

Eigene Programme in MicroStation V8i erstellen

Teil IV: Erstellen eines VBA-Programm-Dialogs

An dieser Stelle wollen wir Ihnen - in einer kleinen Serie - die verschiedenen Möglichkeiten der MicroStation-Programmierung vorstellen. An Hand von kleinen und einfachen Beispielen wollen wir gerade dem "Nichtprogrammierer" oder dem Anfänger einen verständlichen Einblick in die Möglichkeiten der MicroStation Programmierung geben.

Anhand des nachstehenden VBA Beispiels werden wir Ihnen das Erstellen eines einfachen Dialogs erläutern, mit dem Sie die MicroStation Elemente suchen und modifizieren können.

Beispielaufgabe:

Es sollen alle Linien und Linienzüge - die auf einer zu wählenden Ebene liegen - automatisch modifiziert werden, wobei die Wahl der Ebene über einen Dialog durchgeführt werden soll.

Die linearen Elemente sollen nach Abschluss des Programms folgende Eigenschaften besitzen

- Farbe = rot
- Strichstärke = 5
- Strichart = 3

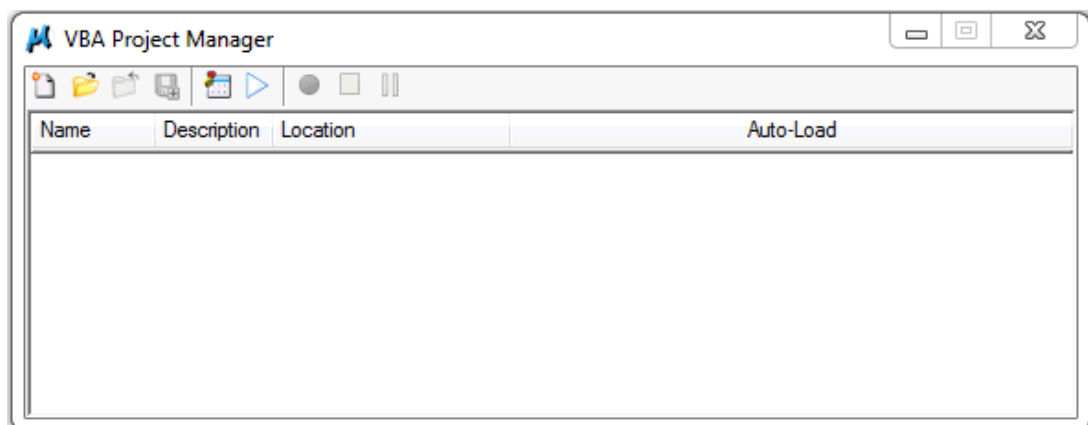
Alle anderen Elemente sollen nicht verändert werden.

Die Hauptaufgabe in diesem Teil besteht in der Erstellung des Dialogs und geringen Modifikationen im bereits bekannten Quellcode

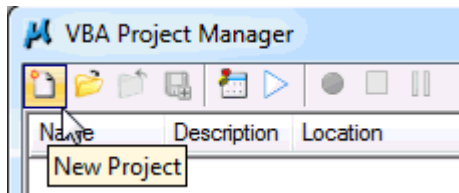
VBA Projekt Manager und IDE:

Über den VBA Projekt Manager wird ein neues VBA Projekt erstellt.

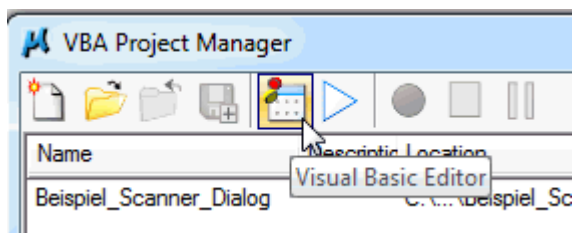
- Öffnen Sie über "Extra/Makro/Projekt Manager" den VBA Projekt Manager



- Über die nachstehende Schaltfläche "Neues Projekt" erzeugen Sie ein neues VBA Projekt

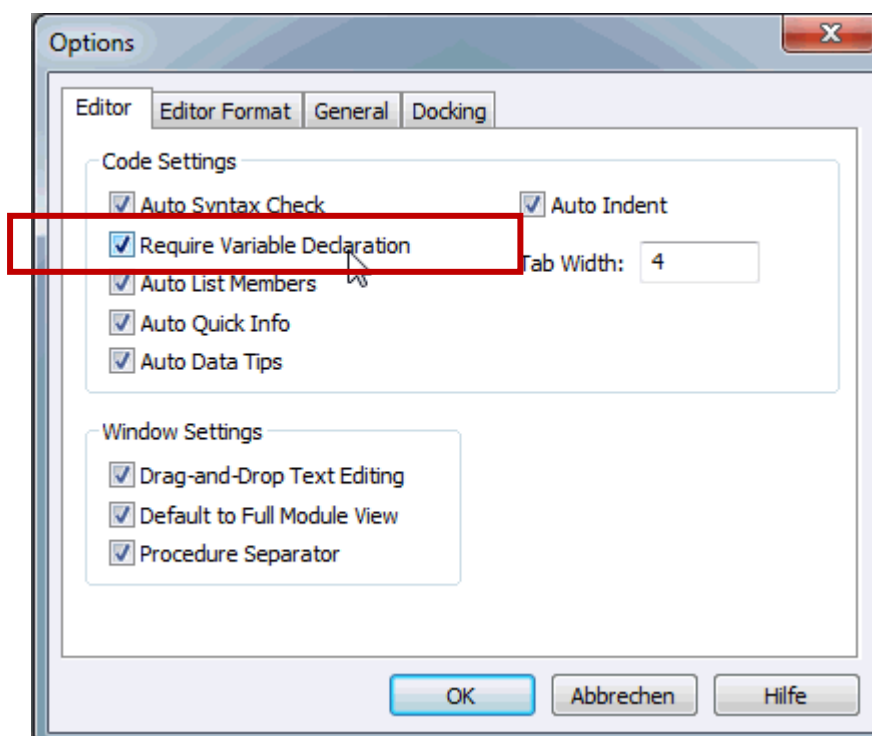


- Speichern Sie das Projekt unter dem Namen "Beispiel_Scanner_Dialog.mvba" in einem Ordner Ihrer Wahl
- Über die Schaltfläche "Visual Basic Editor" gelangen Sie in die VBA Entwicklungsumgebung IDE (Visual Basic for Applications Integrated Development Environment)



Vor dem eigentlichen Start sollte eine wichtige Einstellung vorgenommen werden, die dafür sorgt, dass Regeln bezüglich der Variablenverwendung bei der Erstellung des Codes eingehalten werden.

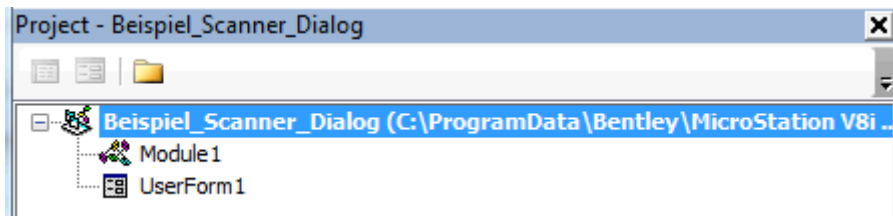
- Über das Menü "Tools/Options" gelangen Sie in den nachstehenden Dialog, in dem Sie die Option "Require Variable Declaration" bitte anhaken



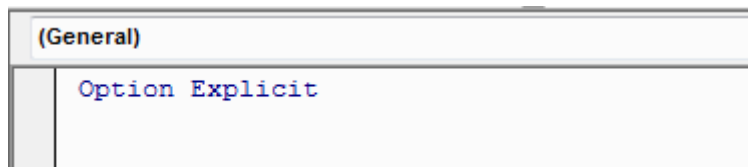
- Die Option "Require Variable Declaration" besagt, dass Sie vor der Verwendung einer Variablen diese zunächst definieren müssen.

VBA Programm:

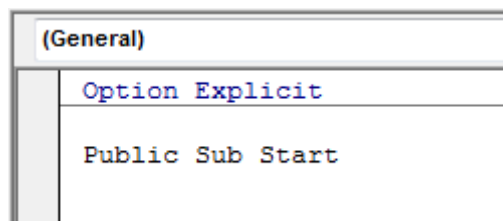
Jetzt sind die wichtigsten Einstellungen gemacht und es kann endlich losgehen. Im linken Teil der IDE (im Projektextplorer) befindet sich das eben neu angelegte Projekt mit dem Namen "Beispiel_Scanner_Dialog". Durch einen Doppelklick auf das Projekt bzw. den Projektnamen öffnet sich der "Projektbaum", in dem sich standardmäßig ein Modul mit dem Namen "Module1" befindet.



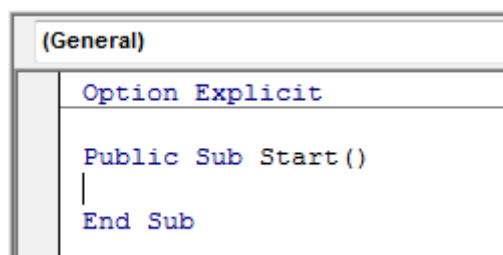
Auf der rechten Seite der IDE befindet sich das Text- bzw. Programmcode Fenster.



- Tragen Sie hier bitte den nachstehenden Code ("Public Sub Start") ein und bestätigen Sie Ihre Eingabe mit einem <Enter>



- Die VBA Engine vervollständigt automatisch Ihre Eingabe, siehe das Klammerzeichen "()" und das Ende der Funktion "End Sub".



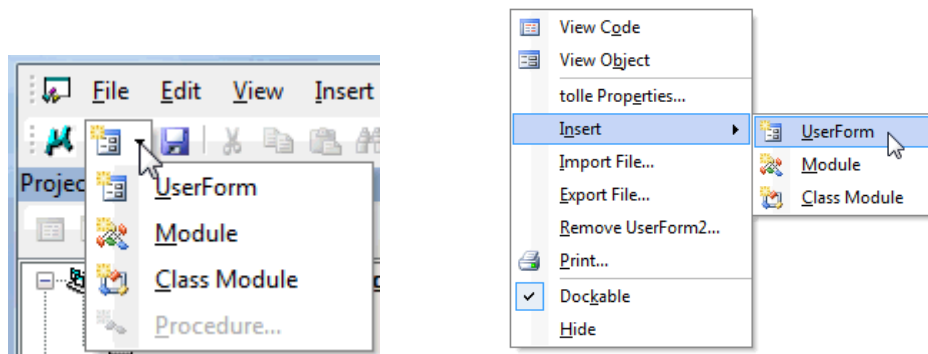
- An dieser Stelle haben wir in den vergangenen Übungen den Programmcode eingearbeitet, um den wir uns jetzt später kümmern werden.

Programmdialog erstellen:

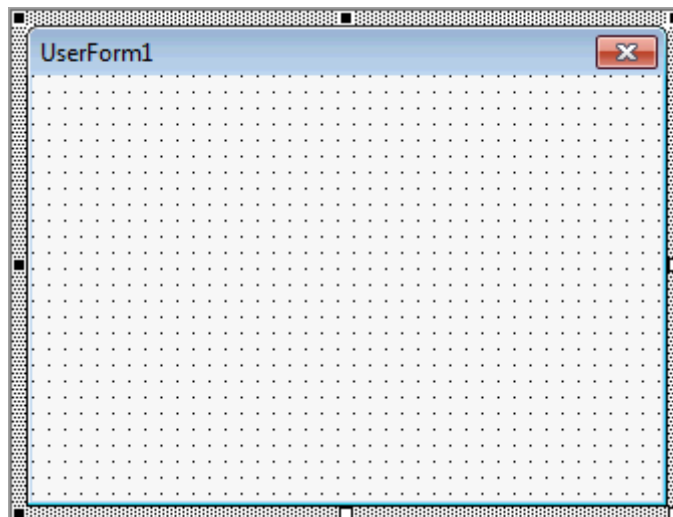
An dieser Stelle wollen wir uns um den Dialog kümmern, dieser soll die verwendeten Ebenen der Zeichnungsdatei in einer Auswahlliste zur Verfügung stellen. Über eine Schaltfläche sollen die Microstation Elemente Linien und Linienzug gesucht und modifiziert werden.

Den Weg zur Erstellung eines Programmdialogs könnte man wie folgt skizzieren

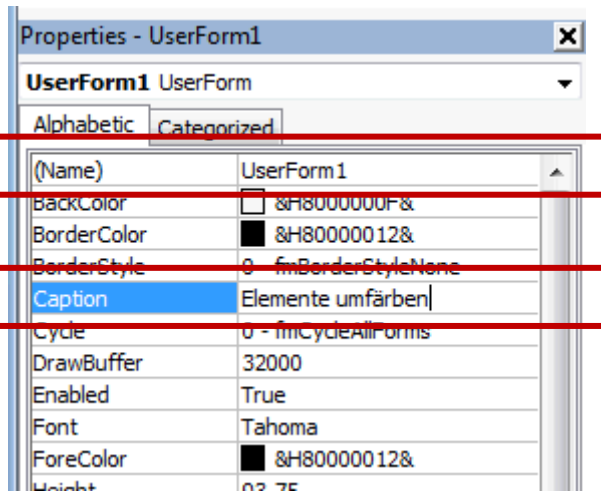
1. Erzeugen eines Formulars
 2. Auswahl der entsprechenden Komponenten per Drag-and-Drop
 3. Einstellen der Komponenteneigenschaften
 4. Wiederhole Schritt 2 und 3
- Über die rechte Maustaste im Fenster des Projektexplorers oder über die Schaltfläche "Insert UserForm" wird ein neues Formular erstellt.



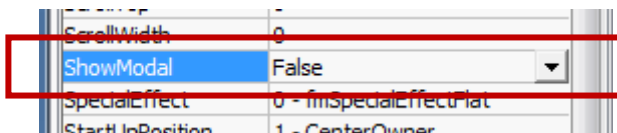
- Das nachstehende Formular mit dem Namen "UserForm1" erscheint, welches wir später in der Höhe und Breite anpassen werden.



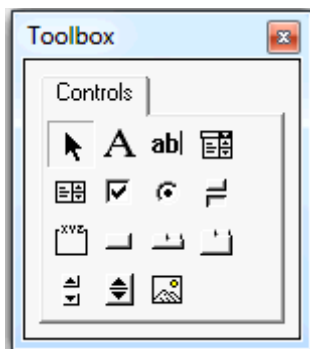
- In der linken Bildschirmhälfte werden die Eigenschaften des gewählten Objekts in einem Properties oder Eigenschaftsfenster dargestellt.
- Beachten Sie den Namen des Formulars "UserForm1", den wir so belassen.
- Verändern Sie die Eigenschaft "Caption" des Formulars in "Elemente umfärben", es handelt sich hierbei um den Titel des Formulars.



- Bei den Formularen gibt es die wichtige Eigenschaft "ShowModal", die wir auf FALSE setzen, um parallel zum geöffneten Dialog noch andere Tätigkeiten machen zu können.



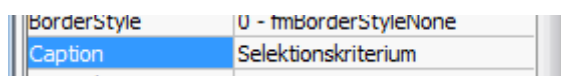
- Mit dem Formular geht automatisch die Toolbox oder Werkzeugleiste auf, in der verschiedene Komponenten (Rahmen, Textlabels, Schaltflächen, Auswahlboxen etc.) zur weiteren Verwendung stehen. Diese werden per "Drag-and-Drop" auf das Formular gezogen.



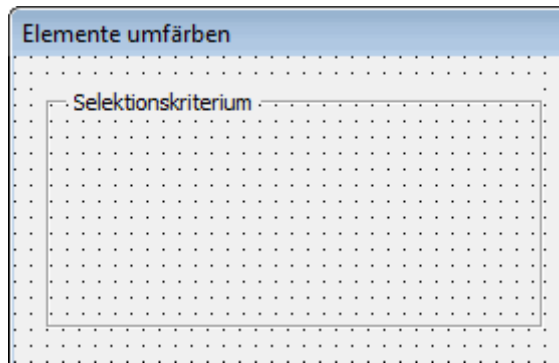
- Ziehen Sie die Umrandungskomponente "Frame" per Drag-and-Drop auf die Formularfläche



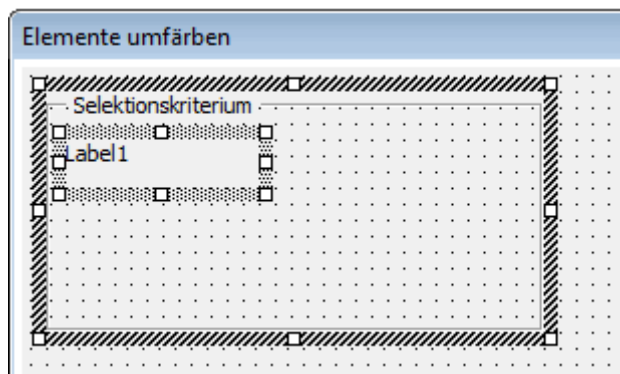
- Verändern Sie die Eigenschaft "Caption = Selektionskriterium"



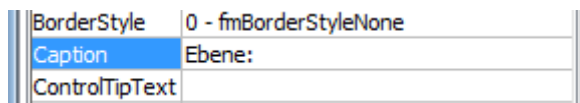
- Die Änderung der Eigenschaft wird auf dem Formular sofort sichtbar



- Ziehen Sie die Textkomponente "Label" per Drag-and-Drop auf die Formularfläche



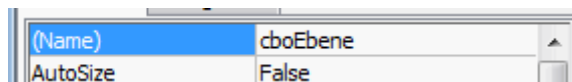
- Verändern Sie die Eigenschaft "Caption = Ebene:"



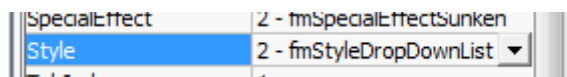
- Ziehen Sie die Listenauswahlkomponente "ComboBox" per Drag-and-Drop auf die Formularfläche



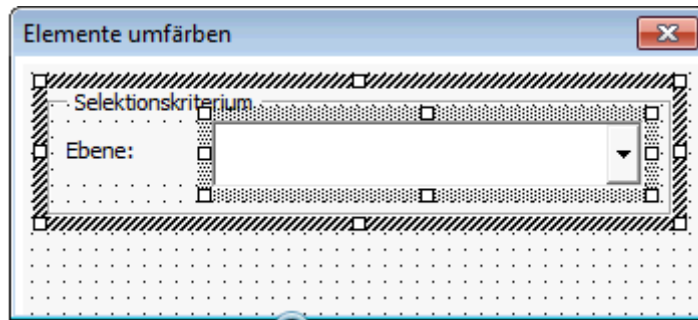
- Verändern Sie die Eigenschaft "Name = cboEbene"



- Verändern Sie die Eigenschaft "Style = fmStyleDropDownList"



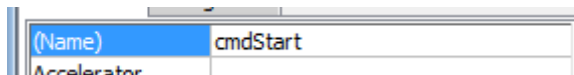
- Passen Sie die Größen der einzelnen Komponenten so an, dass sich ein harmonisches Gesamtbild ergibt



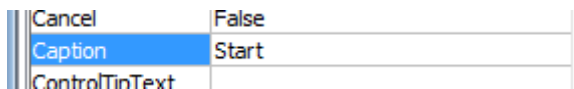
- Zu guter Letzt fehlt noch eine Schaltfläche, die den eigentlichen Bearbeitungsprozess ausführt.
- Ziehen Sie dazu die Schaltflächenkomponente "CommandButton" per Drag-and-Drop auf die Formularfläche



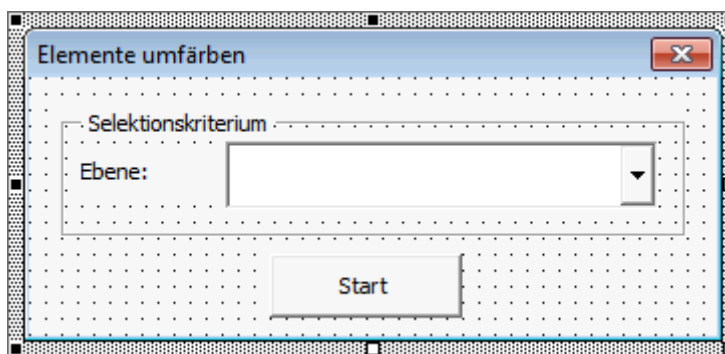
- Verändern Sie die Eigenschaft "Name = cmdStart"



- Verändern Sie die Eigenschaft "Caption = Start"



- Passen Sie die Größen der einzelnen Komponenten so an, dass sich ein harmonisches Gesamtbild ergibt



Das Design des Programmdialogs ist fertig, jetzt kümmern wir uns um den Quellcode, dazu übertragen Sie bitte den nachstehenden Code.

Allgemeine Anmerkungen zur besseren Lesbarkeit des Quellcodes werden durch ein Hochkomma eingeleitet.

Programmcode:

Im Vergleich zu den vergangenen Übungen wollen wir zunächst den Programmdialog laden und erst nach Drücken der Schaltfläche "Start" das Programm ausführen. Dazu benötigen wir zunächst einen Ladeprozess

- Durch einen Doppelklick auf das Modul1 im Projektextplorer erscheint auf der rechten Bildschirmseite der Quellcodeeditor, in dem Sie bitte die Funktion "Start ()" wie nachstehend definieren.

```

(General)
Option Explicit

Public Sub START()

    ' Formular laden und öffnen
    UserForm1.Show

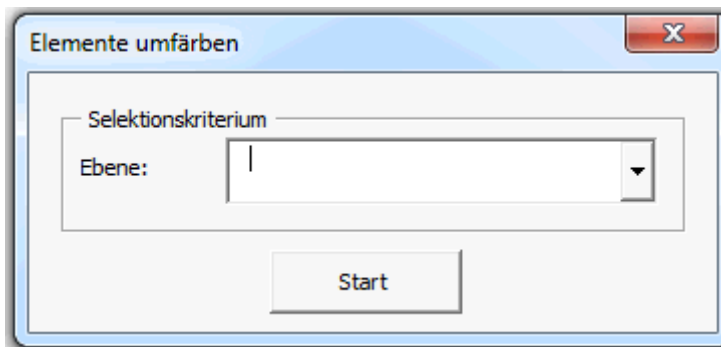
End Sub

```

- Über das Kommando "UserForm1.show" wird unser oben erstellter Programmdialog geladen
- Speichern Sie bitte Ihre Eingaben
- Über die Schaltfläche "Run Macro" können Sie Ihr Programm testen



- Haben Sie alles richtig gemacht, sollte Ihr Programmdialog in der MicroStation Oberfläche erscheinen

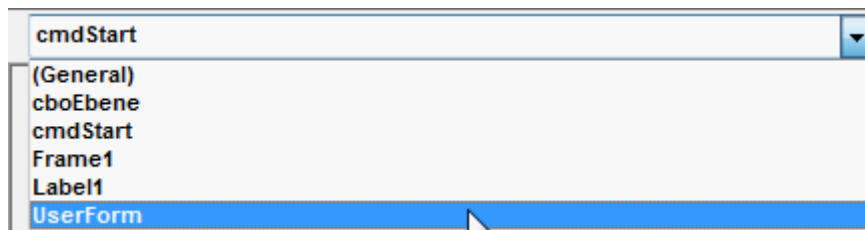


Neben dem Laden des Programmdialogs muss das Formular initialisiert und die Combobox automatisch gefüllt werden. Weiterhin muss die eigentliche Bearbeitungsroutine hinter die Schaltfläche "Start" gelegt werden.

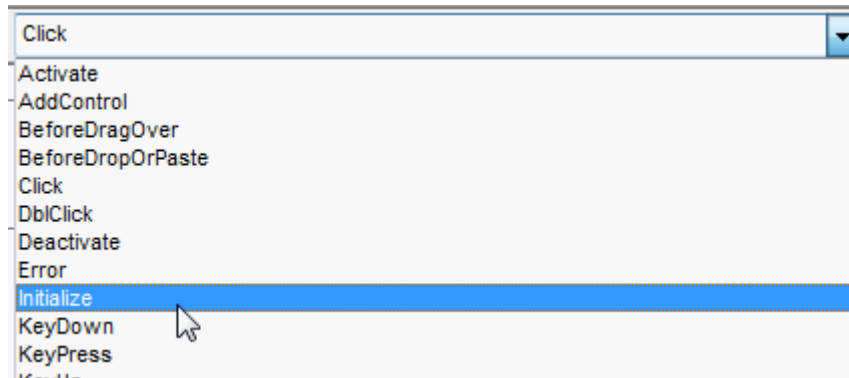
- Zur Initialisierung des Programmdialogs wählen Sie bitte die Darstellung des Quellcodeeditors



- Im Quellcodefenster wählen Sie bitte das Objekt "UserForm" aus



- Wählen Sie rechts daneben die Funktion "Initialize" aus



- Um die Ebenenauswahlliste automatisch zu füllen, vervollständigen Sie die Funktion "UserForm_Initialize ()" wie nachstehend dargestellt

```

Private Sub UserForm_Initialize()

    Dim lv As Level

    ' ComboBox leeren
    Me.cboEbene.Clear

    ' Alle Ebenen des aktuellen DesignFiles in die ComboBox einfügen
    For Each lv In ActiveDesignFile.Levels
        Me.cboEbene.AddItem lv.Name
    Next lv

    ' Aktive Ebene aus MicroStation auch in ComboBox aktiv setzen
    Me.cboEbene.Text = ActiveSettings.Level.Name

End Sub

```

- Damit wird beim Laden des Dialogs zunächst die ComboBox "cboEbene" geleert und anschließend mit den in Microstation zur Verfügung stehenden Ebenen gefüllt. Weiterhin wird die aktive Ebene in der Listenauswahl dargestellt.

Um die eigentliche Bearbeitungsroutine über die Schaltfläche "Start" ausführen zu können, muss der nachstehende Code in die Funktion " cmdStart_Click()" eingearbeitet werden. Dabei werden wir nur auf die Neuerungen bzw. Änderungen des Codes näher eingehen, denn der eigentliche Quellcode zur Bearbeitung der MicroStation Elemente ist in den vorangegangenen Übungen bereits näher erläutert worden.

- Variablendefinition

- "oScan" stellt die Suchmaske mit den später noch zu definierenden Elementkriterien dar
- "ee" repräsentiert eine Elementaufzählungsliste der gefundenen Elemente ("Trefferliste")

```
Private Sub cmdStart_Click()

    Dim ee As ElementEnumerator
    Dim oScan As ElementScanCriteria
    Dim lv As Level
    Dim ele As Element
    Dim ls As LineStyle
```

An dieser Stelle wird eine neue Suchmaske definiert. Mit Hilfe der Suchmaske werden später die relevanten MicroStation Elemente gesucht und gefunden.

- Mit den Befehlen "oScan.ExcludeAllLevels" und "oScan.ExcludeAllTypes" werden alle Ebenen und Elementtypen aus der Suchmaske entfernt

```
' Neue Suchmaske definieren
Set oScan = New ElementScanCriteria

' Zuerst alle Ebenen aus der Suchmaske entfernen
oScan.ExcludeAllLevels

' Alle Elementtypen aus der Suchmaske entfernen
oScan.ExcludeAllTypes
```

- An dieser Stelle wird sichergestellt, dass in der Ebenendefinition der DGN Zeichnung tatsächlich mindestens eine Ebene existiert, wobei die aktive - bzw. die im Dialog ausgewählte - Ebene als Suchkriterium genommen wird.

```
' Ebene im aktuellen Designfile suchen
Set lv = ActiveDesignFile.Levels.Find(Me.cboEbene.Text)
If lv Is Nothing Then
    ' Wurde die Ebene nicht gefunden, Abbruch
    Exit Sub
End If
```

Anschließend muss die Suchmaske mit entsprechenden Kriterien versehen werden.

- Das erste Kriterium ist die Einschränkung der Suche auf die ausgewählte Ebene
- Ein weiteres Kriterium ist die Einschränkung der Suchmaske auf die Elementtypen Linie und Linienzug
- Nach der Definition der Kriterien muss die eigentliche Elementsuche (das "Scannen") im aktiven Modell ausgeführt werden
- Das Ergebnis der Suche ist eine Liste von Elementen ("Trefferliste"), die anschließend Element für Element abgearbeitet werden muss

```

' ausgewählte Ebene dem Suchkriterium zuweisen
oScan.IncludeLevel lv

' Linien als Suchkriterium setzen
oScan.IncludeType msdElementTypeLine

' Linienzüge als Suchkriterium setzen
oScan.IncludeType msdElementTypeLineString

' Aktives Model scannen
Set ee = ActiveModelReference.GraphicalElementCache.Scan(oScan)

```

Über eine Schleife ("Do While / Loop"), die solange durchlaufen wird, bis es in der Trefferliste keine weiteren Elemente mehr gibt, wird jedes Element, welches den Kriterien der Suchmaske entspricht, bearbeitet.

- Setzen der Elementfarbe auf 3
- Setzen der Linienstärke auf 5
- Beim Setzen der Linienart ist eine Sicherheitsabfrage notwendig, die sicherstellt, dass es die zu setzende Linienart tatsächlich gibt
- Zu guter Letzt wird das modifizierte MicroStation Element gespeichert und in die DGN Zeichnung zurückgeschrieben

```

' Gefundene Elemente durchlaufen
Do While ee.MoveNext
  Set ele = ee.Current

  ' Setze Elementfarbe auf 3
  ele.Color = 3

  ' Setze Elementstrichstärke auf 5
  ele.LineWeight = 5

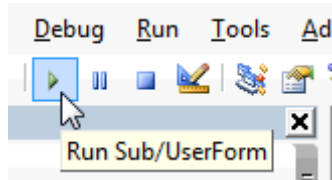
  ' Setze Elementlinienart auf 3
  Set ls = ActiveDesignFile.LineStyles.Find("3")
  If Not (ls Is Nothing) Then
    Set ele.LineStyle = ls
  End If

  ' Geändertes Element speichern
  ele.Rewrite
Loop
End Sub

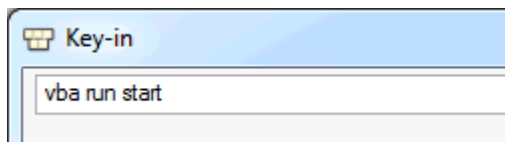
```

Programmaufruf:

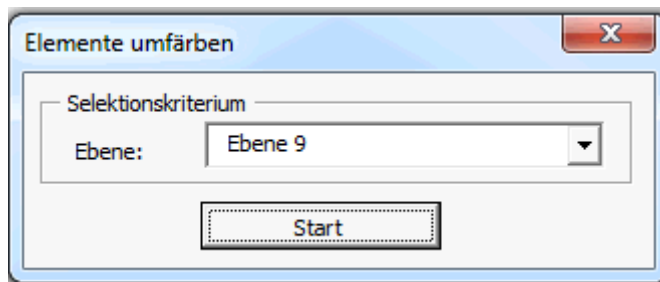
- Über den Menüeintrag "Debug/Compile Beispiel_Scanner_Dialog " wird der Quellcode formal auf Richtigkeit geprüft.
- Sollte ein Fehler im Programmcode vorliegen, wird dieser über einen Dialog angezeigt und muss bereinigt werden
- Über die Schaltfläche "Run Sub/UserForm" kann das Programm direkt ausgeführt werden



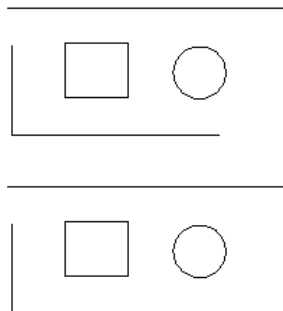
- alternativ kann das MVBA Programm über das MicroStation KeyIn oder Eingabefenster aufgerufen und gestartet werden
- befindet sich das MVBA in dem MicroStation Suchpfad und verwenden Sie nicht mehrere VBAs mit einer Funktion "Start", reicht der nachstehende Programmaufruf in der Regel aus



- Haben Sie alles richtig gemacht, wird die DGN Zeichnung automatisch durchsucht und alle Linien und Linienzüge, die auf der Ebene "Ebene 9" liegen, werden dick, rot und gestrichelt erscheinen.



Vorher



Nachher

