

Unpaarige Anführungszeichen und Klammern interaktiv korrigieren

Inhalt

1	Aufgabe	1
2	Benutzerformular	1
2.1	Steuerelemente	1
2.2	Befehlsschaltflächen	3
2.3	Bezeichnungsfelder	3
2.4	Ereignisprozeduren	3
3	"Dokument öffnen"-Ereignis	4
4	Standardprozeduren	4
4.1	Hauptprogramm	4
4.2	Unterprogramm	6
5	Erweiterungsmöglichkeiten	6
6	Abbildungen	8
7	Literatur	8

1 Aufgabe

In diesem Beitrag wird beschrieben, wie unpaarige Anführungszeichen und Klammern in einem WORD-Dokument gefunden und im Dialog korrigiert werden können. Gesteuert wird die Anwendung durch ein Benutzerformular mit dem Namen *frmDecisionBox*.

2 Benutzerformular

2.1 Steuerelemente

Das Benutzerformular *frmDecisionBox* ist mit vier Steuerelementen ausgestattet, zwei Befehlsschaltflächen und zwei Bezeichnungsfeldern (siehe [Abb. 1](#)).

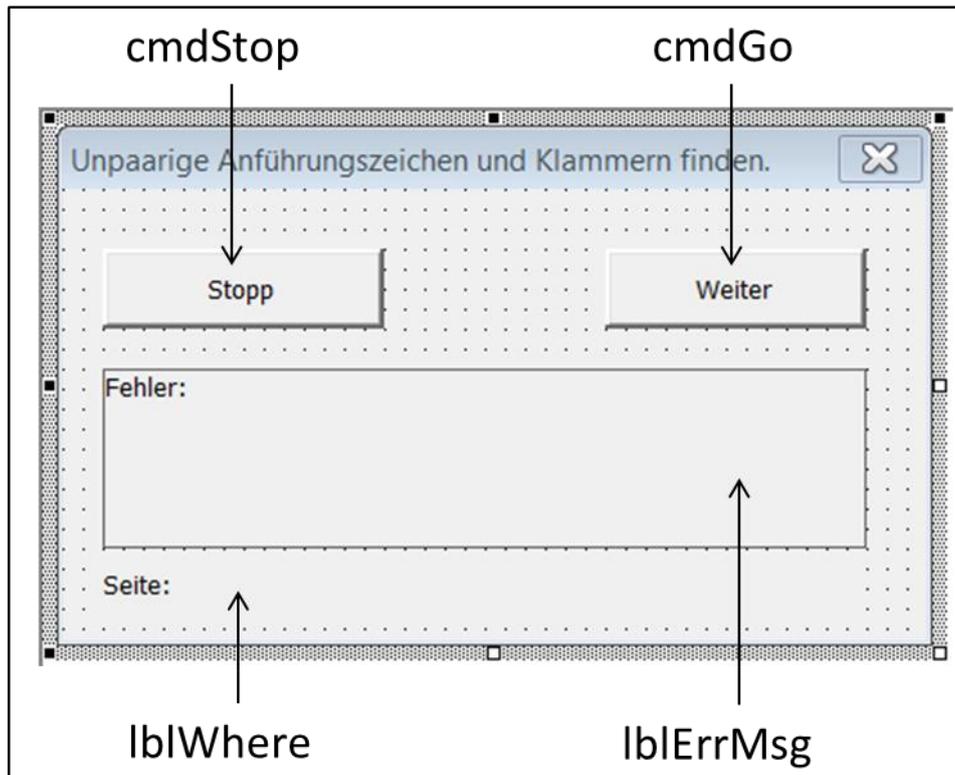


Abb. 1: Steuerelemente im Entwurf des Benutzerformulars mit dem Namen *frmDecisionBox*

Im Entwurf des Benutzerformulars (siehe Abb. Abb. 1) kennzeichnet die Vorsilbe *cmd* jeweils eine Befehlsschaltfläche, *lbl* jeweils ein Bezeichnungsfeld.

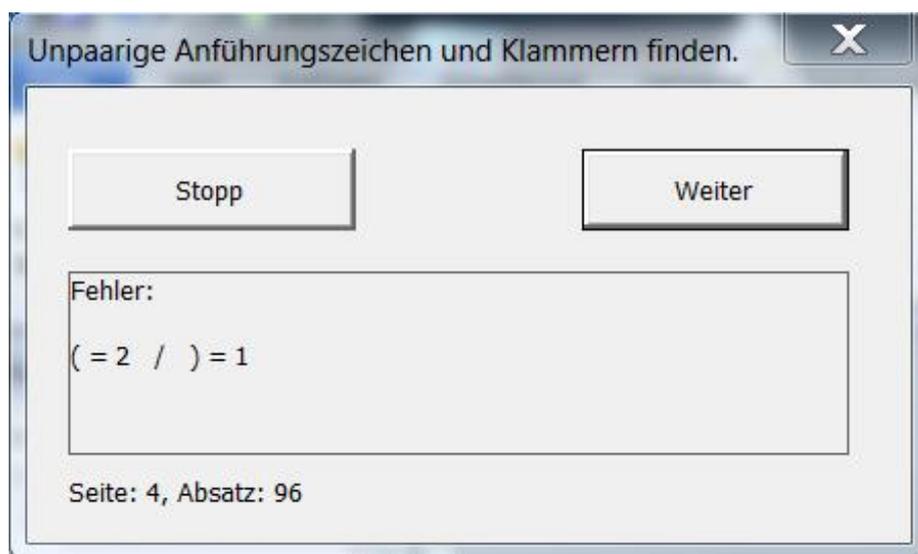


Abb. 2: Anzeige des Benutzerformulars bei gefundenen unpaarigen runden Klammern

Die Fehlermeldung in Abb. 2 besagt, dass 2 linke runde Klammern im Absatz Nr. 96 auf Seite 4 gefunden wurden, aber nur 1 rechte runde Klammer.

2.2 Befehlsschaltflächen

Die Befehlsschaltfläche *Stopp* besitzt den Namen *cmdStop*. Die Befehlsschaltfläche *Weiter* ist mit dem Namen *cmdGo* verbunden.

2.3 Bezeichnungsfelder

Das obere Bezeichnungsfeld trägt den Name *lblErrMsg*. Es dient zur Ausgabe von Fehlermeldungen. Das untere Bezeichnungsfeld mit dem Namen *lblWhere* zeigt entsprechend an, wo der Fehler gefunden wurde.

2.4 Ereignisprozeduren

Folgende vier Ereignisprozeduren sind mit dem Benutzerformular verbunden (siehe Listing 1):

```
Private Sub UserForm_Initialize()
    ' Anfangswerte setzen
    With Me
        .Left = 10           ' Position links
        .Top = 10           ' Position oben
        .StartupPosition = 0 ' manuelle Positionierung des Benutzerformulars
        .lblErrMsg.Caption = "" ' Bezeichnungsfeld für Fehler leeren
        .lblWhere.Caption = "" ' Fundstelle für Fehler leeren
    End With
End Sub

Private Sub cmdGo_Click() ' Schaltfläche "Weiter"
    frmDecisionBox.Hide ' Benutzerformular ausblenden
    ' Markierung nach rechts verschieben
    Selection.MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdMove
    Call LookForMismatchedPairs ' Standardprozedur aufrufen
End Sub

Private Sub cmdStop_Click() ' Schaltfläche "Stopp"
    frmDecisionBox.Hide ' Benutzerformular ausblenden
    Unload frmDecisionBox ' Benutzerformular aus dem Speicher entfernen
    Set frmDecisionBox = Nothing
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    ' Das Schließen des Benutzerformulars mit dem roten Kreuz verhindern
    If CloseMode = vbFormControlMenu Then Cancel = True
End Sub
```

Listing 1: Ereignisprozeduren des Benutzerformulars

Ein Klick auf die Befehlsschaltfläche *Weiter* startet die Standardprozedur mit dem Namen *LookForMismatchedPairs* (siehe gelbe Hervorhebung in Listing 1).

3 "Dokument öffnen"-Ereignis

Folgende Ereignisprozedur, die beim Öffnen des aktuellen WORD-Dokuments ausgeführt wird, ist im *ThisDocument*-Klassenmodul enthalten (siehe [Listing 2](#)).

```
Private Sub Document_Open()
    Load frmDecisionBox          ' Benutzerformular laden beim Öffnen des aktuellen Dokuments
    frmDecisionBox.Show vbModeless ' Das Benutzerformular wird ungebunden angezeigt
End Sub
```

Listing 2: „Dokument öffnen“-Ereignis in *ThisDocument*-Klassenmodell

Die Konstante *vbModeless* bewirkt, dass das Benutzerformular *frmDecisionBox* ungebunden aufgerufen wird. Deshalb kann der Inhalt des aktuellen Dokuments angesehen und korrigiert werden, ohne dass das Benutzerformular zuvor geschlossen werden muss.

4 Standardprozeduren

4.1 Hauptprogramm

Im Hauptprogramm mit dem Namen *LookForMismatchedPairs* (siehe [Listing 3](#)) besitzt das 2-dimensionale Datenfeld mit dem Namen *arrCheck* eine wichtige Aufgabe für die Kontrolle der Paarigkeit von Anführungszeichen und Klammern. Dort wird definiert, was paarig zusammengehört.

In der *Do-Until-Schleife* wird das zu untersuchende Dokument Absatz für Absatz durchlaufen und nach Unpaarigkeiten gesucht. In diesem Fall werden die beiden Bezeichnungsfelder (siehe oben) des Benutzerformulars aktualisiert und angezeigt. Da das Benutzerformular ungebunden angezeigt wird, kann der gefundene Fehler sofort interaktiv korrigiert werden.

```
Sub LookForMismatchedPairs()
    ' Aufgabe: Unpaarige Anführungszeichen und Klammern in allen Absätzen
    '           eines WORD-Dokuments finden und dabei interaktive Korrekturen
    '           mittels Benutzerformular zulassen.
    ' Benötigt: GetParaIndex
    ' In Anlehnung an: (o.V., 2006)
    '-----
    ' Höchster Index für 1. Dimension des zweidimensionalen Datenfeldes
    Const intCheck As Integer = 6
    Dim bolErrFlag As Boolean    ' Fehlerschalter
    Dim intLoop As Integer, intPos1 As Integer, intPos2 As Integer
    Dim strText As String, strErrMsg As String, strWhere As String
    ' zweidimensionales Datenfeld deklarieren
    Dim arrCheck(1 To intCheck, 1 To 2) As String
    ' Fehlerroutine definieren
    On Error GoTo Err_Point
    ' Datenfeldvariablen explizit deklarieren
    arrCheck(1, 1) = ChrW(8222) ' Anführungszeichen unten
    arrCheck(1, 2) = ChrW(8220) ' Anführungszeichen oben
    arrCheck(2, 1) = "("       ' runde Klammer auf
    arrCheck(2, 2) = ")"       ' runde Klammer zu
    arrCheck(3, 1) = "["       ' eckige Klammer auf
    arrCheck(3, 2) = "]"       ' eckige Klammer zu
    arrCheck(4, 1) = "{"       ' geschweifte Klammer auf
    arrCheck(4, 2) = "}"       ' geschweifte Klammer zu
```

```

arrCheck(5, 1) = ">"          ' dopp. spitzes Anführungszeichen (frz. guillemets) links
arrCheck(5, 2) = "<"          ' dopp. spitzes Anführungszeichen (frz. guillemets) rechts
arrCheck(6, 1) = "<"          ' spitzes Anführungszeichen links
arrCheck(6, 2) = ">"          ' spitzes Anführungszeichen rechts
' Do-Until-Schleife beginnen/fortsetzen
Do
  ' zum nächsten Absatz springen
  Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
  ' Text des aktuellen Absatzes zwischenspeichern
  strText = Selection.range.Text
  ' Fehlermeldung vorbereiten
  strErrMsg = "Fehler:" & vbCr & vbCr
  ' Fehlerschalter initiieren
  bolErrFlag = False
  ' aktuellen Absatz nach unpaarigen Anführungszeichen und Klammern durchsuchen
  For intLoop = 1 To intCheck
    intPos1 = UBound(Split(strText, arrCheck(intLoop, 1), -1, vbTextCompare))
    intPos2 = UBound(Split(strText, arrCheck(intLoop, 2), -1, vbTextCompare))
    If intPos1 <> intPos2 Then
      bolErrFlag = True ' Fehler gefunden
      Exit For          ' Schleife beenden
    End If
  Next intLoop
  If bolErrFlag Then ' Bei Fehler ...
    ' Fehlermeldung zusammenbauen
    strErrMsg = strErrMsg & arrCheck(intLoop, 1) & " = " & intPos1 & _
      " / " & arrCheck(intLoop, 2) & " = " & intPos2 & vbCr
    ' Fehlerposition bestimmen
    strWhere = "Seite: " & CStr(Selection.Information(wdActiveEndPageNumber)) & _
      ", Absatz: " & CStr(GetParaIndex)
    ' Fehler hervorheben
    With Selection
      ' fehlerhaften Absatz türkis einfärben
      .range.HighlightColorIndex = wdTurquoise
      .Collapse Direction:=wdCollapseStart
      ' Fundstelle des Fehlers gelb hervorheben
      If intPos1 > 0 Then
        .MoveUntil Cset:=arrCheck(intLoop, 1)
        .MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdExtend
        .range.HighlightColorIndex = wdYellow
      End If
      If intPos2 > 0 Then
        .MoveUntil Cset:=arrCheck(intLoop, 2)
        .MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdExtend
        .range.HighlightColorIndex = wdYellow
      End If
    End With
    ' ungebundenes Benutzerformular anzeigen und Korrekturen zulassen
    With frmDecisionBox          ' mit Formular ...
      .lblErrMsg.Caption = strErrMsg ' Fehlermeldung belegen
      .lblWhere.Caption = strWhere   ' Fundstelle
      .Show vbModeless            ' Formular ungebunden anzeigen
    End With
    Exit Sub
  End With
  ' Markierung nach rechts verschieben
  Selection.MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdMove
  ' Schleifenende prüfen mit in WORD eingebauter Textmarke
  Loop Until (Selection.Bookmarks.Exists("\EndOfDoc") = True)
MsgBox "Fertig!", vbExclamation, "LookForMismatchedPairs"

```

```

Exit_Point:
    Exit Sub
Err_Point:
    MsgBox Prompt:="Laufzeitfehler # " & err.Number & ", " & err.Description,
Buttons:=vbExclamation, Title:="LookForMismatchedPairs"
    Resume Exit_Point
End Sub

```

Listing 3: Quellcode des Hauptprogramms

4.2 Unterprogramm

Das Hauptprogramm benötigt das Funktions-Unterprogramm *GetParaIndex*. Es dient dazu, die Nummer des Absatzes zu ermitteln, in dem die jeweilige Unpaarigkeit festgestellt wurde (siehe Listing 4).

```

Function GetParaIndex() As Long
    ' Aufgabe: Aktuelle Absatznummer bestimmen
    ' Aufgerufen von: LookForMismatchedPairs
    On Error GoTo Exit_Point
    GetParaIndex = ActiveDocument.range(0, Selection.Paragraphs(1).range.End).Paragraphs.Count
    Exit Function
Exit_Point:
    GetParaIndex = 0
End Function

```

Listing 4: Aktuelle Absatznummer der gefundenen Unpaarigkeit bestimmen

Das Ergebnis der Funktion wird jeweils im Bezeichnungsfeld *lb/Where* des Benutzerformulars angezeigt, zusammen mit der zugehörigen Seitennummer.

5 Erweiterungsmöglichkeiten

Es bietet sich an, das vorgestellte Benutzerformular mit einer weiteren Befehlsschaltfläche zum Einfügen des Inhalts eines beliebigen anderen Word-Dokuments zu ergänzen (siehe Abb. 3).

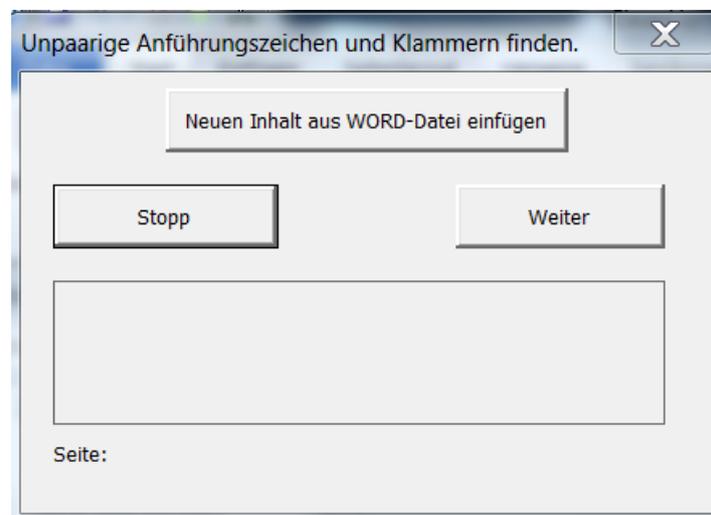


Abb. 3: Befehlsschaltfläche zum Einfügen von neuem Inhalt in das aktuelle WORD-Dokument

Die neue Befehlsschaltfläche heißt *cmdInsert*. Die zugehörige Ereignisprozedur befindet sich in Listing 5. Sie bewirkt den *Austausch* des gesamten Inhalts im aktuellen Dokument. Das erweiterte Benutzerformular sowie alle präsentierten Ereignis- und Standardprozeduren bleiben unverändert.

```
Private Sub cmdInsert_Click()
    Call DeleteAllStoryRanges ' Inhalt des aktuellen Dokuments komplett löschen
    Call InsertWordFile      ' neuen Inhalt aus ausgewähltem WORD-Dokument einfügen
    If ActiveDocument.Characters.Count > 1 Then ' Wenn neuer Inhalt vorhanden ist ...
        Me.Repaint          ' Benutzerformular erneut anzeigen
    Else
        MsgBox "Fehler: Kein Inhalt!", vbCritical, "Inhalt einfügen"
        Unload Me
    End If
End Sub
```

Listing 5: Ereignisprozedur für die neue Befehlsschaltfläche

Die beiden in Listing 5 gelb hervorgehobenen Standardprozeduren enthalten folgenden VBA-Code (siehe Listing 6):

```
Sub DeleteAllStoryRanges()
    ' Aufgabe: Inhalt des aktuellen WORD-Dokuments komplett löschen
    Dim rngStory As range
    For Each rngStory In ActiveDocument.StoryRanges
        rngStory.Delete
    Next rngStory
End Sub

Sub InsertWordFile(Optional strFileFilter as String = "*.docx")
    ' Aufgabe: Auswahl eines anderen WORD-Dokuments
    With Dialogs(wdDialogInsertFile)
        .Name = strFileFilter
        .Show
    End With
End Sub
```

Listing 6: Standardprozeduren zum Austausch des Inhalts im aktuellen Dokument

Um den gezeigten VBA-Code robuster zu programmieren, sind Verfeinerungen möglich, aber diese werden hier nicht weiter dargestellt.

Ein anderer Lösungsansatz für die behandelte Aufgabenstellung wird in (Wyatt, 2013) präsentiert.

6 Abbildungen

Abb. 1: Steuerelemente im Entwurf des Benutzerformulars mit dem Namen <i>frmDecisionBox</i>	2
Abb. 2: Anzeige des Benutzerformulars bei gefundenen unpaarigen runden Klammern	2
Abb. 3: Befehlsschaltfläche zum Einfügen von neuem Inhalt in das aktuelle WORD-Dokument	6

7 Listings

Listing 1: Ereignisprozeduren des Benutzerformulars	3
Listing 2: „Dokument öffnen“-Ereignis in ThisDocument-Klassenmodell	4
Listing 3: Quellcode des Hauptprogramms	6
Listing 4: Aktuelle Absatznummer der gefundenen Unpaarigkeit bestimmen	6
Listing 5: Ereignisprozedur für die neue Befehlsschaltfläche	7
Listing 6: Standardprozeduren zum Austausch von Inhalt im aktuellen Dokument	7

8 Literatur

o.V. 2006. Durchsuchen von einzelnen Textbereichen in Word. [Online] 02. 12 2006. [Zitat vom: 18. 12 2016.] <http://www.vb-fun.de/cgi-bin/forumarchiv.pl?archiv=0373&ID=1&action=zeigeseite&nummer=139366>.

Wyatt, Allen. 2013. Checking for Matching Parentheses. [Online] 04. 03 2013. [Zitat vom: 23. 12 2016.] http://word.tips.net/T001308_Checking_for_Matching_Parentheses.html.