auf die längste Zeichenfolge

```

```
Columns("C:C").EntireColumn.AutoFit
Columns("D:D").EntireColumn.AutoFit
Range("A14").Select
AktiviereDiagramm
```

End Sub

---

```
Private Sub cmdVorschau_Click()           'Erzeugt eine Vorschau des Diagramms
frmDatenVisualisierung.Hide             'Das Formular wird ausgeblendet
AktiviereDiagramm                       'Wählen des Diagramms
ActiveWindow.SelectedSheets.PrintPreview 'Die Druckvorschau des Diagrammes (Vergrößert
auf Seitenbreite)
End Sub
```

---

```
Private Sub UserForm_Initialize() 'All diese Befehle werden beim Laden des Formulares ausgeführt
```

```
'Hinzufügen von Listeneinträgen zu einem Kombinationsfeld. Der Wert jedes Eintrages
'entspricht dem jeweiligen Wert für ListIndex in dem Kombinationsfeld.
```

```
With cboDiagramme 'dem Auswahlfeld cboDiagramme werden die folgenden Punkte hinzugefügt
.AddItem "XY Linie mit Datenpunkten" 'ListIndex =0 (wird als default eingestellt)
.AddItem "XY interpolierte Linie mit Datenpunkten " 'ListIndex =1
.AddItem "XY nur Datenpunkte" 'ListIndex =2
.AddItem "Säulen gruppiert" 'ListIndex =3
.AddItem "Kegelsäulen gruppiert" 'ListIndex =4
```

End With

```
cboDiagramme.BoundColumn = 0 'Werte im Kombinationsfeld entsprechen den Werten fürListIndex
cboDiagramme.ListIndex = 0 'Kombinationsfeld auf ersten Eintrag festlegen
```

```
chkDatenanzeigen.Enabled = False 'Datenanzeigen deaktivieren da noch keine Daten vorhanden
End Sub
```

---

```
Public Sub AktiviereDiagramm() 'Aktiviert das Diagramm, wird für die Änderungen benötigt
ActiveSheet.ChartObjects("Diagramm 59").Activate
ActiveChart.ChartArea.Select
With ActiveChart 'Formatieren der Legende und
.HasTitle = True 'Benennung der Achsen
.ChartTitle.Characters.Text = "Messwerte vom " + Zeit
End With
End Sub
```

---

```
Public Sub datenaus()
'Auf die Sub-Prozedur kann von allen anderen Prozeduren in allen Modulen zugegriffen werden
AktiviereDiagramm 'Prozedur zum aktivieren des Diagramms wird aufgerufen
ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowNone, AutoText:=True, LegendKey:=False
'Beim aktiven Diagramm wird die Datenbeschriftung der Datenreihen wieder abgeschaltet
End Sub
```

### **Code der Tabelle 1**

```
Private Sub cmdFormular_Click()  
frmDatenVisualisierung.Show  
ActiveWindow.WindowState = xlMaximized 'Maximieren der aktuellen Arbeitsmappe  
End Sub
```

---

### **Code aus „Dieser Arbeitsmappe“**

```
Private Sub Workbook_Open()  
frmDatenVisualisierung.Show 'öffnet das Formular bei der Initialisierung der Arbeitsmappe  
End Sub
```

---

'Dies ist eine Möglichkeit ein Diagramm komplett automatisiert zu erstellen, allerdings  
'wird bei jedem neuen Diagramm der Name+1 vergeben (z.B. Diagramm1, neues Diagramm -  
>Diagramm2 usw)  
'daraus ergibt sich das Problem des ansprechens des neuen Diagrammes, auf dessen interne Namens-  
'vergabe man keinen Einfluss hat.  
'Daher ist es praktischer ein vorgefertigtes Diagramm (bei uns Diagramm 59) zu aktualisieren.

```
Private Sub cmdDiagramm_Click() 'Erstellen des Diagrammes aus den zuvor eingelesenen  
Range("A1:D12").Select 'Messdaten  
Charts.Add  
ActiveChart.ChartType = xlXYScatterLines 'Momentaner Diagrammtyp  
ActiveChart.SetSourceData Source:=Sheets("Tabelle1").Range("A1:D12"), PlotBy _  
:=xlColumns 'Festlegen der Diagrammwerte  
ActiveChart.Location Where:=xlLocationAsObject, Name:="Tabelle1" 'Festlegen des Namens des  
neuen Diagramm Blattes
```

```
With ActiveChart 'Formatieren der Legende und  
.HasTitle = True 'Benennung der Achsen  
.ChartTitle.Characters.Text = "Messwerte vom " + Zeit  
.Axes(xlCategory, xlPrimary).HasTitle = True  
.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Messpunkt-Nr."  
.Axes(xlValue, xlPrimary).HasTitle = True  
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Spezifische Einheit"  
.ChartTitle.Font.Size = 10 'Grösse der Überschrift auf 10 festlegen  
End With
```

'Formatierung der Größe des Diagrammes und der Position deselbigen

```
ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowNone, LegendKey:=False  
ActiveSheet.Shapes("Diagramm 2").IncrementLeft -178.5  
ActiveSheet.Shapes("Diagramm 2").IncrementTop 69#  
Windows("messdaten.xls").SmallScroll Down:=5  
ActiveSheet.Shapes("Diagramm 2").ScaleWidth 0.97, msoFalse, _  
msoScaleFromTopLeft  
ActiveSheet.Shapes("Diagramm 2").ScaleHeight 1.29, msoFalse, _  
msoScaleFromTopLeft  
Windows("messdaten.xls").ScrollRow = 1
```

```
cboDiagramme.Enabled = True 'aktiviert das Pulldown Menü cboDiagramme, nachdem das  
'Diagramm erstellt wurde
```

```
End Sub
```